Development of a CFD model and methodology for the internal flow simulation in a hydrogen-powered UAV

ALESSANDRO PORCARELLI
Development of a CFD model and methodology for the internal flow simulation in a hydrogen-powered UAV

Utveckling av CFD-modell och metodik för intern flödesimulering i vätgasdriven UAV

Alessandro Porcarelli
KTH, Royal Institute of Technology, Stockholm, Sweden
September 20, 2021

Abstract—In the context of an aviation industry whose top priority is to face the sustainability challenge, the growing civil UAV branch is not an exception. Hydrogen-powered UAVs equipped with PEM (Polymer Electrolyte Membrane) fuel cells are more and more frequently identified as the most convincing and promising technology, particularly for long-endurance mission requirements. However, the on-board carriage of a hydrogen fuel cell leads to unexplored internal flow characteristics, including the introduction of water vapour. The purpose of this master thesis is to develop a valid CFD model and methodology for the internal flow simulation of hydrogen-powered UAVs. Given the strict environmental operational requirements of PEM fuel cells, the intended application of the model is to effectively assess the evolution of the internal bay flow temperature and humidity fields. An explicit-time fourth-order Runge-Kutta projection method is tested successfully on a sample 2D case setup. The case geometry and flow conditions are inspired by the Green Raven UAV project conceived by the Department of Aeronautical and Vehicle Engineering at KTH.

I. INTRODUCTION

A. Hydrogen-powered UAVs

An unmanned aerial vehicle (UAV) can be defined as any aircraft without an onboard human pilot. It could be either remotely piloted or controlled with an autopilot. [1]

During the last decade, civil applications of UAVs are expanding their horizons, particularly in the surveillance field. A growth of the UAV industry is following from this development. The most promising solution to deal with the long endurance requirement of surveillance missions is combining batteries and hydrogen fuel cells in a hybrid propulsion system. On one hand, the high specific energy provided by hydrogen fuel cells allows to reach a large flight endurance, which common lithium polymer batteries cannot assure by themselves [2]. On the other hand, the batteries cope with the limited specific power of fuel cells, providing additional power during critical flight phases such as take-off and maneuvers [1].

There are many different available fuel cell types. Together with several other hydrogen UAV design groups, both Lapeña et al. [2] and Herwerth et al. [1] identify Polymer Electrolyte Membrane (PEM) fuel cells as the best choice for long-endurance UAV mainly due to their low operating temperature (around 60 °C). Other advantages include a long life cycle, the ability to withstand large pressure differentials and low noise and vibrations. PEM use hydrogen as fuel and oxygen from the air for the reaction. According to Lapeña et al. [2], considering the MTOW < 25 kg UAV category, PEM fuel cells at the current maturity stage already meet the weight target and offer competitive advantages.

B. Hydrogen for sustainable aviation

The hydrogen alternative for a more sustainable aviation has been widely discussed in the last decade. Pereira et al. [3] evaluate the environmental impact of alternative aviation fuels including liquid hydrogen (LH$_2$) using a
well-to-wake approach. Their study also considers the influence of the energy source from which $\text{LH}_2$ is produced. They show that the emissions of pollutants can be reduced up to 60% considering $\text{LH}_2$ produced through electrolysis with electricity from renewable sources. Hence, they conclude that renewable hydrogen is the less polluting alternative.

As it will be further explained in Section I-C1, PEM fuel cells do not deliver any pollutant, given that the only product of their chemical reaction is water. Therefore, it is necessary to focus on the way hydrogen is produced and/or supplied on-site in order to assess the environmental impact of PEM-powered UAVs. Considering hydrogen production, as also pointed by Pereira et al. [3], the price and carbon footprint will follow the development of renewable energies, which is currently seeing a significant growth.

Taking into account hydrogen supply systems, there are two main possible alternatives for PEM-powered unmanned aerial vehicles. The majority of UAVs rely on compressed gas hydrogen delivered to the UAV user’s location. The second option is producing hydrogen on board by means of chemical hydride systems. Lapeña et al. [2] suggest an exploration of the latter. However, the outcome related to both practical and environmental sustainability of the system turned to be disputable. Their work exposed the immature stage of the technology, underlining particular system maintenance challenges.

C. The output of the fuel cell reaction: water

1) Fuel cell reaction: In PEM fuel cells, hydrogen from the fuel gas stream is consumed at the anode. While the electrons produced go through an external circuit, the hydrogen ions (protons) move towards the cathode through the electrolyte. At the cathode, protons combine with air oxygen to produce water. [2]

Water is the only output of the reaction and is expelled from the cathode into the oxidant gas stream. The latter transports water in vapour state out of the fuel cell.

2) Importance of water disposal: Given the fuel cell chemical reaction, for each mole of reacting $\text{H}_2$ one correspondent mole of $\text{H}_2\text{O}$ is produced. The molar masses of $\text{H}_2$ and $\text{H}_2\text{O}$ are respectively around 2 g/mol and 18 g/mol. Hence, the total mass of water vapour produced and exhausted in the UAV bay can be roughly estimated to account for nine-ten times the hydrogen fuel "burnt" by the fuel cell.

Clearly, a flying vehicle with a landing weight above the take-off weight sounds itself very counter-intuitive, and would lead to unexplored flight mission managements. Moreover, the water vapour exhausted inside the aircraft internal bay might condense at the walls, potentially causing the centre of gravity to shift and thus affecting the overall vehicle stability. These two issues underline the need to dispose this exhausted water outside the bay.

Considering the possible environmental consequences of water disposal, Khandelwal et al. [4] analyse the global warming potential (GWP) of contrails and reassure about the sustainability of the process.

3) Fuel cell humidification: Before being exhausted, the water produced at the fuel cell cathode is used to continuously wet the fuel cell radiator to enable its evaporative cooling. This process is known as "humidification". Werner et al. [5] remark the important role of a proper humidification for the fuel cell performance. Their study explores the influence of several aviation-relevant parameters affecting the humidification process, such as operating pressure and temperature. Moreover, the influence of dynamic loads is further accounted by Gong et al. [6], and their study ultimately identifies in the limited operational robustness the major barrier to maturing fuel cells for commercial UAVs.

D. Ventilation of the UAV bay

As introduced in section I-C1, at the fuel cell cathode protons combine with oxygen from air. Clearly, the air in the proximity of the fuel cell intake should be sufficiently cool such that the cathode is always supplied with oxygenated air. As the total volume of air inside the UAV’s bay is limited, it is necessary to develop an intake-exhaust system such that proper ventilation is allowed.

Furthermore, most fuel cells are designed to operate up to a limit ambient temperature. At the same time, the fuel cell exhaust air-stream temperature is generally larger than the designed limited ambient temperature. An example manual where these temperatures are specified can be found for the Horizon H-1000 [7]. Therefore, a proper ventilation is necessary also to avoid overheating of the bay temperature, which could lead to overheating of the fuel cell itself and thus automatic shut off, as also pointed by Gong et al. [6]. Barroso et al. [8] remark the importance of the internal cooling and perform a very interesting wind tunnel heat transfer analysis. They explore the influence of geometrical and airflow parameters with the goal of designing an optimal passive cooling system for a hydrogen-powered UAV.

E. Intake-exhaust systems in UAVs

At the current state of the art in hydrogen fuel cell powered UAVs, there is very very limited information regarding intake-exhaust system designs. Many of the proposed designs do not consider this issue at all, while other implement basic systems supported by rough calculations or wind tunnel assessments.

A representative example of the way the problem has been approached so far can be found in two studies by Kim et al. [9] and Kim et al. [10], in which the emphasis is placed on the necessity to cool the stack to ensure its performance. In Figure 1, intakes have been introduced at the fuselage nose of the glider-shaped UAV. Hence, the air flows through the bay and is exhausted at the rear of the fuselage, taking advantage of the pusher propeller which attracts the flow. Further, Kim et al. [10] clarify that the intakes have been dimensioned such that they
Figure 1: Intakes of the UAV designed by Kim et al. [9].

Figure 2: Manufactured inlets 2a and outlets 2b of the UAV built by Herwerth et al. [11].

introduce a mass flow rate equal to five times the mass flow rate pulled by the fuel cell fan. Once the UAV has been equipped with these roughly dimensioned intakes, the wind tunnel tests provided evidence that sufficient air was supplied to cool down the stack. No optimization of the system is documented.

The most interesting example of an intake-exhaust system on hydrogen-powered UAVs is provided by Herwerth et al. [1] and is further developed by Chiang et al. [11]. Figures 2a and 2b respectively show the equipped intake and the exhaust ducts. The authors state that the inlet has been dimensioned to exceed the fuel cell fan flow rate by 25% [1]. Such mass flow rate turned out to be sufficient to cool down the stack and guarantee the quality of the chemical reaction. This is significantly lower than the 500% suggested by Kim et al., thanks to a much more refined bay flow management.

