CFD Simulation and 3D Visualization on Cultural Heritage sites: The Castle of Mytilene

Orfeas-Theodoros Eleftheriou[1,3][0000−0002−0672−9869], Radostin Mitkov[2][0000−0002−2129−9860], Vasilis Naserentin[3,4][0000−0002−3485−9329], and Christos-Nikolaos Anagnostopoulos[1][0000−0002−4496−275X]

[1] University of Aegean, Social Science School, Cultural Technology and Communication Dpt, Mytilene, Greece
ct21002@ct.aegean.gr
canag@ct.aegean.gr
[2] GATE Institute, Sofia University "St. Kliment Ohridski", Bulgaria
radostin.mitkov@gate-ai.eu
[3] Chalmers University of Technology, Gothenburg, Sweden
vasilis.naserentin@chalmers.se
[4] Aristotle University of Thessaloniki, Thessaloniki, Greece

Abstract. This paper presents a CFD and 3D Visualization pipeline to simulate wind flow over a heritage site and then visualize the results. As case study, a coarse 3D geometry model of the Fortress of Mytilene, Lesvos island, Greece and its surrounding was generated from open access Digital Elevation Models. The CFD simulation of the air flow on the wider heritage site area was performed using the steady-state version of the in-house flow solver IBOFlow®(Immersed Boundary Octree Flow Solver) developed at Fraunhofer-Chalmers Research Centre. All the simulations were completed considering the mean wind direction and wind speed during the last 22 years from actual weather data retrieved from Open Weather Map. The visualization results were achieved through Unreal Engine, using built-in visualization tools and a tailored-made plugin to visualize the air flow over the monument and on the monument's wall. As discussed in the conclusion section, the overall process proposed in this paper can be implemented for an initial assessment of the effect of environmental parameters over any heritage site and moreover, it may form the basis for valuable assistive tool for conservators and engineers.

Keywords: Computational fluid dynamics · 3D modelling · Cultural Heritage.

1 Introduction

1.1 Computational Fluid Dynamics

Computational fluid dynamics (CFD) is the analysis and investigation of problems involving fluid flow by means of numerical approaches [1]. Some advantages of CFD over experimental-based approaches include more detailed information

on relevant flow variables, better controlled environment and boundary conditions, and reduced cost on the evaluation of new designs [2]. Due to these reasons, along with the advancements of computer technologies in the past few decades, CFD has seen a surge in industrial and research applications and has enabled engineers and researchers acquire a deeper understanding of numerous problems [3]. Example application areas include automobile and aircraft aerodynamics, ship hydrodynamics, turbomachinery, biomedical engineering, external and internal building environment studies. It is important to note, however, that CFD should be used to complement the other approaches of theoretical and experimental fluid dynamics and not replace them [4].

1.2 CFD for Cultural Heritage (CH)

In Cultural Heritage (CH), CFD methods are mostly used to simulate the behaviour of air flow around cultural sites. This is due to the fact that, in our physical environment, the air is the carrier and distributor of other measurable substances (i.e. pollutants, particles, dust, salt), it interrelates with environmental parameters (i.e temperature, relative humidity) and it even strains and/or wears directly the monuments. Along with air flow patterns, usually the temperature and relatively humidity distribution are also calculated, since the latter have a crucial role in the deterioration of heritage materials. An early research of CFD air flow simulation on CH was reported in [5] more than a decade ago. This research provided an efficient framework to simulate the wind environment over large scale heritage sites investigating the wind impact to the overall area of the Giza Pyramids including the Great Sphinx monument. The authors focused on estimating the wind pressure and surface friction distribution causing sand wearing on the monuments' external surfaces. A case of use of 3D models with CFD simulation is reported in [6]. D'Agostino et al. observed the Crypt of the Cathedral of Lecce in Italy for a period of one year and developed a digital model in which a simulation of thermo-hygrometric parameters and air flow patterns was applied to reproduce the crystallization of salts and the deterioration of works of art. The simulations aimed to propose an appropriate management scenario to decrease risky microclimatic conditions. In [7] the authors investigated the effect of wind erosion in the Yongling Mausoleum in China. In their experiments a 3D model of the heritage site was produced and by the application of CFD, they predicted the erosion due to wind in each monument of the Mausoleum complex. Another research for heritage site preservation was demonstrated in [8]. The authors produced a 3D mesh for the heritage site in Baelo Claudia in Spain and analyzed the sand-loaded air flow erosion using CFD. After the collection of the necessary environmental information, they predicted the deterioration of the site for the next 50 and 100 years. The goal of our paper is to provide an initial framework for the simulation and visualization of wind flow over historical and cultural heritage places using CFD and Unreal Engine, presenting as a case study, a coarse 3D model of the Fortress of Mytilene and its surrounding. The remaining paper is structured as follows. Section 2 describes briefly historical issues of the castle, the acquisition of environmental data, as

