Pontifical Catholic University of Parana

WALTER MAZUROSKI

Intelligent co-simulation: a strategy to solve complex building energy simulation problems

Curitiba
August, 2018

Pontifical Catholic University of Parana

WALTER MAZUROSKI

Intelligent co-simulation: a strategy to solve complex building energy simulation problems

Ph.D. Thesis submitted as a qualifying examination in partial fufillment of the requirements for the degree of Doctor of Mechanical Engineering of the Pontifical Catholic University of Parana.

Area: Engineering and Thermal Sciences

Supervisor: Nathan Mendes
Co-supervisor: Ricardo Coração de Leão Fontoura de Oliveira

Curitiba
August, 2018

Dados da Catalogação na Publicação
Pontifícia Universidade Católica do Paraná
Sistema Integrado de Bibliotecas – SIBI/PUCPR
Biblioteca Central
Edilene de Oliveira dos Santos-CRB9/1636

M476i
2018

Mazuroski, Walter
Intelligent co-simulation: a strategy to solve complex building energy simulation problems / Walter Mazuroski ; supervisor, Nathan Mendes ; co-supervisor, Ricardo Coração de Leão Fontoura de Oliveira. -- 2018
112 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2018.
Bibliografia: f. 105-112

1. Engenharia mecânica. 2. Aprendizado do computador. 3. Dinâmica de fluídos computacional. 4. Inteligência artificial. 5. Calor – Transmissão. I. Mendes, Nathan. II. Oliveira, Ricardo Coração de Leão Fontoura de. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia Mecânica. IV. Título.

CDD 20. ed. – 620.1

Pontifícia Universidade Católica do Paraná
Escola Politécnica
Programa de Pós-Graduação em Engenharia Mecânica

Document of Approval

Walter Mazuroski

Intelligent Co-Simulation: a Strategy to Solve Complex Building Energy Simulation Problems

Doctoral thesis approved as a partial requirement for obtaining the doctoral degree at the Graduate Program in Mechanical Engineering of the Polytechnic School of Pontifícia Universidade Católica do Paraná, by the following committee members:

Mr. Frederic Wurtz
University Grenoble Alpes, CNRS, Grenoble INP*, G2Elab, France

Mr. Etienne Wurtz
Commissariat à L'énergie Atomique et Aux Énergies Alternatives -CEA, France

Mr. Luís Mauro Moura
Pontifical Catholic University of Parana – PUCPR, Brazil

Mr. Adrien François Gros
Pontifical Catholic University of Parana – PUCPR, Brazil

Mr. Ricardo Coração de Leão Fontoura de Oliveira
University of Campinas -UNICAMP, Brazil, Thesis Co-Advisor

Mr. Nathan Mendes
Pontifical Catholic University of Parana – PUCPR, Brazil, Thesis Advisor

Curitiba, August 31, 2018

Abstract

Building Performance Simulation (BPS) tools are under continuous development since the 70's mainly due to energy crises. However, the tools available nowadays still considerably simplify the building physics to perform yearly simulations of whole buildings, especially regarding convective heat transfer, three-dimensional transport and airflow. In addition, Heating, Ventilation and Air Conditioning (HVAC) systems are commonly represented by steady-state, lumped or empirical models. All those simplifications may significantly increase the gap between reality and simulation. To narrow this gap, this thesis proposes a methodology to bring advanced physics into the building simulation field. First, a one-to-one co-simulation approach used by the BPS tool `Domus`, focusing on Computational Fluid Dynamics (CFD) case studies, is presented. The very first results show an excessive computer run time, which motivated the development of a new co-simulation strategy to drastically reduce the high computational cost. This new approach consists of designing a new model for a specific physical phenomenon, called *prediction model*, capable to provide results, as close as possible to the ones provided by the complex model, with a lower computational time. The synthesis of the prediction model is based on artificial intelligence, being the main novelty of this thesis. Basically, the prediction model is constructed by means of a learning procedure, using the input and output data of a co-simulation where the complex model is being used to simulate the physics. Then, the synthesized prediction model replaces the complex model with the purpose of reducing significantly the computational burden with a small impact on the accuracy of the results. Although it is shown that the intelligent co-simulation approach is very promising, the cost associated to the training period is still considerably high. Thus, it is carried out a deep investigation of two techniques - recurrent neural networks (RNN) and proper orthogonal decomposition (POD) to reduce the training time period and improve the accuracy. The results are applied to a two-dimensional diffusive heat transfer problem through a building envelope. To illustrate the use of the new co-simulation approach in a more sophisticated case, the complexity has been raised by including non-uniform radiative boundary conditions and airflow between two zones. A new formulation to represent the incidence and reflections of radiation in building enclosures and on external surfaces is then proposed to obtain the radiative boundary conditions so that the solar radiation is projected precisely on all internal and external surfaces, thanks to the use of a pixel counting technique and a generic view factor algorithm, providing more realistic boundary conditions to the 3-D CFD model that simulates both the solid walls and the air domain of a two-storey residential building. Promising results lead to the conclusion that the proposed strategy enables to bring the accuracy of advanced physics to the building simulation field - using prediction models - with a very reduced computational cost. In addition, re-simulations might be run solely with the already designed prediction model, demanding computer run times even lower than the ones required by lumped models available in BPS tools.

keywords: whole building energy simulation; co-simulation; intelligent co-simulation; machine learning; computational fluid dynamics; radiation distribution.