Considering non-hydrogen-powered UAVs, more detailed and complex passive cooling systems are documented. For instance, Panagiotou et al. [12] dimension cooling ducts to efficiently supply the payload with properly directed cooling air. The ducts are connected to aerodynamically-efficient NACA-shaped intakes whose preliminary sizing methodology is validated through an internal flow CFD simulation.

F. Goals and objectives

The purpose of the present work is to develop a CFD model and methodology to simulate the internal flow in the 'Green Raven' UAV. The 'Green Raven' project is a multidisciplinary research project aiming to develop a hybrid-electric fixed-wing UAV powered by a hydrogen fuel cell [13].

The Green Raven falls within the $MTOW < 25 \text{ kg}$ UAV category and its design consists of a blended-wing-body (BWB) configuration (Figure 3), with a wingspan of 4m [14]. It is planned to be equipped with the Horizon H-1000 PEM fuel cell supplied by hydrogen stored onboard in a compressed tank. According to the user manual [7], the fuel cell system is self-humidified through an internal controller. Given that the UAV is intended to fly at operating altitudes around sea level, the operating temperature and pressure are not expected to vary in a wide range. Therefore, for the purposes of this study, the humidification system can be safely considered a 'black box'.

For its overall mission, the Green Raven is expected to carry onboard a total of 80g of $H_2$. According to the estimation in Section I-C2, this means that during the flight around 800g of water vapour to dispose will be produced. None of the studies available in literature (Section I-E) mention the way the fuel cell products are exhausted. However, water vapour disposal is more crucial in a BWB configuration like the Green Raven’s, since a potential near-wall condensation could shift the centre of gravity and affect the BWB light stability.

The Horizon H – 1000 is designed to operate up to an ambient temperature of 30 °C, and shut off occurs whenever its inner operating temperature exceeds 65 °C [7]. Due to its narrow internal spaces, the BWB configuration requires a more careful management of the heat released by the fuel cell. Moreover, the 1000 W fuel cell will introduce in the UAV bay a greater internal dissipation heat flux in comparison to the 150 W of Herwerth et al. [1], thus probably requiring a larger flow rate of cooling air.

With these considerations in mind, the developed CFD solver for the internal flow should consider both the heat transfer and the humidity. Given a sample 2D geometry with a sample intake-exhaust system, different ways of accounting these two combined flow features will be explored.
and assessed.

II. THEORY

A. Governing equations

Employing the continuum hypothesis, the fluid-dynamic behaviour of air is described by the Navier-Stokes equations, i.e. the mass scalar, the momentum vector and the energy scalar conservation equations. At this stage, these 5 partial differential equations (PDE) contain 7 unknowns. As the Green Raven cruise speed is $U = 20 \text{ m/s}$ [14], it is safe to assume the flow to be incompressible. Hence, the Navier-Stokes equations can be simplified, and the energy equation can be decoupled from the mass and momentum equations, which will form a new closed system. Furthermore, as the turbulent entities are a defining feature for internal flows, it is necessary to employ a turbulence modelling. To achieve a good quality of the mean turbulent flow without reaching too high computational cost, a Reynolds averaged Navier-Stokes (RANS) approach has been chosen. RANS are based on a statistical modelling technique, that consists in a decomposition of the flow variables (e.g. the velocity $U$) into a mean $\bar{U}$ and a fluctuating contribution $u$, where the $\bar{U}$ corresponds to an ensemble average:

$$U = \bar{U} + u.$$ (1)

This so-called "Reynolds decomposition" is applied to the Navier-Stokes equations resulting in the RANS equations:

$$\frac{\partial U_i}{\partial t} + \frac{\partial}{\partial x_j} (U_i U_j) = - \frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\nu \frac{\partial U_i}{\partial x_j} - \bar{u}_i u_j).$$ (2a)

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = - \frac{1}{\rho} \frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j} (\nu \frac{\partial U_i}{\partial x_j} - \bar{u}_i u_j).$$ (2b)

Whereby all $U_i$ correspond to $\bar{U}_i$, as the mean symbol has been skipped for simplicity. In accordance with the UAV internal flow CFD methodology employed by Panagiotou et al. [12], the gravity term has been neglected in the momentum equation, since the flow is expected to be dominated by the advection and the diffusion term. Further considerations on this assumption will be discussed in Section V.

B. Turbulence

The last term in Equation 2b consists of the divergence of the so-called "Reynolds stress tensor", and the latter should be modelled in order to close the system. Most of RANS turbulent closures are based on the Boussinesq hypothesis, where the traceless part of the Reynolds stress tensor is assumed to be proportional to the Strain Rate Tensor $S_{ij}$ with the coefficient $\nu_T$ (Eddy Viscosity):

$$\bar{u}_i \bar{u}_j = 2 \nu_T S_{ij} - \frac{2}{3} k \delta_{ij}. \quad \text{(3)}$$

By dimensional analysis, it holds that $\nu_T$ is proportional to the product of a turbulence velocity and length scale. These two parameters can either be prescribed (algebraic models), partially prescribed (one-equation models) or a transport equation can be solved for both of them or for related quantities (two-equations models).

C. Heat transfer

Given the incompressible flow assumption (Section II-A), the temperature $T$ can be be solved as a passive scalar. Neglecting the radiation term, it follows a standard advection-diffusion equation:

$$\frac{\partial T}{\partial t} + U_i \frac{\partial T}{\partial x_i} = D_T \frac{\partial^2 T}{\partial x_i^2}. \quad \text{(4)}$$

One could object intuitively that a substantial change in temperature changes the flow thermodynamic pressure. In an incompressible flow in an open domain box, however, this thermodynamic pressure is decoupled from the fluid dynamic pressure included in Equation 2b, so the solution of the governing equations is not affected.

Furthermore, the constant-density assumption could appear restrictive if a wide temperature range is experienced in the flow domain, thus causing a non-negligible density change. However, as otherwise the model complexity would increase substantially, the density in the momentum Equation 2b is considered constant. The mean flow features should not change significantly, but the influence of this assumption on the solution should be further assessed (Section V).

The air thermal diffusivity $D_T$ is given by the sum of a physical contribution $\alpha$ and a modelled physical turbulent contribution $\alpha_T$, where $\alpha$ can be found as

$$\alpha = \frac{k}{\rho c_p},$$ (5)

whereby $k$ is the thermal conductivity and $c_p$ is the specific heat capacity. Given a constant thermodynamic pressure, $k$ and $c_p$ only vary with $T$ itself. As the maximum temperature range should be limited ($\Delta T_{\text{max}} \approx 40 \text{ °C}$), $\alpha$ should not vary as substantially to affect the scalar transport significantly. Therefore, in order to simplify the model, $\alpha$ is assumed constant. Considering $\alpha_T$, a constant value educated guess should be performed by means of Reynolds analogy, i.e. defining a turbulent Prandtl number

$$Pr_T = \frac{\nu_T}{\alpha_T}. \quad \text{(6)}$$

D. Humidity

To account for humidity in the computations, it is necessary first to choose a proper humidity definition. The specific humidity could be pointed as the most indicated parameter, and it is defined as the weight of water vapor $m_v$ contained in a unit weight of moist air $m_v + m_d$

$$q = \frac{m_v}{m_v + m_d} = \frac{r}{1 + r} \quad \text{(7)}$$

where $m_d$ is the mass of dry air. As Equation 7 shows, the definition can be re-arranged as a function of the mixing ratio $r$, which is defined as $m_v/m_d$ [16] and can be found given the equilibrium water vapour pressure $p_0$ in $mbar$ as

$$r = \frac{0.622 p_0}{p_{\text{atm}} - p_0} \quad \text{(8)}$$

for which $p_0$ can be found for a given constant temperature from Antoine’s equation [17].
Humidity variation can affect the density in the momentum Equation 2b, and varies itself with temperature, as \( p_0 \) is dependent on \( T \). Similarly to the assumption performed for the temperature influence on the density in Section II-C, in order to limit the model complexity, the influence of \( q \) on \( \rho \) is not accounted in the model. Hence, \( q \) is similarly solved as a passive scalar, and its transport equation reads:

\[
\frac{\partial q}{\partial t} + U_i \frac{\partial q}{\partial x_i} = D_q \frac{\partial^2 q}{\partial x_i^2}.
\]  

(9)

Moreover, in order to keep the two scalars separated and substantially simplify the model, the influence of \( T \) on \( q \) is neglected. As a consequence, the model will imprecisely estimate the same \( q \) in two points with the same \( m_v \) and different \( T \). However, with the goal of understanding the humidity transport and distribution, and ultimately evaluate water vapour disposal, this assumption can be considered acceptable, as will be confirmed in Section IV-D.