well as the mathematical background and methods of the CFD pipeline. Section 3 outlines the Unreal Engine visualization results of the air flow over the site and on the fortifications of the castle, while the paper ends with a conclusion section.

2 Materials and Methods

2.1 Heritage site description

The Castle (or Fortress) of Mytilene in Lesvos Island is located around a hill between the city's northern and southern ports (Latitude 39° 1'1 N, Longitude 26° 5'6 E). The first phase of the monument construction was completed during the Byzantine period (5th century AC) over the ruins of the ancient Acropolis and a sanctuary dedicated to Demeter, Kore and Cybele. The second construction phase of the castle took place during the period of the domination of the ruler Francisco Gatillusio in 14th century. Nowadays, it is considered as one of the largest and well-preserved castles of Greece that expands in an area of 200,000 sq. meters. According to its configuration, it is divided into three sections, the Upper, the Middle and Lower Castle (see Fig. 1).

Fig. 1. The Castle of Mytilene (Google Maps).

2.2 3D geometry modeling and mesh generation

To prepare the data for the simulations, an accurate, watertight geometry was generated. The geometry included the castle and the ground in as much detail as

possible. To build the ground mesh (terrain) in a resolution of 20m height for the area of interest, an open access Digital Elevation Model (DEM) from Copernicus [9] was used. With the retrieved data it was possible to build a heightmap texture which could be used as a starting point for the terrain's mesh. However, the texture's resolution was quite low which resulted in jagged edges in some cases on the ground level. To mitigate this issue, BigJPG AI image upscaler [10] was used to create a detailed and closer to real life looking texture.

In order to generate the castle's mesh, a topdown DEM view of the middle and upper parts of the castle [9] was layered in the heightmap from the previous step. Afterwards, a 2D mesh was generated which snapped along the edges of the topdown view of the castle. To match the data from the DEM view, the walls of the 2D mesh had to be extruded upwards. By combining the two resulting geometries, a 3D digital model of the middle and upper part of the Fortress of Mytilene was built, which can be used in CFD simulations.

2.3 Environmental data acquisition

To prepare an accurate simulation, real environmental data of the location of interest were retrieved. Specifically, it included hourly values of wind direction and wind speed from Open Weather Map [11] for each day over the span of 22 years, from January 1st 2000 until July 5 of 2022. The measurements came from the town's airport METAR station. The wind sensor is located on a 10m height. The indications, as well as the sensors height, are required to achieve accurate initial conditions for the simulation. Once the required data were retrieved, a python script was developed which extracted the mean values for the wind speed. By observing the retrieved values, it was determined that the mean wind speed was 3.42 m/s, which is equivalent to a moderate breeze while the mean wind degrees was roughly 155°which corresponds to South East direction from a wind rose.

2.4 Flow solver

For the purposes of the simulations in this paper, the steady-state version of the in-house flow solver IBOFlow®(Immersed Boundary Octree Flow Solver)[12], developed at Fraunhofer-Chalmers Research Centre, was used. The solver, procedure and settings are continuously being validated [12–14]. The solver assumes an incompressible, isothermal flow and integrates the Reynolds Averaged Navier-Stokes (RANS) equations:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\frac{\nabla p}{\rho} + \nabla \cdot ((\nu + \nu_t)\nabla \boldsymbol{u}), \tag{2}$$

along with the turbulent transport equations. Here $\boldsymbol{u}$ denotes the mean velocity of the fluid, ρ is the fluid density, p is the dynamic pressure, ν is the kinematic

viscosity of the fluid and ν_t is the eddy viscosity. Further, the immersed boundary (IB) condition is solved,