Acknowledgments

I would like to thank everyone who helped me on this journey. Especially to Prof. Nathan for having been my greatest supporter and having believed in my potential, to Prof. Ricardo for all the advice and valuable shared knowledge. To my colleagues for the support and patience and the members of the examination committee for having made themselves available to evaluate my work.

To Julien, thanks to the French whip I was able to produce my first articles. Thank you Sue for welcoming me during my stay in France and for the valuable tips.

Many thanks to Jane for all the help not only during this PhD, but during all those years I've been at PUC.

Thanks to my family and principally my wife Jucélia for all support and for believing in my choices. Last but not least, Frodo and Arya for always being with me, bringing joy to life even in the most difficult times.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Somewhere, something incredible is waiting to be known.
Carl Sagan

Contents

Contents iii

List of Figures v

List of Tables viii

Introduction 1

1 Co-Simulation to Bring Advanced Physics to Building Thermal Performance Analysis 4
1.1 Introduction 5
1.2 CFD in Buildings 5
1.2.1 Domus - Co-simulation 5
1.3 Simulation 7
1.3.1 Verification 7
1.3.2 Case Study 8
1.3.3 Space discretization 8
1.3.4 CFD Model 10
1.3.5 Co-simulation approach 11
1.4 Discussion 12
1.4.1 Computational Time 15
1.5 Conclusion 15

2 An artificial intelligence based method to efficiently bring CFD to building simulation 18
2.1 Introduction 19
2.2 Literature Review 20
2.2.1 Computational Fluid Dynamics (CFD) 20
2.2.2 Learning techniques 21
2.3 Domus-CFX Co-simulation 21
2.3.1 Definition 21
2.3.2 Domus-CFX co-simulation for a dwelling attic 22
2.3.3 Co-simulation Verification 23
2.4 Intelligent co-simulation strategy 24
2.4.1 General statement 24
2.4.2 Co-simulation training a RNN 25
2.5 Case study 27
2.5.1 CFD Attic model 29
2.5.2 Co-simulation results 29

2.5.3 Applying Intelligent co-simulation methodology . . . 30
2.5.4 Intelligent co-simulation results . . . 32
2.6 Conclusion . . . 33
2.A Simulation tools: Domus and CFX . . . 35
2.A.1 The building energy simulation program `Domus` . . . 35
2.A.2 Computational Fluid Dynamics(CFD) . . . 36

3 Intelligent co-simulation: neural network vs. proper orthogonal decomposition applied to a 2D diffusive problem 38
3.1 Introduction . . . 39
3.2 Intelligent co-simulation approach . . . 40
3.2.1 General statement . . . 40
3.2.2 The Proper Orthogonal Decomposition as a machine learning model . . . 41
3.2.3 The Recurrent Neural Network as a machine learning model . . . 43
3.2.4 Comparing the numerical solutions . . . 44
3.3 Numerical investigation of a simple case study . . . 44
3.3.1 Definition of the problem . . . 44
3.3.2 Numerical application . . . 45
3.4 Intelligent co-simulation for whole-building energy simulation with 2-dimensional diffusive problem . . . 47
3.4.1 The whole building energy program `Domus` . . . 48
3.4.2 The 2D Wall Model . . . 48
3.4.3 The co-simulation between `CFX` and `Domus` . . . 49
3.4.4 Numerical results and discussions . . . 49
3.5 Conclusion . . . 58

4 Intelligent co-simulation: complex 3D analysis considering non-uniform radiative boundary conditions 62
4.1 Introduction . . . 63
4.2 Literature Review . . . 64
4.2.1 Radiation Models . . . 64
4.2.2 Detailed Building Simulation . . . 65
4.2.3 Co-simulation . . . 66
4.2.4 Machine Learning . . . 67
4.3 Methodology . . . 68
4.3.1 Radiation Model Improvements . . . 68
4.3.2 Detailed Building Model . . . 71
4.3.3 Co-simulation `Domus-CFX` . . . 77
4.4 Results . . . 80
4.4.1 Radiation Distribution . . . 80
4.4.2 Detailed Building Simulation . . . 83
4.4.3 Intelligent Co-Simulation . . . 90
4.5 Conclusions . . . 97

Conclusion 99

Bibliography **101**

List of Figures

1.1 Interface of association between Domus and FMU models. . . . 6
1.2 Domus using Pixel Counting technique for complex shading calculation for a non-planar tree. . . . 7
1.3 Comparison of room air temperature between Domus and Domus-CFX. . . . 8
1.4 Temperature and velocity fields in hollowed concrete block. . . . 8
1.5 Case study representation in Domus 9
1.6 GCI value in relation to mesh refinement ratio 10
1.7 Attic mesh generated by ANSYS Mesh. . . . 10
1.8 Outside conditions for temperature. . . . 11
1.9 Outside conditions for relative humidity. . . . 11
1.10 Outside conditions for solar radiation heat flux. . . . 12
1.11 Room air temperature on the winter day 13
1.12 Room air temperature error on a summer day 13
1.13 Room air temperature error on a winter day 13
1.14 Attic air temperature error on summer and winter days 14
1.15 Attic temperature evolution on the winter. . . . 14
1.16 Internal attic temperature and velocity field in a winter day - 17h 15
1.17 Contour plots of the room zone ceiling local range temperature (Winter) 16
1.18 Contour of velocity magnitude, overlaid by the velocity vector field. Calculated at a vertical plane 2-cm away from the attic floor surface (Winter). . . . 17