Under particular conditions, for instance in near-wall, low-speed or low-temperature flow, the water vapour can condense into liquid water. Nonetheless, for the class of internal flows investigated, the moist air leaving the fuel cell will be invested by a moving flow of dry air with a relevant momentum. This flow feature, together with a limited particle resident time inside the bay, establishes non-favourable conditions for condensation. Therefore, for the considered class of UAVs internal flows, in order to limit the model complexity, the water condensation is not taken into account. Even if considering a larger fuel cell stack temperature and a lower ambient temperature, in the wind tunnel model tested by Barroso et al. [8] there is no hint that condensation occurs, thus raising confidence in the reliability of the assumption.

In analogy to \( D_r \), the water vapour diffusivity in air \( D_q \) consists of the sum of a physical contribution \( D_{q_p} \) and a modelled turbulent contribution \( D_{q_T} \). The former can be found for a given temperature as

\[
D_{q_p} = 0.230 \left( \frac{T}{273} \right)^{1.81},
\]  

(10)

according to Schirmer’s formula found in Lee et al. [18]. Similarly to \( \alpha \) in Section II-C, chosen a reference temperature, \( D_{q_p} \) is taken constant. Considering the latter, a constant value should be again educatedly guessed with Reynolds analogy as for \( \alpha_T \), defining a turbulent Schmidt number

\[
Sc_T = \frac{\nu_T}{D_{q_T}}.
\]  

(11)

E. Numerical schemes and solver

In order to numerically resolve the flow, the governing equations have to be discretized in space and time. The choice of the discretization numerical schemes will determine the consistency and the stability of the problem which are necessary conditions for its convergence.

The consistency property is related to how well the discretized equations approximate the PDE. The difference between the exact PDE and its discrete representation is called truncation error. To allow the consistency of the problem, this error needs to go to zero decreasing the grid spacing in space (\( \Delta x \), i.e. mesh spacing) and time (\( \Delta t \)). On the other hand, stability is related to the sensitivity of the solution to disturbances, which can be errors from any source. If a problem is unstable, the solution might blow up at some time. For convectively dominated transient problems, the so-called CFL stability condition can be derived considering the advection term of Equation 2b

\[ Co = \frac{U \Delta t}{\Delta x} \leq 1, \]  

(12)

where \( Co \) is called ‘Courant’ number. The presence of \( \Delta x \) in the condition also points out the importance of the mesh quality to determine the problem stability, as a single poorly shaped cell can blow up the simulation. Hence, particular attention should be focused on the fulfillment of the standard mesh quality criteria prescribed by the solver. [19]

For the Lax equivalence theorem, consistency and stability are the necessary and sufficient conditions for the convergence of the problem. The convergence property links the numerical solution to the exact solution of the PDE. If a simulation is converged to very low residuals, the two solutions will be very close, i.e. the numerical simulation closely reproduces the real flow field. However, there are some cases where the two solutions do not accurately match even if numerical convergence is achieved. This generally occurs in one or more of the following conditions:

- the chosen order of accuracy is too low, i.e. the truncation error introduced is too large;
- the grid spacings \( \Delta x \) and \( \Delta t \) are large, leading similarly to a large truncation error;
- the overall mesh quality is poor for complex flow geometries discretized with unstructured grids.

In light of the above considerations, with the goal of obtaining the best numerical solution, it might seem obvious that one should employ the highest order of accuracy possible with very refined grid spacings. However, the higher the order of accuracy, the higher is the possibility that instability and uncontrolled oscillations grow [19]. Moreover, higher-order schemes and refined grid spacings require larger computational time and are thus more expensive. Therefore, a reasonable compromise should be found.

F. OpenFOAM solvers

OpenFOAM is a an open-source C++ toolbox for the development of customized numerical solvers, most prominently addressing CFD [20]. It provides a wide range of built-in solvers suitable for a broad variety of classes of flows. Its flexibility lies in the possibility to customize these built-in solvers according to the particular features required by the flow case investigated. For instance, in
this work an incompressible PISO-based algorithm will be adjusted to account for heat transfer and humidity, implementing the equations reported in Sections II-C and II-D.

The PISO algorithm (Pressure Implicit with Splitting of Operators, of Issa 1986 [21]) is a non-iterative transient calculation procedure, where all time-dependent terms are retained in the momentum and continuity equations [22]. PISO consists of an implicit second-order backward time integration method which couples pressure and velocity in each timestep [21]. The coupling involves one predictor step, where an intermediate velocity is computed using the pressure of the previous timestep, and a number of corrector steps, where the intermediate and final velocity and pressure are obtained iteratively [23]. If solved for a long period of time with guessed initial conditions, a steady state can sometimes be achieved. Even if computationally expensive, a transient solver is highly recommended for internal flows accounting for heat transfer and other scalars. Further details on the PISO algorithm can be found in academic textbooks, such as Versteeg and Malalasekra [22].

As mentioned in Section II-E, low-quality cells can cause instability and potentially simulation divergence. While most commercial solvers are equipped with numerical correctors limiting mesh-originated instabilities, OpenFOAM is very sensitive to the mesh quality. Hence, particular attention should be focused on the mesh quality criteria especially the so-called ‘cell non-orthogonality’.

G. Pointwise

Pointwise is a commercial mesh generator software specialized in CFD applications. It covers all aspects of pre-processing, from import of geometric models, to meshes export directly with the proper format for selected flow solvers, including OpenFOAM. High-quality grids can be created quickly, intuitively, and eventually through script programming. Its highly customizable grids and its quality-check tool are very helpful to allow a good simulation convergence, particularly given the complex geometry and flow pattern investigated (Section III-A) and the strict quality limitations of the solver.

III. CFD CASE

A. Geometry

In order to properly test the developed CFD model, a realistic sample case should be built. The case geometry should desirably fulfil the following requirements.

1) It should be computationally cheap to allow fast iterations of the model tests.
2) It should reproduce as close as possible the real features of the Green Raven UAV, so that the results will be representative of realistic internal flow features.
3) The sample intake and exhaust should be placed and dimensioned appropriately, assuming that this configuration will be a starting point for a potential future optimization.
4) In order to appreciate relevant temperature and humidity changes, and assess the model functioning, the intake-exhaust system should introduce a coolant flow rate comparable with the stack flow rate. The resulting strong temperature and humidity increase inside the UAV bay will be deliberately exaggerated.

In order to substantially save computational cost, a 2D configuration has been chosen. The mid-cross-section of the Green Raven has been reconstructed. It consists of the airfoil MH104 with chord $c = 1.7m$ rotated according to the geometrical twist and the cruise angle of attack $\alpha = 1.357 ^{\circ}$ [14]. The fuel cell and the tank have been placed inside the internal bay according to a preliminary stability analysis performed in [14]. The resulting CAD configuration is shown in Figure 4.

In the configuration by Herwerth et al. [1] (Section I-E), the intake-exhaust system takes advantage of the fuel cell fans to efficiently convey the flow longitudinally towards the rearward outlet. In contrast, Figure 4 shows that the only way for the fuel cell to fit inside the BWB is to lie down on the bay floor. Hence, the fans will pull the air through the fuel cell and out of the stack perpendicularly to the longitudinal direction. Therefore, the designed intake-exhaust system cannot exploit the external flow momentum. On the other hand, a beneficial feature of the BWB configuration lies in its airfoil’s external pressure distribution. In fact, the intake-exhaust system can take advantage of the higher pressure regions to introduce cool air inside the bay and of lower pressure regions to exhaust the internal flow outside the bay. Therefore, as shown in Figure 4, the intake has been placed at the leading edge close to the external flow stagnation point, and the exhaust is located in the upper surface in the proximity of the airfoil maximum thickness. The dimension of the intake and the exhaust has been kept as small as possible in accordance with the fourth requirement listed above. The consequence of this choice will be discussed in Section III-B.

B. Grid

The computational domain consists of the 2D internal bay geometry shown in Figure 4 (Section III-A). The inlet
and the outlet are located respectively in correspondence of the leading edge intake and the upper surface exhaust. Their presence allows to thermodynamically consider the domain an open box, thus permitting to decouple thermodynamic and fluid-dynamic pressure as mentioned in Section II-C. Moreover, the fuel cell bottom fans pull the flow out of the domain, while the top stack re-introduces in the domain the same air flow rate. The rest of the internal surfaces act as walls, including the tank and the airfoil interior shell.