$$\boldsymbol{u} = \boldsymbol{u^{ib}}, \tag{3}$$

which constrains the velocity of the fluid to the velocity at the immersed boundary, $\boldsymbol{u^{ib}}$. This is achieved by through the implementation of the second-order accurate mirroring immersed boundary method which accounts for the presence of all geometries and eliminates the need of a body-fitted mesh [13].
IBOFlow®employs the finite-volume method to discretise all equations on a Cartesian octree grid that can be automatically refined around geometries. The SIMPLEC algorithm [15] is used for pressure-velocity coupling while the pressure-weighted flux interpolation [16] prevents the decoupling of the pressure and velocity fields. Finally, artificial time stepping is employed and the simulation concludes once all relative matrix residuals are converged.
In this paper, the Spalart-Allmaras (S-A) turbulence model is used. The eddy viscosity is given by:

$$\nu_t = \tilde{\nu} f_{\nu 1}, \qquad f_{\nu 1} = \frac{\chi^3}{\chi^3 + c_{\nu 1}^3}, \qquad \chi \equiv \frac{\tilde{\nu}}{\nu} \tag{4}$$

where $c_{\nu 1}$ is a model constant and $\tilde{\nu}$ is the Spalart–Allmaras (S-A) working variable and adheres to the following transport equation:

$$\frac{\partial \tilde{\nu}}{\partial t} + \boldsymbol{u} \cdot \nabla \tilde{\nu} = P - D + \frac{1}{\sigma} \left[\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} |\nabla \tilde{\nu}|^2 \right] \tag{5}$$

P and D denote the production and destruction term respectively while c_{b2} is a model constant. For more details, see [17]. Using the implicit immersed boundary condition, $\tilde{\nu}$ is set to zero on the geometries.

2.5 Computational geometry and grid

The dimensions of the computational domain were set as $width \times depth \times height = 800m \times 1170m \times 400m$ based on the best practice guidelines (BPGs) provided by COST732 [18] and Architectural Institute of Japan (AIJ) [19]. The domain size can also be seen in Fig.2 where H denotes the distance between the bottom boundary and the highest point of the castle which is approximately 50 meters. Some of the dimensions were intentionally made larger than the prescribed offsets in the BPGs in order to achieve a blockage ratio of less than 3%. The domain consisted solely of hexahedral cells with base grid size of $0.1H(x) \times 0.1H(y) \times 0.1H(z)$. Three refinement levels were applied around the castle for finer discretisation in the area of interest resulting in a minimum cell size of $0.625m$.
A set of 70 points, shown in Fig.3, was selected around the castle and velocity was recorded at each point. The locations of the points were chosen so that different parts of the flow can be captured and compared - upstream of the castle, its wake and around it. A grid convergence study based on the velocities at these points is later presented in Section 3.1.

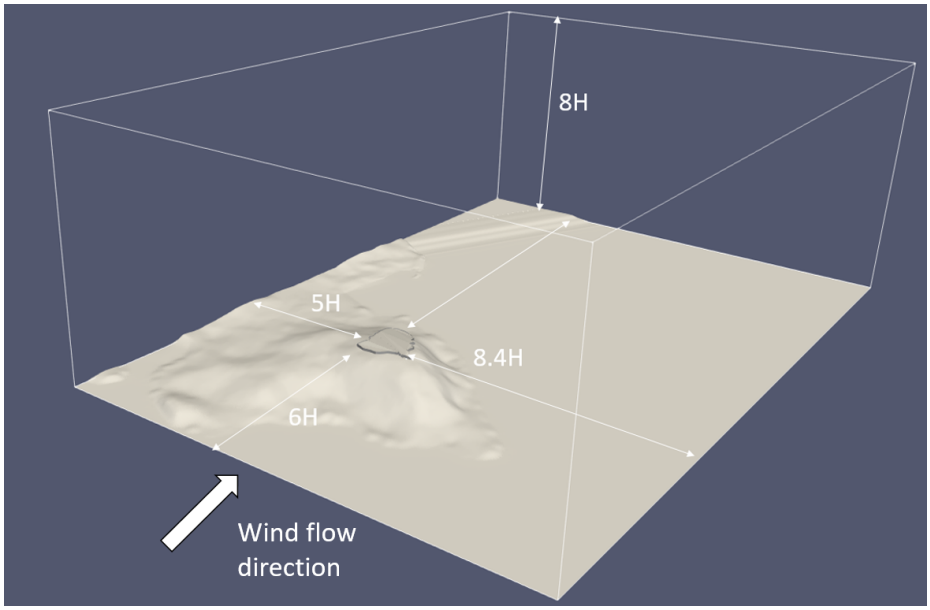

Fig. 2. Computational domain and geometry

Fig. 3. Measurement points

2.6 Boundary conditions

The inlet boundary, as seen in Fig.2, is defined as a velocity inlet with prescribed profiles for velocity and turbulent viscosity. The turbulent viscosity at the inlet is defined as:

$$\nu_t = \frac{C_\mu^{3/4} k^2}{\varepsilon} \tag{6}$$

where C_μ is a constant ($= 0.09$), k is the turbulent kinetic energy and ε is the turbulence dissipation rate. The formulations of the profile for velocity and the aforementioned turbulent quantities are provided by [20]:

$$U(z) = \frac{u^*_{ABL}}{\kappa} \ln\left(\frac{z + z_{0,b}}{z_{0,b}}\right) \tag{7}$$

$$k(z) = \frac{{u^*_{ABL}}^2}{\sqrt{C_\mu}} \tag{8}$$

$$\varepsilon(z) = \frac{{u^*_{ABL}}^3}{\kappa\,(z + z_{0,b})}. \tag{9}$$

In these relationships, u^*_{ABL} denotes the atmospheric boundary layer friction velocity, κ is the von Karman constant ($= 0.42$), z is the height coordinate and $z_{0,b}$ is the roughness length ($= 0.003m$). A reference wind speed of $3.42\ m/s$ was used for calculating the friction velocity, the procedure for which is described in more detail in [21].
Opposite the inlet, an outflow was positioned with a zero pressure Dirichlet boundary condition. Symmetry conditions were applied to the top and lateral boundaries of the domain. The ground and the castle were modelled as no-slip smooth walls.

2.7 Solver settings

The 3D steady RANS equations were solved in combination with the S-A turbulence model using IBOFlow®. The third order QUICK convective scheme [1] was used. All residuals reached a minimum level of 10^{-6}.

3 Results

3.1 Grid independence study

A grid independence study, based on three different grids, was performed in order to reduce the spatial discretisation errors. Coarsening and refining was done with an overall linear factor of 1.5, as advised by [22]. The coarse grid had $727,189$ cells, while the medium and fine grids had $2,679,412$ and $8,296,262$ cells respectively. The normalized velocities (normalization was done with the reference velocity measured at $h = 10m$) at each point for the different grids can be seen in Fig.4. The results indicate a negligible difference between the velocities computed with the medium and fine meshes. Thus, the medium grid was considered in further analysis.

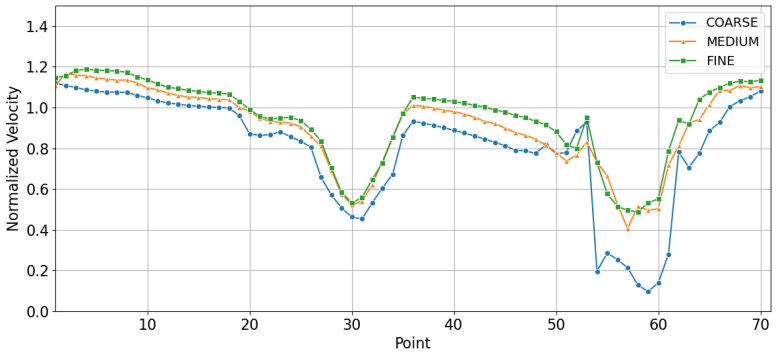

Fig. 4. Grid independence study

3.2 Visualization of the results in Unreal Engine

Unreal Engine 5 (UE5) was used to visualize the results of the simulations. UE5 provides a set of tools which can be leveraged to display scientific results. By combining ready-to-use tools of the engine, such as Blueprints, Materials and the Niagara Particle System, the visualization of the air velocity on the castle's walls became possible. Additionally, a plugin was developed for the visualization of the velocity streamlines. Before using both the built-in UE5 systems and the plugin, the data had to be packed in a form that can be used from the engine. The simulation results follow the Hierarchical Data Format 5 (HDF5), which is a high-performance format used to achieve fast I/O processes and storage for big data [23]. While Unreal Engine doesn't support this format, an intermediate python converter was built, which transformed h5 data into comma separated values (csv). The csv format is particularly useful in this case since it is supported out of the box in UE5. By using this method, the simulation's results were unpacked inside the engine which were later used as input for the developed visualization systems. Figure 5 describes the visualization process inside Unreal.