2.1 Co-simulation scheme between `Domus` and `CFX`. . . . 22
2.2 Diagram indicating Domus and CFX solving parts and the exchange data between the two programs. . . . 23
2.3 Comparison of room air temperature (a) and the error (b) between `Domus` and `Domus-CFX`. 24
2.4 Training (a) and prediction (b) phases. . . . 26
2.5 Case study in the `Domus` graphical interface. . . . 28
2.6 Diagram of the attic model in the case study. . . . 28
2.7 GCI value as a function of the mesh refinement ratio. . . . 29
2.8 Attic mesh generated by ANSYS Mesh. . . . 29
2.9 Outside conditions for temperature, relative humidity and solar radiation flux. . . . 30
2.10 Room air temperature (a) and Room air temperature error (b). . . . 31
2.11 Contour plots of the air local range temperature, overlaid by the velocity vector field. Calculated at a vertical plane 5-cm above the attic floor surface. . . . 31
2.12 Contour plots of the air local range temperature, overlaid by the velocity vector field. Calculated at a horizontal plane in the middle of the attic. . . . 32

2.13 Room air temperature (a) and Room air temperature error (b). 33

3.1 Schematic description of the intelligent co-simulation. 40
3.2 Schematic description of an example of intelligent co-simulation. 41
3.3 Temperature boundary conditions of the training phase (a), of the prediction phase (b) and evolution of the temperature surface for the RNN and POD prediction models (c) with their respective errors (d). 47
3.4 Error ε as a function of the number of the training days. 48
3.5 Schematic description of the case study. 48
3.6 Inside and outside boundary conditions. 50
3.7 Power spectrum of the yearly outside equivalent temperature T_e^{eq} of Curitiba. 51
3.8 Evolution of the outside incident radiation heat flux (a) and temperature (b) of the training climate. 52
3.9 Outside heat flux (a) and temperature (b) evolution during the 8^{th} day with the temperature fields within the wall computed with the co-simulation between `CFX` and `Domus`, during the training phase (c–f). 53
3.10 Evolution of the surface temperature during the whole year (a) and during the last week (b-c). 54
3.11 Time evolution of the error ε with the reference solution. 55
3.12 Variation of the error ε (a) and of the POD order and RNN nodes number (b) as a function of the duration τ of the training phase. 55
3.13 Definition of the improved training climate for the hottest days. 56
3.14 Variation of the outside equivalent temperature T_e^{eq} for $\tau = 2$ days (a) and $\tau = 6$ days (b). . 56
3.15 Time evolution of the error. 57
3.16 Evolution of the inside spatial average surface temperature during the last week of the year. 57
3.17 Variation of the error as a function of the duration τ of the training phase. 57
3.18 Power spectrum of the yearly outside equivalent temperature T_e^{eq} of Miami (a) and Paris (b). 58
3.19 Variation of the utside equivalent temperature T_e^{eq} of Miami (a-b) and Paris (c-d) for $\tau = 2$ days (a-c) and $\tau = 6$ days (b-d). 59
3.20 Time evolution of the error for Miami (a) and Paris (b) cases. 60
3.21 Evolution of the inside spatial average surface temperature for Miami (a) and Paris (b) cases during the last week of the year. 60

4.1 `Domus` using Pixel Counting technique for complex shading calculation for a tree represented by a very complex geometry. 65
4.2 Isometric view of the case study in the SpaceClaim graphical interface. 71
4.3 Details of case study building - west view. 71
4.4 Details of case study building - south view. 71
4.5 Case study with complex shadings in the `Domus` graphical interface. 72
4.6 Outside conditions for temperature, relative humidity and solar radiation flux. 74
4.7 Shadowing pattern example in the `Domus` interface. 75
4.8 `Domus` projected shadowing over the north facade (a) and plot of the matrix with deprojected points (world coordinates) (b). 75
4.9 `SALOME` locating mesh faces based on coordinate points provided by `Domus` (a) and final mesh version with sunlit (orange) group identified over the building geometry in `CFX` (b). 76
4.10 Example of final result of sunlit projection (in orange color) on external surfaces (a) and on internal surfaces (b) of the building. 77