As mentioned in Section II-E, the spatial discretization of the computational domain, i.e. the mesh designed, can affect stability, convergence and reliability of the simulation results. First of all, the mesh should be fine enough to capture all the relevant flow features. As the intake and the exhaust act as inlet and outlet and thus play a key role in the flow definition, their patch should consist of a significant number of grid points. At the same time, as mentioned in Section III-A, the intake and exhaust dimension has been kept as small as possible. Therefore, the grid $\Delta x$ results to be very small, as shown in Figure 5. Given the cost-effectiveness of a 2D configuration, from the spatial perspective this does not represent a major expense issue. However, as discussed in Section II-E, in order to preserve the simulation stability, the $\Delta t$ should be adjusted according to $\Delta x$ such that the CFL condition (Equation 12) is verified. In this case, the resulting necessary very small $\Delta t$ increases the computational expense substantially. Nevertheless, as it will be shown in Section IV-A1, for the purposes of this study of assessing the model validity and appreciating relevant temperature and humidity changes, it is not necessary to run the transient simulation for a long physical time and this trade-off is thus accepted. In a future 3D optimization case, conversely, larger local span-wise intakes can be designed, thus allowing a reasonably large $\Delta x$ yielding a reasonably large $\Delta t$.

One of the main mesh design requirements is that its edges should be parallel to the gradients. Hence, in order to capture the expected inlet jet-like flow and the outlet confluence flow, a cone-shaped domain fully consisting of quadrilaterals has been designed in correspondence of the intake and the exhaust (Figure 5a and 5b).

Considering the near-wall region, since the most significant gradients related to the boundary layer are in the perpendicular direction to the surface, layers of structured rectangular high aspect ratio cells are inflated above the surfaces. Even if the boundary condition is not "wall" (see Section III-E), the inflation is effectively designed also in correspondence of the fuel cell stack and fans. In fact, as it will be confirmed in Section IV, the order of magnitude of the in-flowing and out-flowing perpendicular velocity is much smaller than the horizontal velocity of the "primary" flow.

The $y^+$ is a non-dimensional wall unit calculated as

$$
y^+ = \frac{y u^+}{v} \quad (13)
$$

where $y$ is the distance from the wall and $u^+$ is the Shear Velocity. In very close proximity to the wall, i.e. in the so-called viscous sub-layer where $y^+ < 5$, the turbulent fluctuations are gradually suppressed by viscous effects. To allow a good near-wall resolution, the height of the first inflation cell should be small enough to fully enclose only the viscous sub-layer region. For most external flows, a dimension estimation can be derived by means of the flat plate turbulent boundary layer formulas, given a reference length $L$ and velocity $U$. For the considered complex internal flow, $h_1 = 0.5 \text{ mm}$ is estimated in the same manner. However, as it is very difficult to realistically choose $L$ and $U$, the estimation validity should be further verified by means of a $y^+$ check, which will be later performed in Section IV-A3. Nevertheless, considering the case geometrical features, the flow is not expected to be significantly affected by the near-wall resolution.

The design of the mesh features is realized by means of Pointwise T-rex algorithm. Given the inflation first cell thickness $y_1$ and a growth factor, Pointwise inflates rectangular cells until a very smooth dimension transition with the parent mesh is achieved. The parent mesh consists of triangles and quadrilaterals which are automatically shaped to accomplish high-quality criteria. Overall, the mesh counts 50k cells. It can be roughly estimated that this cell count allows performing an efficient problem space parallelization up to 4 logical processes in the computation.

As mentioned in Section II-F, OpenFOAM is very sensitive to the mesh quality, and a single low-quality cell can cause instability and simulation divergence. Pointwise
provides a very wide toolbox of functionalities giving the possibility to prescribe strict quality criteria to be fulfilled by the algorithm, and a very efficient quality check tool. The main checks generally to be performed for 2D meshes involve the skewness and the aspect ratio.

- The skewness is kept well below the suggested threshold value of 0.8 in the whole domain. Only a total of 7 cells in correspondence of the concave corners in the cavity between the two fans exceed the limit. Nevertheless, given the irrelevance of the flow in that cavity, the solution stability is not affected. To achieve this high-quality, given the expected irrelevance of the flow in the trailing edge region, the rear concave cusp has been cut as shown in Figure 6.
- The aspect ratio is kept below 10 in the whole domain, including the boundary layer cells.

Furthermore, as introduced in Section II-F, OpenFOAM has its peculiar rule of thumb to assess the mesh quality, i.e. the so-called 'cell non-orthogonality' should be kept desirably below 70. Its built-in tool 'checkMesh' allows to quickly verify this condition. The checkMesh performed on the final design provided a maximum non-orthonality of around 66, thus ultimately finding evidence of the mesh high-quality.

In most of the CFD studies, a best-practice to prove the simulation consistency and convergence is to perform a so-called grid-convergence check. It consists of monitoring the solution variation progressively decreasing the mesh $\Delta x$. In the studied internal flow, however, there is no computed solution coefficient to monitor the variation with $\Delta x$. Moreover, the sensitivity of OpenFOAM to instabilities ensures that an improper mesh will cause the simulation to diverge instead of providing artificial and non-realistic converged results. Hence, for the purposes of the present study, the computationally-consuming grid-convergence check can be avoided.

C. Numerical methods

As introduced in Section II-E, the temporal and spatial discretization of the governing equations determines the simulation convergence and the accuracy of its results. While higher-order schemes introduce a minor truncation error and are thus more precise, lower-order schemes are more dissipative and prevent instabilities to grow.

Considering the temporal discretization, two different transient integration methods have been used and compared. The first method consists of the default implicit-time second-order backward PISO solver with two corrector steps (Section II-F). However, the large simulation time caused by the very small $\Delta t$ (Section III-B) led to the search for a more efficient solver. The choice fell on the low-dissipative fourth-order explicit Runge-Kutta (RK) projection method published by Vuorinen et al. [23]. Mathematically, RK is considered one of the standard methods to solve transient problems of the form

$$\frac{dU}{dt} = f(t, U).$$

The classical fourth-order RK-method (RK4) consists of the solution $U^n$ progression to $U^{n+1}$ according to the following update sequence:

$$\begin{align*}
    k_1 &= f(t^n, U^n)\Delta t, \\
    k_2 &= f(t^n + \Delta t/2, U^n + k_1/2)\Delta t, \\
    k_3 &= f(t^n + \Delta t/2, U^n + k_2/2)\Delta t, \\
    k_4 &= f(t^n + \Delta t, U^n + k_3)\Delta t, \\
    U^{n+1} &= U^n + (k_1 + 2k_2 + 2k_3 + k_4)/6, \\
    t^{n+1} &= t^n + \Delta t.
\end{align*}$$

In contrast to the operator splitting and corrector loop in PISO (Section II-F), the new solver employs the projection method to couple pressure and velocity, as is often the case when dealing with explicit integration time methods. The method consists of the projection of the velocity field onto its solenoidal counterpart using the pressure gradient in a single projection step [23]. The final OpenFOAM RK-projection code has been implemented following the algorithm account, development and test provided by Vuorinen et al. [23]. Their study also shows an overall algorithm efficiency increase up to nearly 50% in comparison to PISO, whilst not detecting any significant instability growth due to the higher order of accuracy.

Considering spatial discretization, particular attention has been focused on divergence schemes. OpenFOAM provides a variety of options ranging from a highly-stable but highly-dissipative first-order scheme (Gauss upwind) and a more accurate but more unstable second-order scheme (Gauss linear) [24]. A quick test for the analysed flow case showed that the Gauss linear divergence scheme for the velocity field is too unstable and brings the simulation to divergence. In contrast, the Gauss upwind scheme quickly converged to steady conditions due to its high dissipation feature. Hence, a reasonable compromise should be found in order to achieve sufficiently accurate results whilst avoiding the simulation divergence. Following the OpenFOAM guide [24], a set of preliminary comparative simulations between divergence schemes has been performed. The blend Gauss linearUpwind scheme has been proved to be the most reasonable compromise, providing stable results whilst not suppressing the most relevant turbulent structures through numerical dissipation. This scheme is thus employed for the velocity, temperature and humidity field. For the rest of the fields, the tests showed that the Gauss linear scheme can be retained without generating.
stability issues. Considering the discretization schemes for the remainder spatial operators, some tests showed that they do not significantly affect the stability and the accuracy of the solution for the analysed case. Hence, the default setup has been kept considering a standard OpenFOAM tutorial for the PISO solver ("pitzDaily", [25]) and is reported in Table I.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient</td>
<td>Gauss linear</td>
</tr>
<tr>
<td>Laplacian</td>
<td>Gauss linear corrected</td>
</tr>
<tr>
<td>Interpolation</td>
<td>linear</td>
</tr>
<tr>
<td>Surface-normal gradient</td>
<td>corrected</td>
</tr>
</tbody>
</table>

Table I: Spatial discretization schemes employed for each operator.

D. Turbulence modelling

For the internal flow investigation of the present study, the two-equations \( k - \omega \) SST has been selected as the most indicated RANS-based turbulence model. In fact, the model is recognized as the most reasonably general and reliable for a wide class of flows [26]. Furthermore, this choice is in accordance to the UAV internal flow simulation provided by Panagiotou et al. [12].