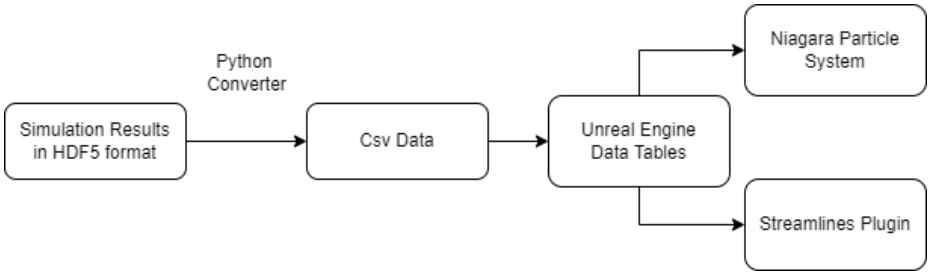

Fig. 5. Data Preparation Flow for Unreal Engine

3.3 Niagara Particle System

Unreal Engine's Niagara Particle System [24] was used in order to visualize the velocity on the castle's walls. Niagara is a robust system when it comes to creating visual effects of any kind. The system is made up of a stack of modules and emitters, executing from top to bottom during different stages of the particle's lifetime. As a first step, the simulation data were packed in two different arrays. Both arrays were a collection of 3D vectors. The first one was used to store the world location of each point while the second array stored the velocity along each axis.

Two different emitters were built inside the particle system. The first emitter accepts the array containing the locations of the points and spawns a spherical mesh on the provided location. In each spawned mesh, a specific material was applied. The material accepts a single 3D vector in which the velocity for each axis is stored. The second emitter passes the velocity values to the material code. By passing the data into the material, it is possible to map values to colors, using a specific colormap. In order for the system to work, minimum and maximum velocity values are determined, followed by the normalization of the results. Finally, the normalized result is mapped to the colormap's texture UV coordinates to get the final color. Figure 6 displays the simulation's results in Unreal Engine.

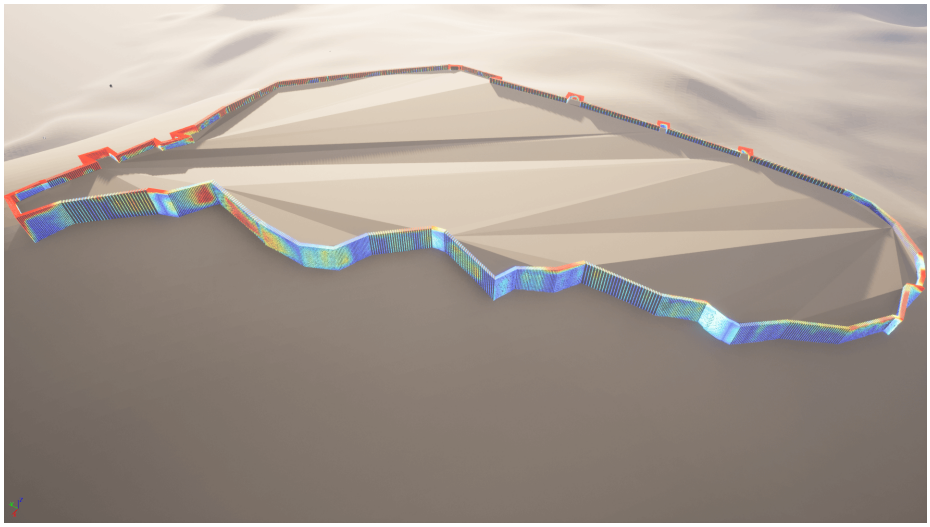

Fig. 6. Velocity Visualization on castle's walls. Low velocities are colored as blue while the red color corresponds to high velocity values.