4.11 Sensors position within the building model. 77
4.12 Co-simulation scheme between `Domus` and `CFX`. 78
4.13 Definition of the improved training climate for the hottest days. 80
4.14 Variation of the outside equivalent temperature for a 6-day training period. 80
4.15 Internal solar radiation flux on the floor (a) and on the ceiling (b). 81
4.16 Internal solar radiation flux on different walls. 82
4.17 Total internal solar radiation flux. 83
4.18 Accumulated solar radiation fluxes on all surfaces. 83
4.19 Solar radiation flux on the floor sunlit area. 83
4.20 Solar radiation flux on east wall sunlit area. 84
4.21 Solar radiation flux on west wall sunlit area. 84
4.22 Averaged air temperature with respect to the values obtained for sensors 1, 5 and 9 (a) and sensors 2, 6 and 10 (b), on 21st and 22nd of June. 85
4.23 Averaged air temperature with respect to the values obtained by virtual sensors 1, 5 and 9 (a) and virtual sensors 2, 6 and 10 (b), 21st and 22nd December. 85
4.24 Averaged air temperature with respect to the values for sensors 3, 7 and 11 (a) and sensors 4, 8 and 12 (b), on 21st and 22nd of June. 85
4.25 Averaged air temperature with respect to the values for sensors 3, 7 and 11 (a) and sensors 4, 8 and 12 (b), on 21st and 22nd of December. 86
4.26 Mean sensor temperature and standard deviation values, on 21 and 22 of June(a) and on 21st and 22nd of December(b). 86
4.27 Mean air velocity at the position of sensors 1, 5 and 9 (a) and sensors 2, 6 and 10 (b), on 21st and 22nd of June. 87
4.28 Mean air velocity at the position of sensors 1, 5 and 9 (a) and sensors 2, 6 and 10 (b), on 21st and 22nd of December. 87
4.29 Mean air velocity at the position of sensors 3, 7 and 11 (a) and by virtual sensors 4, 8 and 12 (b), on 21st and 22nd of June. 88
4.30 Mean air velocity at the position of sensors 3, 7 and 11 (a) and sensors 4, 8 and 12 (b), on 21st and 22nd of December. 88
4.31 Mass airflow at the opening, on 21st and 22nd of June(a) and on 21st and 22nd of December(b) 89
4.32 Contour plots of the air local range temperature, overlaid by the velocity vector field, calculated at a vertical plane 1 m from the west wall, on December 21st. 89
4.33 Contour plots of the east wall local range temperature, at 14h30 of the day 21 of June, overlaid by the velocity vector field calculated at a vertical plane 1 cm from the east wall (a), zoom in the left corner showing the sunlit mesh (b). 90
4.34 Contour plots of the surface local range temperature, 13:30h to 15h on June 21st. 91
4.35 Contour plots of the surface local range temperature, 15:30h to 17h on June 21st. 92
4.36 Air temperature and RNN prediction for sensor 1 (a) and temperature prediction RMS error (b), in the winter period. 93
4.37 Air temperature and RNN prediction for sensor 1 (a) and temperature prediction RMS error (b), in the summer period. 94
4.38 Air temperature and RNN prediction for sensor 7 (a) and temperature prediction RMS error (b), in the winter period. 94
4.39 Air temperature and RNN prediction for sensor 7 (a) and temperature prediction RMS error (b), in the summer period. 95

4.40 Velocity and RNN prediction for sensor 1 (a) and prediction RMS error (b), in the winter period. . . . 95
4.41 Velocity and RNN prediction for sensor 1 (a) and prediction RMS error (b), in the summer period. . . . 95
4.42 Velocity and RNN prediction for sensor 7 (a) and prediction RMS error (b), in the winter period. . . . 96
4.43 Velocity and RNN prediction for sensor 7 (a) and prediction RMS error (b), in the summer period. . . . 96
4.44 Mass airflow at the opening between the two zones (a) and prediction RMS error (b), in the winter period. . . . 96
4.45 Mass airflow at the opening between the two zones (a) and prediction RMS error (b), in the summer period. . . . 97

List of Tables

1.1 Range of parameters used in simulation 9
1.2 National standard equivalent layer properties 12
1.3 Comparison of Computing Time for performed simulations 15

2.1 Properties of the material used in the verification. 24
2.2 Link between tools and model in each phase. 27
2.3 Range of parameters used in simulation. 28
2.4 Computational times for the performed simulations. 32
2.5 Computer run time for a 10 days period. 34
2.6 Computer run time for a one-year period. 34

3.1 Computational time of the co-simulation. 54

4.1 Range of parameters used in simulation. 72
4.2 Glass properties. 73
4.3 Positions of the sensors with respect to the coordinate system of the building. 78
4.4 Table of weights calculated for the RNN-FD model to predict temperature in sensor-1 . . . 97
4.5 Table of weights calculated for the RNN-RC model to predict temperature in sensor-1 . . . 97
4.6 Table of weights calculated for the RNN-RC model to predict temperature in sensor-7 . . . 98

Introduction

Introduction

A great concern about buildings energy consumption emerged during the energy world crisis in the seventies. According to Brazilian Energy Balance [1], 51% of Brazil's electricity consumption in the year 2016 was attributed to residential, commercial and public buildings. In commercial buildings, 30% of energy consumed is mainly due to air conditioning systems [2], having a directly impact on the cost of operating a building and an indirect impact on the environment, thus revealing the urgent need for rapid improvement of energy-efficient products and the importance of energy-based computational tools for the construction sector. The precise calculation of heat transfer phenomena in buildings is a topic where much attention has been given since the appearance of the first building simulation programs in the seventies such as DOE-2 [3], BLAST [4] and ESP-R [5] and more recent tools such as EnergyPlus [6] and Domus [7].

These programs, also know as Building Performance Simulation (BPS) tools, are the only ones capable to perform whole-building energy simulation, where different physical phenomena need to be simultaneously simulated. However, due mainly to high computational time demanded to simulate the complex mathematical models associated to heat transfer phenomena, a general characteristic of BPS tools is the adoption of simplified mathematical models, where advanced physics are neglected.