The \( k - \omega \) SST model consists of solving a transport equation for the turbulent kinetic energy \( k \) and the turbulent frequency \( \omega \). The turbulent frequency is defined as

\[
\omega = \frac{\varepsilon}{C_{\mu} k},
\]

whereby \( \varepsilon \) is the so-called "turbulent dissipation rate" and \( C_{\mu} = 0.09 \) is a model constant. There follows the model expression for the Eddy Viscosity

\[
\nu_T = \frac{k}{\omega},
\]

which allows to close the system of Equations 2. A complete most-recent illustration of the model, including the transport equations for \( k \) and \( \omega \), can be found in Menter et al. [27].

E. Boundary conditions

In order to correctly define and solve the simulation problem, a realistic definition of the physical boundary conditions (BC) should be performed. For internal flow computations, it would be good-practice to extend the domain and introduce an artificial external co-flow in the proximity of the real boundaries in order to develop some numerical dissipation and make the solution more stable. However, the co-flow introduction in a transient and low-\( \Delta t \) simulation complicates the case setup significantly and substantially increases the computational expense. Hence, as the well-thought-out grid design and numerical setup of the present case allows the simulation to converge regardless of the co-flow, its addition is skipped. An overview of the final geometrical patches associated to each boundary condition name is shown in Figure 7.

1) Boundary velocity: The boundary conditions for the velocity field \( U \) for each patch are reported in Table II.

<table>
<thead>
<tr>
<th>Patch</th>
<th>BC type</th>
<th>( U ) uniform value [m/s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>inlet</td>
<td>fixedValue</td>
<td>(20, 0, 0)</td>
</tr>
<tr>
<td>outlet</td>
<td>zeroGradient</td>
<td>-</td>
</tr>
<tr>
<td>fuelcellstack</td>
<td>fixedValue</td>
<td>([6.42 \cdot 10^{-5}, 2.71 \cdot 10^{-5}])</td>
</tr>
<tr>
<td>fuelcellfans</td>
<td>fixedValue</td>
<td>([1.22 \cdot 10^{-5}, 5.14 \cdot 10^{-5}])</td>
</tr>
<tr>
<td>airfoil</td>
<td>noSlip</td>
<td>(0, 0, 0)</td>
</tr>
<tr>
<td>fuelcellhotwalls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fuelcellwalls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table II: Boundary conditions for \( U \).

While a noSlip condition has been prescribed at the walls, the fuel cell stack and fan velocities have been estimated considering the regime flow rate processed by the fuel cell of \( Q = 10 \text{ l/min} \) [7]. Their resulting value is orders of magnitude smaller than the primary flow. As discussed in Section II-D, this flow feature establishes unfavourable conditions for condensation, thus allowing to reasonably neglect its consideration in the model. Moreover, Table II shows that the inlet is assumed to consist of a uniform horizontal velocity of \( 20 \text{ m/s} \), i.e. equivalent to the external flow cruise speed. On one hand, this 'ideal' intake guess can be considered reasonable, as the inlet has been placed in very close proximity to the airfoil stagnation point. On the other hand, the real inlet velocity profile is not expected to be uniform and as large, because the very small intake dimension introduces consistent viscosity and blockage effects. Hence, the introduced flow rate in the simulation may be significantly over-estimated. Nevertheless, for the model development purpose of the present study, the inlet velocity BC is kept uniform to \( 20 \text{ m/s} \) in order to simplify the case setup. Further research on the inlet boundary condition is recommended if a future 3D intake-exhaust optimization study will be performed.

2) Boundary pressure: Considering the pressure field \( p \), a zeroGradient boundary condition is set up for all the patches except for the outlet. In fact, at the outlet a fixedValue condition should be prescribed in order for the problem to be well-posed [19]. In most of the external-flow outlets, a uniform 0 value is conventionally applied. In contrast, in this internal-flow case study, the exhaust is placed to take advantage of the suction enhanced by the low-pressure region on the upper surface of the airfoil, as discussed in Section III-A. Hence, a realistic negative
relative pressure value to prescribe should be estimated. The estimation is performed by means of a preliminary steady-state 3D external flow simulation over the model in cruise conditions. A contour plot of the resulting pressure field at the mid-cross-section is shown in Figure 8. The relative pressure found in the proximity of the exhaust position and prescribed as a uniform boundary condition for the internal flow is

\[ p_{out} = -144.3 \text{ Pa}. \]  

(18)

Similarly to the inlet-intake aforementioned (Section III-E1), this outlet condition is 'ideal' and is considered acceptable even if its uniformity and precision should be further assessed for a real flow case.

3) Boundary turbulence quantities: As described in Section III-D, the \( k - \omega \) turbulence model consists of solving a transport equation for \( k \) and for \( \omega \). Hence, a boundary value for these two parameters should be prescribed as well. The value \( k \) is defined as

\[ k = \frac{3}{2}(IU_\infty)^2; \]

\[ I = \frac{u_{rms}}{U_\infty}, \]

(19a)

(19b)

where \( u_{rms} \), is the flow characteristic turbulent velocity scale. A rough guess for the inflow \( I \) can be estimated and thus the boundary \( k \) can be derived. According to Wallin [28], in non-turbomachinery internal flows

\[ I \cong 0.3 \%, \]

(20)

and this boundary guess is not expected to affect the solution mean flow field significantly, given its small magnitude [28]. Considering the fuel cell inflow conditions, in contrast, it is reasonable to predict a larger turbulence intensity created by its internal flow path, i.e.

\[ I \cong 3 \%. \]

(21)

Nevertheless, the much smaller inflow speed (see Table II) will yield a very small inflow turbulent kinetic energy.

The inlet boundary \( \omega \) can be further obtained from Equation 17 given a reasonable \( \nu_T \). A good estimate for \( \nu_T \) can be again provided from the aforementioned external flow simulation (Section III-E2). In the intake proximity the computed parameter reads

\[ \nu_T \cong 0.2 \text{ m}^2/\text{s}. \]

(22)

On the other hand, to estimate the fuel cell inflow conditions, it is more convenient to introduce a turbulent length scale \( L \) which can be assumed around 10% of the geometrical length of the patch [28]. Given that

\[ L = \frac{k^{1/2}}{\varepsilon}, \]

(23)

the following expression for \( \omega \) can be derived from Equation 16:

\[ \omega = \frac{k^{1/2}}{C_r \mu L}. \]

(24)

The final turbulent boundary inflow conditions are summarised in Table III.

<table>
<thead>
<tr>
<th>Patch</th>
<th>BC type</th>
<th>( k \ [m^2/s^2] )</th>
<th>( \omega \ [1/s] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>inlet</td>
<td>fixedValue</td>
<td>( 5.4 \cdot 10^{-3} )</td>
<td>( 1.8 \cdot 10^{-9} )</td>
</tr>
<tr>
<td>fuelcellstack</td>
<td>fixedValue</td>
<td>( 9.9 \cdot 10^{-9} )</td>
<td>( 4.2 \cdot 10^{-2} )</td>
</tr>
<tr>
<td>fuelcellfans</td>
<td>fixedValue</td>
<td>( 3.6 \cdot 10^{-8} )</td>
<td>( 2.3 \cdot 10^{-4} )</td>
</tr>
</tbody>
</table>

Table III: Inflow boundary conditions for the turbulent parameters. All the reported values are uniform across the patch.

Furthermore, the implementation of the \( k - \omega \) SST model in OpenFOAM requires also a file prescribing the boundary \( \nu_T \). For the inflow patches, however, the condition is of type "calculated", i.e. the model computes \( \nu_T \) itself given \( k \) and \( \omega \) (Equation 17).

The boundary wall patches include all those where the noSlip condition has been prescribed in Table II. Given the uncertainty on the first cell thickness estimation provided in Section III-B, in order to safely resolve the near-wall flow region, a wall function should be set up at the boundary wall patches for \( k \), \( \omega \) and \( \nu_T \). The default wall functions of the most popular PISO tutorials (pitzDaily) have been retained and are reported in Table IV.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>BC type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k \ [m^2/s^2] )</td>
<td>kqRWallFunction</td>
<td>( 5.4 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>( \omega \ [1/s] )</td>
<td>omegaWallFunction</td>
<td>( 3.2 \cdot 10^{-4} )</td>
</tr>
<tr>
<td>( \nu_T \ [m^2/s] )</td>
<td>nutkWallFunction</td>
<td>0</td>
</tr>
</tbody>
</table>

Table IV: Wall boundary conditions for the turbulent parameters. All the reported values are uniform across the patch.