3.4 Visualizing Velocity Streamlines

To visualize the velocity streamlines an internal plugin was developed. The plugin uses the simulation results to create procedural meshes in Unreal Engine. In this case, the inputs are again a collection of world locations along with their respective velocity values. The plugin accepts the world location of each point as inputs and creates a certain number of vertices around it. Afterwards, it connects vertices of each consecutive point with each other resulting in a cylindrical mesh which visualizes the air's behavior. When all the streamline meshes have been generated, the system iterates over the created vertices and assigns a normalized value as their vertex color. The normalized value is determined from the minimum and maximum velocities and is passed on the material code, similar to the workflow in section 3.3. By assigning a normalized value as a vertex color the velocity value is mapped to the desired colormap. Figure 7 displays the procedural meshes for the streamline data.

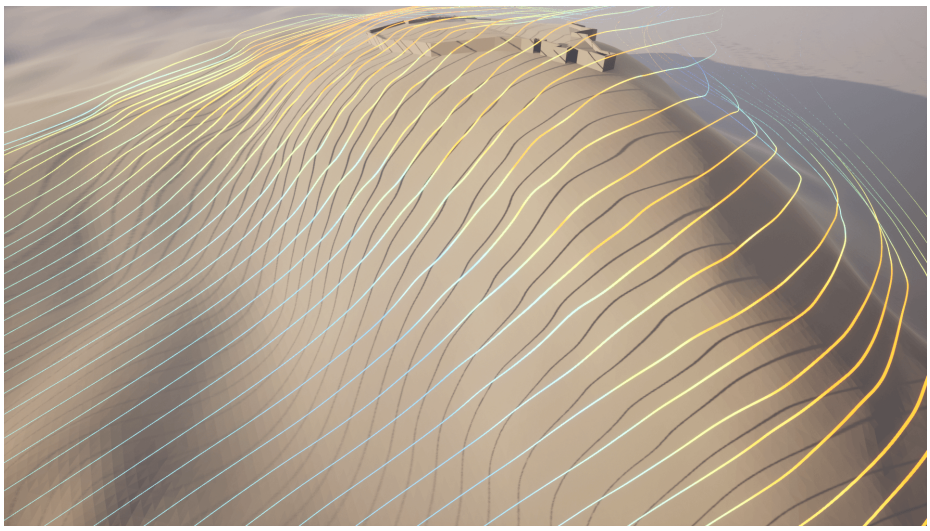

Fig. 7. Streamlines visualization. Low velocities are colored as blue while the red color corresponds to high velocity values.

4 Conclusions

As mentioned in Section 1.1, CFD should not be used as a replacement to experimental studies and decisions should not be made based solely on computational results. Ideally, field measurements or experimental tests should be required, and the retrieved results should be compared with computational data in order to prove the reliability of the CFD method. The pipeline proposed in this paper

can be implemented for an initial assessment of the effect of environmental parameters over any heritage site, as long as solid 3D models of the monuments and their natural environment are provided. A future extension may incorporate other parameters such as temperature and humidity and visualize pressure coefficient and/or shear stress around monuments. To this end it may form a valuable assistive tool for conservators and engineers when developing a holistic conservation or restoration plan for heritage sites, since such kind of simulations may be a mean to generate several hypothesis to be investigated. For instance, the simulation results of wind velocity and temperature in monuments located near the sea, could indicate parts of masonry that suffer from subflorescence (deposition of salts inside the masonry). Finally, more realistic simulation results can be generated, increasing the complexity (Level-of-Detail) of the watertight 3D geometry model of the heritage site.

Acknowledgements The authors would like to thank Vangelis Angelakis [5] for making 3D generation from GIS data look trivial, Mariya Pantusheva [6] at GATE Institute for her feedback on the CFD sections of the paper, Sanjay Somanath[7] at Chalmers University of Technology for his GIS data processing advice, Andreas Mark[8] at the Fraunhofer-Chalmers Centre for his insightful help with IBOFlow and Anders Logg[9] for always managing to find time to cure our computational headaches. This work is part of the Digital Twin Cities Centre supported by Sweden's Innovation Agency Vinnova under Grant No. 2019-421 00041, and GATE Project supported by the Horizon 2020 WIDESPREAD-2018-2020 TEAMING Phase 2 Programme under Grant Agreement No. 857155, the Swedish Research Council for Sustainable Development Formas (grants 2019-01169 and 2019-01885), by Operational Programme Science and Education for Smart Growth under Grant Agreement No. BG05M2OP001-1.003-0002-C01 and by Scientific Fund of Sofia University, agreement No. 80-10-11/10.05.2022.