A computational procedure known as *co-simulation* [8, 9] has been proposed in the literature as a possibility to extend the capabilities and improve the accuracy of building performance simulation tools. Basically, the strategy relies on a real-time data exchanging between a BPS tool and a specialized software, where specific physical phenomena are simulated more accurately thanks to a more complex model, enabling to take advanced physics into account rather than considering, for instance, 1-D models or lumped models based on integral formulation for energy and mass balances. Among many possibilities where this technique can be employed, one could mention airflow, three-dimensional heat transfer or detailed HVAC systems simulation, which are commonly simplified in BPS tools.

In this context, this thesis first presents a coupling technique used by the building simulation tool `Domus`, focusing in CFD case studies applying the one-to-one *co-simulation* approach to show, for instance, the effects of considering the full Navier-Stokes formulation and turbulence modelling on the thermal performance transient simulation of a dwelling. It has been shown with the case study that transient CFD coupling is inviable in the building simulation field due to an excessive computer run time. Therefore, the main purpose of this PhD thesis is to develop a new co-simulation strategy to drastically reduce the high computational cost observed for a simple CFD application.

The new approach consists of designing a new model for a specific physical phenomenon, called prediction model, capable to provide results, as close as possible to the ones provided by the complex model, with a lower computational run time. The synthesis of the prediction model is based on artificial intelligence, being the main novelty of this thesis. Basically, the prediction model is constructed by means of a learning procedure, using the input and output data of a co-simulation where the complex model is being used to simulate the physics. Then, the synthesized prediction model replaces the complex model with the purpose of reducing significantly the computational burden with a small impact on the accuracy of the results. Technically speaking, the learning phase is performed using a machine learning technique, and the model investigated here is based on a recurrent neural network (RNN) model and its features and performance are investigated on a case study, where a single-zone house with a triangular-prism-shaped attic model is co-simulated with both `CFX` (CFD tool) and `Domus` (BPS tool) programs. Promising results lead to the conclusion that the proposed strategy enables to bring the accuracy of advanced physics to the building simulation field - using prediction models - with a very reduced computational cost. In addition, re-simulations might be run solely with the already designed prediction model, demanding computer run times even lower than the ones required by the lumped models available in the BPS tool.

Although it is shown that the intelligent co-simulation approach is very promising, the cost associated to the training period is still too high. Thus, it is carried out a deep investigation of two techniques - recurrent neural networks (RNN) and proper orthogonal decomposition (POD) reduction method -, whose main goal is to reduce the training time period and to improve the accuracy. The case study focuses on a co-simulation between `Domus` and `CFX` , performing a two-dimensional diffusive heat transfer problem through a building envelope. An important contribution is to show the POD as a promising alternative to reduce the training time period of intelligent co-simulations for purely diffusive problems in more than one dimension.

The last part of this manuscript proposes to raise to a new level the complexity of the case study, focusing mainly on the boundary conditions. A new formulation to represent the incidence and reflections of radiation in building enclosures and on external surfaces is proposed to obtain non-homogeneous radiative boundary conditions so that the solar radiation is projected precisely on all internal and external building surfaces. This task can be accomplished thanks to the use of a pixel counting technique and a generic view factor algorithm, that enable to provide more realistic boundary conditions to the 3-D CFD model that simulates both the solid walls and the air domain of a two-storey residential building.

Thesis outline

Two scientific papers, published in the Journal of Building Performance Simulation, and a conference paper, published in the proceedings of the Building Simulation 2017 conference, form the three first chapters of this document. The conference paper, in **Chapter 1**, emphasizes the importance of the co-simulation techniques to bring advanced physics to building performance simulation. Meanwhile, some results from a case study are provided showing discrepancies when advanced physics is brought for an attic simulation. The paper in **Chapter 2** provides a deeper analysis of the intelligent co-simulation approach, as well devising a solution to overcome the main limitation of co-simulation with very time consuming tools by using a machine learning technique. A step forward is taken in the paper presented in **Chapter 3** that explores alternatives to reduce the training time, with different approaches for the training period and a comparison with an other machine learning technique. The last chapter, **Chapter 4**, presents a new level of complexity in building 3-D simulation and also applies the *intelligent co-simulation* approach to validate the technique in a more complex situation.

Chapter 1

Co-Simulation to Bring Advanced Physics to Building Thermal Performance Analysis

Paper presented in the IBPSA Building Simulation 2017 - San Francisco, California, August 7 – 9, 2017. Id: 2709.

1.1 Introduction

A more accurate assessment of building energy efficiency through computer simulation in general requires the use of highly efficient and specialized software. Presently, there is a wide variety of models and software to simulate an entire building or a specific component. However, the most powerful tools available are rarely combined for highly detailed simulation of buildings and their components. To go deeper into the subject, highly relevant issues for building simulation are addressed in [10], such as the significant reduction of emission of greenhouse gases and a substantial improvements in health, comfort and productivity that could be achieved with the help of a BPS tool when buildings and their systems are treated as a complete optimized entity. According to [11], the state-of-the-art BPS tools cannot always perfectly represent the physical phenomena, because of fragmented development and rapid innovations in building and system technologies, which makes difficult the software maintenance and its evolution. In addition, in line with [11] the tool development team needs to have an in-depth knowledge of the software architecture, programming language and modeling approaches and strategies. One proposition adopted to solve this problem of evolution of BPS tools is the co-simulation procedure ([9]), also known as external coupling ([12]). To bring advanced physics, such as CFD, to building thermal performance analysis, co-simulation techniques can be applied to combine several tools from different domains and development teams. In the present work, co-simulation techniques enables the use of Computational Fluid Dynamics (CFD) tools, Modelica Models and coupling with other simulation tools compatible with standard communication interface FMI for co-simulation.