Even if both \( \nu_T = 0 \) and \( k = 0 \) at the walls, in the very near-wall region \( k \) increases suddenly. Hence, it is suggested to initialize the wall boundary value with the same \( k \) as the inlet [29]. In addition, the following expression dependent on the grid first cell thickness \( h_1 \) is recommended for the wall boundary \( \omega \) initialization:

\[ \omega_{wall} = 10 \cdot \frac{6 \nu}{\beta_1 h_1}, \quad \beta_1 = 0.075 \ [30]. \]

(25)
Further, during the simulation, \( k \) and \( \omega \) are solved across the iterations by means of the wall function and their wall value is updated.

At last, \( k \) and \( \omega \) BCs are set to type "zeroGradient" at the outlet similarly to \( U \), while the outlet \( \nu_T \) is set to type "calculated" and its value is thus computed from \( k \) and \( \omega \) across the simulation.

4) Boundary temperature: The intake supplies the internal bay with external air. Hence, as the Green Raven flies in cruise conditions close to the sea level, a standard \( T = 20 ^\circ C \) has been prescribed at the inlet. Furthermore, according to the fuel cell manual \([7]\), the operating stack temperature is set to \( T = 60 ^\circ C \).

The fuel cell has an operating efficiency of 40\% \([7]\). This means that, known the fuel cell rated power of 1000 W, approximately 600 W are dispersed into heat. This thermal power is introduced inside the internal bay through the fuel cell hot walls and should therefore be accounted for in the model. In order to provide a rough estimate for a thermal flow boundary condition, this power should be divided by the surface of the hot walls. In OpenFOAM, this thermal flow condition is prescribed through the "fixedGradient" BC type.

In the reminder of the patches, a "zeroGradient" BC is set up. A summary of the boundary conditions for \( T \) is reported in Table V.

<table>
<thead>
<tr>
<th>Patch</th>
<th>BC type</th>
<th>BC value</th>
</tr>
</thead>
<tbody>
<tr>
<td>inlet</td>
<td>fixedValue</td>
<td>293.15 K</td>
</tr>
<tr>
<td>fuelcellstack</td>
<td>fixedValue</td>
<td>333.15 K</td>
</tr>
<tr>
<td>fuelcellhotwalls</td>
<td>fixedGradient</td>
<td>3800 W/m²</td>
</tr>
<tr>
<td>airfoil</td>
<td>zeroGradient</td>
<td></td>
</tr>
<tr>
<td>fuelcellfans</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fuelcellwalls outlet</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table V: Boundary conditions for \( T \). All the reported values are uniform across the patch.

5) Boundary humidity: The only source of moist air in the domain is the fuel cell stack. Given the stack temperature \( T = 60 ^\circ C \), the equilibrium water vapour pressure \( p_0 \) can be found with Antoine’s equation (see Section II-D). Hence, considering the atmospheric pressure \( p = 1 \text{ atm} \), the mixing ratio \( r \) can be found by means of Equation 8. Finally, the specific humidity \( q \) is found from Equation 7 and can be prescribed as a "fixedValue" BC at the fuel cell stack:

\[
q = 0.13155 \text{ kg/kg.} \tag{26}
\]

In these calculations, particular attention should be focused on the units of measurement. For the remainder of the patches, a "zeroGradient" type condition is again set up.

**F. Initial conditions and flow properties**

The remainder of the flow properties and initial conditions for the internal flow initialization are reported in Table VI.

### Table VI: Flow properties and initial conditions.

<table>
<thead>
<tr>
<th>Property</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viscosity</td>
<td>( \nu ) [m²/s]</td>
<td>1 \cdot 10^{-5}</td>
</tr>
<tr>
<td>Velocity</td>
<td>( U ) [m/s]</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>Pressure</td>
<td>( p ) [Pa]</td>
<td>0</td>
</tr>
<tr>
<td>Temperature</td>
<td>( T ) [K]</td>
<td>293.15</td>
</tr>
<tr>
<td>Humidity</td>
<td>( q ) [kg/kg]</td>
<td>0</td>
</tr>
<tr>
<td>Turbulent kinetic energy</td>
<td>( k ) [m²/s²]</td>
<td>5.4 \cdot 10^{-3}</td>
</tr>
<tr>
<td>Turbulent frequency</td>
<td>( \omega ) [1/s]</td>
<td>1.8 \cdot 10^{4}</td>
</tr>
<tr>
<td>Turbulent viscosity</td>
<td>( \nu_T ) [m²/s]</td>
<td>Calculated</td>
</tr>
<tr>
<td>Time step</td>
<td>( \Delta t ) [s]</td>
<td>2.5 \cdot 10^{-6}</td>
</tr>
<tr>
<td>Air thermal diffusivity</td>
<td>( D_T ) [m²/s]</td>
<td>2.82 \cdot 10^{-4}</td>
</tr>
<tr>
<td>Water vapour diffusivity</td>
<td>( D_q ) [m²/s]</td>
<td>2.86 \cdot 10^{-4}</td>
</tr>
</tbody>
</table>

As \( Pr_T = 1 \), the above \( \nu_T \) value corresponds to \( \alpha_T \). To obtain \( D_T \), this contribution should be summed to \( \alpha = 2.17 \cdot 10^{-5} \) calculated at \( T = 20 ^\circ C \) with Equation 5. It can be immediately observed that the turbulent contribution is predominant and defines the order of magnitude of \( D_T \). Hence, even if the variation of \( \alpha \) had been accounted according to the temperature distribution as objected in Section II-C, the total thermal diffusivity and thus the scalar transport would not have been affected significantly. Physically, a predominant \( \alpha_T \) over \( \alpha \) means that the scalar transport follows the turbulence.

Considering the water vapour diffusion in air \( D_q \), according to Equation 11, to estimate its turbulent contribution \( D_{qt} \) it is necessary to guess a reasonable turbulent Schmidt number \( Sc_T \). The Lewis number is defined to relate the transport of any scalar to the transport of
temperature as $Le = Pr/Sc$. Using Reynolds analogy, a turbulent Lewis number can be defined as

$$Le_T = \frac{Pr_T}{Sc_T}. \quad (29)$$

Since it is physically reasonable to assume that the transport of water vapour in air will follow the turbulence behaving similarly to the temperature diffusion, $Le_T = 1$ is assumed, hence leading to

$$Sc_T = 1. \quad (30)$$

Furthermore, Goldman et al. [33] achieve an estimation close to $Sc_T = 1$ considering the transport of similar gases in air, thus raising confidence in the assumption validity. As $Sc_T = 1$, the magnitude of $D_{qr}$ will be equal to the one of $\nu_T$ reported in Equation 28. To obtain $D_q$, this contribution should be summed to $D_{qr} = 2.57 \cdot 10^{-5}$ calculated from Equation 10 given $T = 20^\circ C$. Once again, the turbulent contribution is predominant and defines the order of magnitude of $D_q$, thus confirming that the water vapour diffusion will follow the turbulence. Moreover, as similarly discussed for $\alpha$, accounting for a variable $D_{qr}$ with $T$ would not have significantly affected the total $D_q$ and thus the scalar transport behaviour.

IV. Results

A. Quality

1) Convergence and computational cost: In light of the very careful simulation setup discussed in Section III, both the RK4 and the PISO simulation converged to low residuals without displaying any instability issue. In fact, the maximum Courant number safely oscillates in the proximity of 1, thus fulfilling the condition for stability according to Equation 12. On the other hand, due to the aforementioned very small $\Delta t$, both the solvers required a computational time of the order of magnitude of days to display appreciable transient flow features. The space parallelization gave a significant contribution to speed up the iterations performed at each time step, but the limited spatial extent of the problem makes ineffective any further increase of the number of processes. Furthermore, even if RK4 improves the performance at each iteration in comparison to PISO, it is not itself sufficient to significantly scale down the simulation time expense. On one hand, as discussed in Section III-A and III-B, the particular needs dictated by the model development process and the limited amount of simulations necessary have made this high computational cost acceptable. In fact, a limited physical time extension of $t = 10$ s has been proved to be sufficient to achieve relevant results to draw significant conclusions on the validity of the models. On the other hand, in a future 3D optimization project the problem can be addressed with a more effective space parallelization and a mesh allowing a larger $\Delta t$.

2) Steady state: Apart from a few flow regions where some intrinsically unsteady vortex structures keep developing in time, the PISO simulation achieves an overall steady state after around ten seconds of physical time for all its solution fields. As shown in the lower part of Figure 9, the primary flow path is stabilized surrounding the fuel cell. In contrast, the flow solved with RK4 shows a solution dominated by turbulent swirling motions which do not display any steadiness trend (upper Figure 9). This discrepancy can be explained by the higher dissipation performed by the lower-order scheme (second order in time) implemented in PISO. A more dissipative solver is more likely to achieve steady state, whilst introducing a larger truncation error and thus trading some reliability of the solution. As a matter of fact, the more unsteady solution provided by the higher-order RK4 solver looks more realistic considering the geometrical complexity of the investigated internal flow.