References

1. H.K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics*. Pearson Education Limited, second edition, 2007.
2. Bert Blocken. Computational Fluid Dynamics for Urban Physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and Environment*, 91:219–245, 2015.
3. J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, first edition, 2001.
4. Jr. John D. Anderson. *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill Education, 1995.

[5] angelakis@hotgates.eu
[6] mariya.pantusheva@gate-ai.eu
[7] sanjay.somanath@chalmers.se
[8] andreas.mark@fcc.chalmers.se
[9] logg@chalmers.se

5. Ashraf S. Hussein and Hisham El-Shishiny. Influences of wind flow over heritage sites: A case study of the wind environment over the giza plateau in egypt. *Environmental Modelling & Software*, 24(3):389–410, 2009.
6. Delia D'Agostino, Paolo Maria Congedo, and Rosella Cataldo. Computational fluid dynamics (CFD) modeling of microclimate for salts crystallization control and artworks conservation. *Journal of Cultural Heritage*, 15(4):448–457, 2014.
7. Xiaoyu Wang, Jinzhu Meng, Tianwei Zhu, and Jingyu Zhang. Prediction of wind erosion over a heritage site: A case study of yongling mausoleum, china. *Built Heritage*, 3(4):41–57, Dec 2019.
8. Paloma Pineda and Alfredo Iranzo. Analysis of sand-loaded air flow erosion in heritage sites by computational fluid dynamics: Method and damage prediction. *Journal of Cultural Heritage*, 25:75–86, 2017.
9. About copernicus, [accessed 20/07/22] https://www.copernicus.eu/en/about-copernicus.
10. Xieyang. Ai super-resolution lossless image enlarging / upscaling tool using deep convolutional neural networks.
11. Open weather map, [accessed 20/07/22], https://openweathermap.org/.
12. A. Mark, R. Rundqvist, and F. Edelvik. Comparison between different immersed boundary conditions for simulation of complex fluid flows. *Fluid Dynamics and Materials Processing*, 7:241–258, 2011.
13. Andreas Mark and Berend G.M. van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227:6660–6680, 2008.
14. R.Mitkov, M. Pantusheva, V. Naserentin, P.O. Hristov, D. Wästberg, F. Hunger, A. Mark, D. Petrova-Antonova, F. Edelvik, and A. Logg. Using the Octree Immersed Boundary Method for urban wind CFD simulations. *IFAC Workshop Control for Smart Cities – CSC 2022*, 2022.
15. G.D. Van Doormaal, J.P.and Raithby. Enhancements of the simple method for predicting incompressible fluid flows, 1984.
16. C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21:1525–1532, 1983.
17. Steven R. Allmaras, Forrester T. Johnson, and Philippe R. Spalart. Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model. *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2015.
18. Jörg Franke, Antti Hellsten, Heinke Schlünzen, and Bertrand Carissimo. *Best Practice Guideline for the CFD Simulation of Flows in the Urban Environment. Cost 732: Quality Assurance and Improvement of Microscale Meteorological Models.* 2007.
19. Yoshihide Tominaga, Akashi Mochida, Ryuichiro Yoshie, Hiroto Kataoka, Tsuyoshi Nozu, Masaru Yoshikawa, and Taichi Shirasawa. AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10-11):1749–1761, 2008.
20. P.J.Richards and R.P.Hoxey. Appropriate boundary conditions for computational wind engineering models using the $k - \epsilon$ turbulence model. *Journal of Wind Engineering and Industrial Aerodynamics*, pages 145–153, 1993.
21. Johannes Brozovsky, Are Simonsen, and Niki Gaitani. Validation of a CFD model for the evaluation of urban microclimate at high latitudes: A case study in Trondheim, Norway. 205, 2021.
22. Joel H. Ferziger and Milovan Peric. *Computational Methods for Fluid Dynamics.* Springer, third edition, 2002.

23. The HDF Group. The hdf5® library & file format, [accessed 20/07/22] https://www.hdfgroup.org/solutions/hdf5/, Jun 2019.
24. Niagara key concepts, unreal engine documentation, [accessed 20/07/22] https://docs.unrealengine.com/5.0/en-us/key-concepts-in-niagara-effects-for-unreal-engine/.