1.2 CFD in Buildings

CFD simulation research in buildings has been reported since the nineties and the reader may refer to [13], [14], [15], [16], [17], [18], [19] among others. For instance, [18] presented a BPS tool integrated to a CFD code for the definition of convective heat transfer coefficients. [20] used a BPS to determine the boundary conditions for a CFD coupling simulation and used a CFD simulation to provide inputs about atrium geometry, such as airflow and heat transfer coefficients, to the BPS tool. [21] described coupling methods between CFD and BPS tool and when they must be used. [22] also showed a guideline for selection of simulation tools for airflow prediction, addressing approaches representing different resolution such as Zonal Airflow Network and CFD. Co-simulation with CFD tools enables to bring information from physical phenomena normally disregarded in building simulation due to the complexity of solving Navier-Stokes formulation and turbulence modeling and due to a high computer run time. Although the dramatic evolution of computer hardware and CFD tools since the seventies - when the first building energy simulation tools appeared -, the use of CFD is still very restricted to a few researchers and a few consultants in the building sector. The difficulties are due, on one hand, to the lack of numerically robust and user-friendly tools and, on the other hand, to the high computer run time. In this work, we try to give one step ahead to bring the fluid mechanics to building simulation by coupling two user-friendly programs. However, the computer cost remains a very important issue as it is shown in the paper.

1.2.1 Domus - Co-simulation

Co-simulation can be defined as a type of simulation where at least two simulation tools jointly solve systems of differential-algebraic equations and exchange data during the time the coupling is performed. Each tool may offer different numerical solutions to a physical or mathematical problem.

There are several co-simulation techniques such as the one-to-one approach and the FMI. Both were implemented in the Domus simulation program to allow coupling of this one with other models and tools. For instance, an one-to-one coupling was adopted to combine Domus to the commercial CFD tool ANSYS-CFX. The type of coupling used was the Ping-Pong method ([8]), creating a weak coupling between the tools. The use of the FMI standard provides greater scope for co-simulation between tools that adopt the same standards such as EnergyPlus, Modelica Models and many others available at https://www.fmi-standard.org.

Domus-FMU

Software support for the Functional Mock-up Interface certainly opens a new range of opportunities in modeling and functionalities expansion, thanks to a large number of tools adhering to this standard and the possibility of using complex models developed in multi-domain languages such as Modelica ([23]). Initiatives such as the creation of the Buildings Library ([24]), a free, open-source library for building and community energy systems, developed within the Annex 60 project, conducted under the umbrella of the International Energy Agency's Energy in Buildings and Communities Programme (IEA EBC)([25]), provides validated model library that can be used with existing building simulation programs. For enabling the use of model libraries, such as the Buildings Library, the Domus program has been modified to support the FMI standard. Currently, for FMI v. 1.0, a communication interface between the user and the chosen FMU model has been developed, as it can be seen in Figure 1.1. The FMU can be define as a simulation model which implements the FMI standard and is distributed in a zip file with the extension ".fmu".

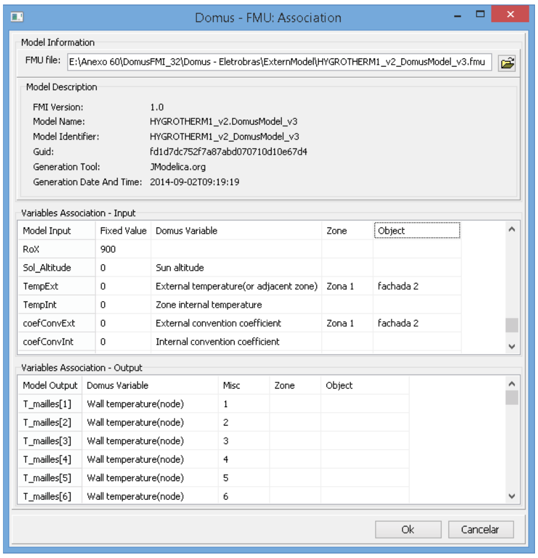


Figure 1.1: Interface of association between Domus and FMU models.

In the first stage of the development, only envelope models are possible to be coupled and Domus'user, after loading the FMU file, will have access to all the input and output variables configured for the external model, linking the intuitively FMU variables with Domus variables.

It is also possible to use a FMU for tool coupling where, for example, the entire envelope simulation can be done through a program such as EnergyPlus, while complex boundary conditions are provided by Domus, such as solar direct radiation on surfaces (Figure 1.2), obtained by means of the pixel counting technique ([26]) implemented in Domus.

Although the attic in Domus can be treated as a building envelope element, there is no FMI interface for complex 3-D CFD simulation available so far, which made impossible to consider the FMI method in the case study presented in this work.