3) $y^+$ check: As discussed in Section III-B, a $y^+$ check is necessary to assess the quality of the estimation of $h_1$. The solutions show that the $y^+$ is kept below 5 in the majority the domain, i.e. the first cell fully encloses the viscous sub-layer. However, a small region exceeds the threshold in the proximity of the intake, where the local velocity magnitude is still large. Nevertheless, the primary flow features are mainly determined by the free-shear vortices rather than by the wall-bounded flow. Moreover, the wall functions introduced in Section III-E3 ensure an acceptable resolution of the higher $y^+$ zones. Therefore, the estimated $h_1 = 0.5 \text{ mm}$ can be considered acceptable.

4) Inflow boundary conditions: As it will be later discussed in Section IV-B and shown in Figure 10, the solutions suggest that fuel cell stack and fan patches are lapped by vortical structures. The inflow "fixedValue" boundary conditions prescribed in their correspondence artificially prevents these vortices to interact with the surfaces, i.e.
they are precluded to exit from the domain through their patches. However, given the very low suction and blowing boundary speed prescribed (see Table II), this is something which is likely to happen in a real case considering the high energy carried by the vortices. Nevertheless, according to the fuel cell manual [7], the real stack consists of a grid whose distributed holes act as internal flow exhausts into the internal UAV bay. Hence, the remainder of the surface is in all respects a wall which actually prevents the vortices from entering into the fuel cell interior. Similarly, the fans are covered by a protective grate. Therefore, from the vortices perspective, the chosen boundary inflow conditions can be considered to model as close as possible the real setup. On the other hand, the presence of the aforementioned grids raises a question mark on the validity of the turbulent inflow conditions. In fact, as a comparison of Tables III and IV highlights, the very low speed set at the stack and fans leads to a substantial discrepancy between their inflow turbulent boundary values and the ones prescribed at the walls. In a real case, in contrast, the lattice-shaped presence of a wall above their interfaces is expected to generate larger turbulent boundary values than the ones reported in Table III.

B. Velocity field

In order to achieve a better understanding of the flow velocity field, the colour bar scaling has been changed to logarithmic and the resulting contour plots are shown in Figure 10. It can be immediately observed that the solution differs substantially between the two solvers. On one hand, RK4 shows that the flow is directed upwards downstream of the inlet, lapping the internal airfoil-shaped upper surface. Hence, the flow diffuses above the fuel cell stack displaying appreciable large-scale turbulent structures on top of it. Finally, the primary flow is either pulled out by the low-pressure outlet or diffused further downstream above the tank or in the area between the tank and the fuel cell. In this last case, it runs in the proximity of the fans and thus rejoins the primary flow originated at the inlet. Overall, as mentioned in Section IV-A2, the solution is unsteady since the vortical entities evolve across the physical time.

On the other hand, the PISO solution shows a flow which is directed downwards downstream of the inlet. After passing the fuel cell fans, it is pulled by the low-pressure outlet from the bottom to the top. Hence, most of the flow laps the airfoil-shaped upper surface in the proximity of the stack moving towards the leading edge. An overall steady state is achieved where the primary flow surrounds the fuel cell in a counterclockwise direction. In contrast to RK4, there is less flow transfer towards the trailing edge. Moreover, the solution field displays a "bubble" of stagnation above the stack and the right hot wall. In RK4, conversely, the flow laps their surfaces with turbulent structures. The reason for this discrepancy probably lies in the higher dissipation performed by the PISO solver, which tends to suppress the turbulent structures. As it will be proved in Section IV-C and IV-D, the presence of this stagnation bubble will not be favourable for the scalar transport.

From a preliminary assessment, it already seems that the RK4 higher-order solution is closer to what one could intuitively expect. In fact, the flow travels across the shortest path from the inlet to the outlet. Moreover, it displays more relevant turbulent structures which are expected to be a constitutive feature in an internal flow with the given geometrical complexity.

C. Temperature field

The contour plot of the temperature field after ten seconds is shown in Figure 11. At first glance, it is imme-
diately clear that the suppression of the turbulent structures observed in the PISO solution (Section IV-B) leads to an overall more consistent heating. In fact, the flow counterclockwise circulation around the fuel cell creates a closed pattern where the air increases its temperature progressively in time under the influence of the fuel cell hot walls and stack. As a consequence of this inefficient cooling flow pattern, once a steady thermal state is achieved, the overall temperature observed in the region around the fuel cell is higher than the one computed by RK4.

Furthermore, PISO’s bubble of stagnation discussed in Section IV-B results in a very high-temperature zone in its correspondence. Hence, in the proximity of the fuel cell stack and the right hot wall, there is a critical overheating region which, if realistic, could lead to the fuel cell overheating and shut off. On the other hand, the turbulent structures developed in the solution computed by RK4 around the fuel cell lead to a much more efficient cooling flow, confirmed by the lower temperature region shown in the upper Figure 11.

With the goal of correctly predicting the transport of scalars, these results point out the substantial importance of an accurate computation of the turbulent structures in the model solution. Therefore, given the lower-order and turbulence-suppression features of PISO, it can be concluded that the RK4 model appears more reliable and suitable for the purposes of the investigated case.

D. Humidity field

The contour plot of the humidity field after ten seconds is shown in Figure 12. As introduced in Section II-D, the computed value of $q$ is not exact in the whole domain because its variation with $T$ has not been accounted. Nonetheless, relevant conclusions on the validity of the models can be drawn evaluating the overall water vapour distribution across the domain, which is an output of a reliable scalar transport advection-diffusion simulation.

Similarly to the temperature field, it is immediately visible that PISO’s lack of turbulent structures and recirculation flow pattern lead to a consistent humidity stagnation around the fuel cell up to the airfoil-shaped walls. In addition, in correspondence of the bubble of low speed above the stack, a region of water vapour concentration close to saturation is observed. Under these conditions, the non-condensation assumption discussed in Section II-D is less robust and most likely non-realistic.

Considering the solution of RK4, in contrast, the high vapour concentration region near the stack is much more restricted, and the humidity drops moving away from its patch. In fact, the high-momentum primary flow invests the stack with its turbulent structures and entrains the water vapour, which is transported either towards the outlet or further downstream above the tank or below the fuel cell. Hence, the computed scalar concentration around the fuel cell is much more limited indicating an overall more efficient ventilation. These considerations suggest that the resident time of the water vapour particles inside the domain will be short, thus raising confidence in the non-condensation assumption. The only stagnation observed concerns the region above the tank, but a more optimized future 3D intake-exhaust system should be designed to avoid their occurrence, thus being suitable to be safely simulated with the RK4 model developed.

Furthermore, comparing the $T$ and $q$ fields computed by RK4 (upper Figures 11 and 12), a very close similarity in their distribution is observed, particularly on the top of the stack. The reason lies in the fact that the diffusivity of the two scalars is very similar, as its predominant contribution is given by the turbulence (Section III-F).

V. Model improvements: the density variation and the gravity term

Having developed the model and methodology and analysed the results, there are two remaining assumptions which can still represent a weakness in the model definition.

1) As stated in Section II-C and II-D, the influence of $T$ and $q$ on $\rho$ in the momentum Equation 2b is neglected;

2) as mentioned in Section II-A, the gravity term in the same momentum equation has been ignored.

It is very difficult to provide any suggestion to cope with the former hypothesis without completely turning the model upside down and increasing consistently its complexity. Nonetheless, the RK4 solution shows that above the stack where a higher fixedValue scalar boundary condition is prescribed, both $T$ and $q$ tend to be quickly restored to the primary flow values (upper Figures 11 and 12). Therefore, it is reasonable to expect a limited density variation under the influence of the stack scalar BCs, which would not affect the output mean flow and the scalar transport significantly. Of course, this solution
feature will depend on the flow rate of cooling air which is introduced in the domain and invests the stack surface. Nevertheless, considering the goal of applying the model to develop an optimized 3D intake-exhaust configuration, such flow rate should be large enough in any efficient design to test. Hence, from this perspective, the RK4 model can be concluded to be acceptably reliable for the intended applications.

Considering the second assumption listed above, the gravity term neglected on the right-hand side of Equation 2b reads

\[ \frac{\dot{\rho}}{\rho}g_i, \]  

(31)

where \( g_i \) is the gravity acceleration vector and \( \dot{\rho} \) is the density adjusted with the contribution of the scalars \( T \) and \( q \). From the inspection of the results obtained in Section IV, it is not possible to predict whether the temperature and humidity gradients will lead to density variations enough significant to originate buoyancy forces, for whose consideration the gravity term addition is necessary. Given this uncertainty and since it is in principle possible to account for the gravity term in the model with a reasonable effort, an implementation attempt is made, with the ultimate goal of evaluating the influence of the buoyancy forces on the solution.

The most simple implementation consists of employing the Boussinesq hypothesis, according to which the density is assumed to vary linearly with the temperature. A similar consideration can be drawn for the density dependence on the specific humidity. Hence, Equation 31 can be rewritten as

\[ [1 - \beta_T(T - T_{ref}) - \beta_q q]g_i. \]  

(32)

The higher is the temperature range, the more inaccurate the hypothesis will be. Nevertheless, in the investigated case, the solution’s \( \Delta T_{max} \) is reasonably limited (Figure 11, Section IV-C), so the assumption can be considered acceptable with the goal of assessing the contribution of the buoyancy forces.

A PISO-based solver implementing the Boussinesq hypothesis for the temperature called "buoyantBoussinesqPisoFoam" is available in OpenFOAM [34], and a detailed description of the equations and the solution procedure can be found in OpenFOAMWiki [35]. Further, the model source code can be easily modified to account for the contribution of \( q \) according to Equation 32 and the new solver has been named "doubleDiffusionPisoFoam". For the setup of a simulation test, the expansion coefficients \( \beta_T \) (thermal expansion coefficient) and \( \beta_q \) have been estimated from a reference temperature of \( T_{ref} = 20^\circ C \) and are reported in Table VII.

| \( \beta_T \)  | \( 3.4 \cdot 10^{-5} \) |
| \( \beta_q \)   | \( 1.45 \cdot 10^{-4} \) |

Table VII: Expansion coefficients.

The results of the model run on the investigated case could be in principle compared with the ones of PISO reported in Section IV. However, surprisingly the case resolution with the new solver encountered convergence problems. After an inspection of the diverged partial results, the spatial origin of the divergence has been found in the correspondence of the fuel cell lower-right corner, as shown in Figure 13. Given that the setup is exactly the same as the previous case converged with PISO, it can be concluded that the gravity term addition increases significantly the numerical stiffness of the problem. Hence, even further attention should be focused on the numerical setup and the mesh quality. Considering the former, some attempts to introduce more dissipative space discretization schemes revealed to be insufficient to reach the simulation convergence. Therefore, the focus has been shifted to the mesh. By means of the examination tool of Pointwise, a critical area ratio above 10 has been detected in correspondence of the diverged cells, as shown in Figure 14. Even if the area ratio is not generally pointed as one of the most crucial quality parameters, in the presence of the gravity term, it turned to be itself sufficient to cause the simulation divergence. Therefore, it is recommended to construct an improved mesh, whose design should get rid of high-area ratio cells, and to test and assess its convergence. Once convergence is achieved, it will be possible to draw
conclusions on the influence of the gravity term on the solution of the PISO-based solver. Nonetheless, in order to allow a consistent comparison, it will be necessary to run the new mesh also with the original PISO solver, thus resulting in a very time-consuming task.

Given the limited reliability demonstrated by the PISO model in the scalar transport and the extensive time requested to effectively get it running, it is rather worth trying to implement the Boussinesq hypothesis and the gravity term in the more advanced RK4 model. In fact, the comparison of the solutions provided by RK4 and RK4+gravity is expected to be more realistic and to allow to draw more robust conclusions on the importance of the gravity term. Moreover, considering the completely different resolution method, it is possible that the gravity term will not increase the numerical stiffness of the problem enough significantly to cause the simulation divergence. Thus, the prospective RK4+gravity solver may in principle work with the original mesh without requiring any cell quality improvement. Following the guidance of the buoyantBoussinesqPisoFoam solver [35], the addition of the gravity term in the governing equations has been easily implemented successfully. However, the development of the model failed while dealing with the pressure correction step. In fact, the guidance of the implicit-time buoyant-BoussinesqPisoFoam code [34] could no longer be followed, as the explicit-time RK4 solver employs the projection method to couple pressure and velocity rather than the operator splitting and corrector loops. Several implementation attempts have been tested on the most simple case studies provided by OpenFOAM’s heat transfer tutorials [36], but none of them succeeded. As the development of a double-diffusion RK-based solver represents a research interest beyond the investigated UAV case, in the future it is worth exerting a further effort to implement and test a working code, perhaps with the help of expert OpenFOAM programmers and researchers in numerical methods.

VI. Conclusions

In conclusion, a complete and successful CFD model has been developed, tested and discussed. The RK4 solver has been proved to be the most reliable choice with the goal of analysing the internal flow of a hydrogen-powered UAV, where the diffusion of both temperature and water vapour should be taken into account. However, so far it has not been possible to understand whether or not the gravity term consideration effectively influences the solution. Together with the development of a working RK4+gravity solver, this topic will be object of future studies. Nevertheless, the examples in literature (e.g. Panagiotou et al. [12]) show that the internal flows are primarily dominated by the advection and diffusion terms of the momentum equation. Therefore, the RK4 solver can be considered ready to be employed for the design of an optimal UAV intake-exhaust 3D system.

Furthermore, the CFD methodology employed for the investigated sample test case can be safely used as a base for the setup of more complex prospective 3D optimization cases. However, there are still some additional or uncertain aspects which are recommended to be considered in future applications.

- The inflow turbulent boundary conditions at the fuel cell stack and fans are recommended to be further investigated. In fact, the parameters $k$ and $\omega$ prescribed introduce very weak turbulence at the boundaries, which could be non-realistic considering the actual grid-shaped geometries of the patches. Nevertheless, in an optimized intake-exhaust configuration, they are not expected to significantly influence the solution.
- In order to effectively reduce the simulation time, the 3D inlet and outlet patches should be meshed with a sufficiently large $\Delta x$ to allow a reasonably large $\Delta t$, whilst fulfilling the CFL condition. This $\Delta x$ feature can be achieved only if the intakes and exhausts designed are themselves not as small as the ones investigated in this study, which is often the case in the examples found in literature (e.g. Panagiotu et al. [12] and Chiang et al. [11]).
- To achieve overall higher-quality boundary conditions at the intakes and the exhausts, it is recommended to combine the internal and external flow in one single simulation. However, running a large external flow domain with a small-$\Delta t$ transient simulation would be highly time-consuming and inefficient. A possible way to cope with this issue would be to first build a mesh for the external flow domain, where the internal flow inlet and outlet patches are already prepared. At these patches, a wall BC should be initially prescribed, so that a steady-state simulation can be performed on the external domain within a reasonable computational time. Thus, after combining the previous external mesh with the internal one, the patches can be converted into free exterior-interior interfaces, and a new global transient simulation can be set up, using the earlier results as initial conditions for the external domain.
- So far, the influence of the propeller has not been considered. Nonetheless, the intake-exhaust system can potentially take advantage of the flow attraction provided by the Green Raven’s pusher propeller, as similarly shown in the configuration provided by Kim et al. [9]. The influence of the propeller on the external flow can possibly be modelled as a momentum source disk, as suggested by Panagiotou et al. [12].

Acknowledgements

A heartfelt thanks goes to the Green Raven project coordinator, prof. Raffaello Mariani, who involved me in the team and in correlated research projects. His ability to recognize students potential and to informally and respectfully relate with them has been very stimulating. I am looking forward to continuing our research collaboration in the future.

A very special thanks goes to my supervisor Luca Brandt for his unconditional support. Thank you also for
coordinating the help provided by all the members of his research team. Last but not least, thank you for involving me in restoring football matches.

I would also like to express my heartfelt gratitude to dr. Matteo Montecchia for patiently supporting me with the case setup and the solver development, sharing with me all his crucial extensive experience with OpenFOAM. Thanks also to Nicolò Scapin for teaching me the fundamentals of moist air flows and for reviewing the drafts of the related sections. Thank you to all KTH researchers of the 6th floor in the Mechanics department for allowing me to breathe a research enthusiast environment.

Another special thanks goes to my company supervisor Faranggis Bagheri for being confident in my capabilities and for her friendly management in these two wonderful years at neptech AB. Thank you also for sharing with me the experience with Pointwise and for allowing me to use the software for this master thesis.

Thank you to all KTH researchers of the 6th floor for sharing with me this amazing experience and for her friendly management in these two wonderful years at neptech AB. Thank you also for sharing with me the experience with Pointwise and for allowing me to use the software for this master thesis.

Last but not least, I would like to thank all my friends at KTH for sharing with me this amazing experience and for the inspiring conversations we had talking about our future.

References


APPENDIX A

ADDITIONAL CONTOUR PLOTS TO ACHIEVE A BETTER FLOW PHYSICS UNDERSTANDING

Figure 15: Shaded contour plot of the complete velocity field at $t = 10$ s computed with RK4 (upper) and PISO (lower).

Figure 16: Logarithmic contour plot of the complete velocity field at $t = 10$ s computed with RK4 (upper) and PISO (lower).
Figure 17: Logarithmic contour plot of the complete temperature field at $t = 10$ s computed with RK4 (upper) and PISO (lower).

Figure 18: Logarithmic contour plot of the complete humidity field at $t = 10$ s computed with RK4 (upper) and PISO (lower).