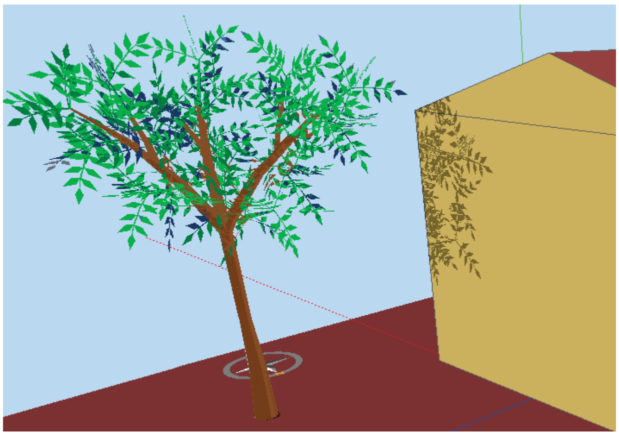


Figure 1.2: Domus using Pixel Counting technique for complex shading calculation for a non-planar tree.

Domus-CFD

The coupling between Domus and ANSYS-CFX is done externally, also called external coupling or even co-simulation. One of the first accomplished couplings was the use of CFD to calculate specific walls, cases that require greater complexity of computational modeling, such as hollowed block wall that receive non-uniform boundary conditions during the run-time period of simulation. This coupling requires the pre-definition of a template of the target geometry, as well as the configuration of the mesh by the user. The information transmitted to the CFD tool, such as air temperature, incident radiation flux and convective heat transfer coefficients, are from the previous time-step simulation. After converging the CFD model, the results are transmitted back to Domus that uses the data to complete the time step as well as the energy and mass zone balances.

1.3 Simulation

1.3.1 Verification

The Co-Simulation between Domus and CFX using the one-to-one approach has been verified by the analysis of a simple monolithic homogeneous wall. First, the wall heat transfer Domus model was used in the simulation of a simple zone with all adiabatic walls except the co-simulated wall. Then, the results were compared with the co-simulation between Domus and CFX, showing a good agreement as noticed in Figure 1.3.

Similar verification (not shown) was done in the co-simulation using the FMU (Functional Mockup Unit) by a pure heat transfer model developed in Modelica and exported to FMU by JModelica tool. The results are promising as they allow simulation of highly complex elements such as hollow blocks, ventilated walls and attics, internal and external airflow. Figure 1.4 illustrates a case study of a hollowed block wall co-simulated by CFX and Domus ([27]).

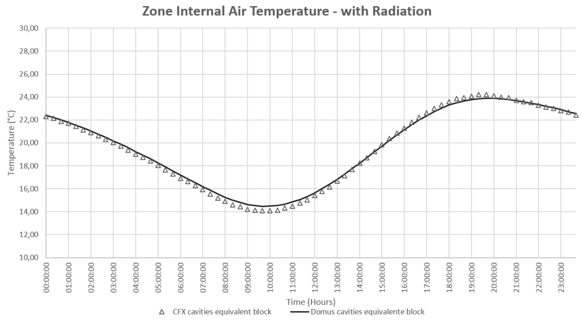


Figure 1.3: Comparison of room air temperature between Domus and Domus-CFX.

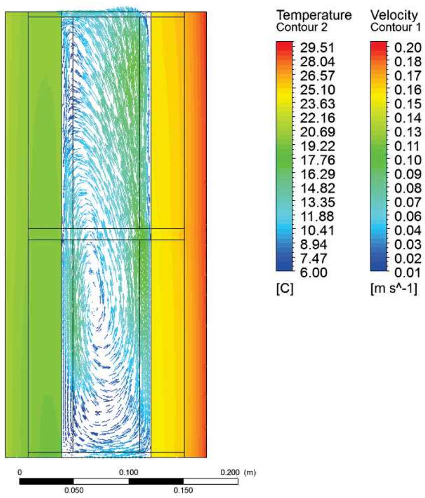


Figure 1.4: Temperature and velocity fields in hollowed concrete block.

1.3.2 Case Study

After the co-simulation of a hollow concrete block wall, a case study of a building with two thermal zones, a conventional 96-m^3 room and a 13.44-m^3 attic, is considered and illustrated in Figure 1.5, that aims at evaluating the effect of natural convection and non-uniform solar distribution in the attic on the temperature of the room zone using different approaches. In order to evaluate the isolated effect of the attic on the temperature of the room, an adiabatic condition was assumed for all the other walls and floor of room the zone. The ceiling and the attic temperatures were evaluated by sensible heat transfer models using Domus and EnergyPlus and a co-simulation between Domus and ANSYS-CFX. In the co-simulation case, the CFD model was responsible for calculating the entire attic and the transient heat flux to the room zone. It was also compared the effect on the temperature of the zone when using the ceiling element composed of layers equivalent to the composition of the attic, according to a national standard. The thermal parameters of different materials used in the simulation are listed in Table 1.1. Simulations were performed using a computer with an Intel Core i7 4790 Processor with a four-core CPU at 3.6 GHz. The BPS tools used only a single core for the simulation, while CFX used two available cores.

1.3.3 Space discretization

A CFD adaptive mesh was generated by ANSYS Mesh generator for the attic geometry and to assure better independence of the CFD simulation with respect to computational grid, a grid-convergence

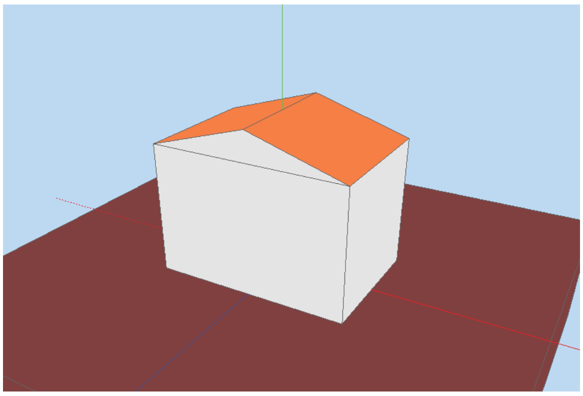


Figure 1.5: Case study representation in Domus

Table 1.1: Range of parameters used in simulation

Parameter	Material		
	Air	Roof Tile	Concrete Slab
Density (kg m^{-3})	1.200	2000	2200
Specific heat ($Jkg^{-1}K^{-1}$)	1007	920	1000
Thermal conductivity ($Wm^{-1}K^{-1}$)	0.024	1.05	1.75

study was performed adopting the Grid Convergence Index (GCI) method, proposed by [28]. This method provides an approach for uncertainty assessment of grid convergence based on a grid refinement error estimator. As GCI was originally proposed to be applied in uniform grids, an alternative option for adaptive grids was suggested in [29] where a number of mesh nodes could be used to calculate refinement ratio. Based in [29], the equation to calculate GCI is expressed as:

$$GCI = 3|\varepsilon|/(r^p - 1), \tag{1.1}$$

$$\varepsilon = (f_2 - f_1)/f_1, \tag{1.2}$$

$$r = (N_{fine}/N_{coarse})^{1/3}, \tag{1.3}$$

where f_1 is the variable value at a point with a fine grid; f_2 is the variable value at the same point with a coarse grid; N_{fine} and N_{coarse} are the grid numbers of fine and coarse mesh, respectively, and p denotes the numerical scheme order of accuracy. The GCI calculation was performed within the limitations imposed by the computational capacity of the work environment. With a refinement ratio of 1.5, four mesh models with 10, 30, 100 and 300 thousand nodes were generated and evaluated. The GCI results (Figure 1.6), enables to notice that a mesh of 100 thousand nodes presents good mesh independent results if it is considered that greater refinements do not make it any better. Figure 1.7 shows the 100 thousand node mesh generated by ANSYS Mesh.

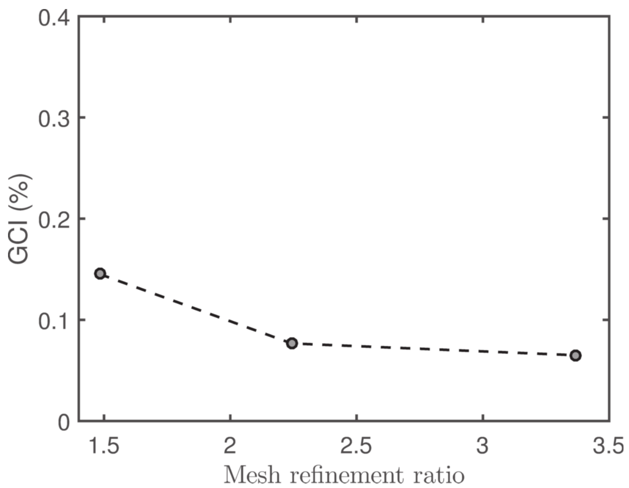


Figure 1.6: GCI value in relation to mesh refinement ratio

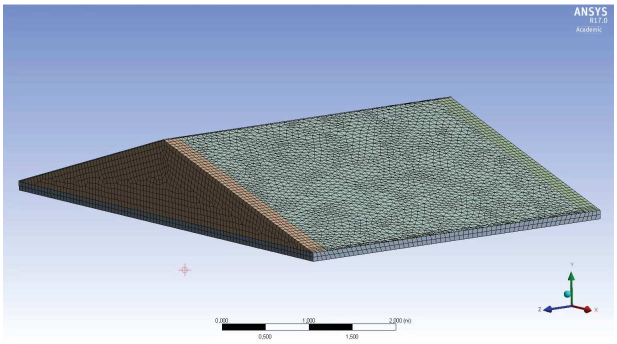


Figure 1.7: Attic mesh generated by ANSYS Mesh.

1.3.4 CFD Model

The CFD model was configured using a high resolution transient scheme that alternates between the use of second-order backward Euler scheme and first order backward Euler scheme. The convergence criterion was defined as a root-mean-square (RMS) residual level of 10^{-3}, which residual could be considered a loose convergence criterion but it was sufficient to fulfill the objective of this case study. As the condition presented in the attic model represents mainly the natural convection phenomenon, the model adopted for turbulence was the k-Omega. The effect of gravity was taken into account in the buoyancy term. There is no air exchange or infiltration into the attic. For the boundary conditions, a non-slip condition was imposed at solid surfaces with a third-type boundary condition. The following radiative heat fluxes calculated by Domus were used by ANSYS-CFX to be added to the convective heat flux boundary condition:

$$q_{lw} = -\varepsilon\sigma F_f (T_{Sup}^4 - T_{Ref}^4), \tag{1.4}$$

$$q_{sw} = q_{dir} + q_{diff} + q_{refl}, \tag{1.5}$$

$$q = q_{lw} + q_{sw}, \tag{1.6}$$

$$T_{Ref} = (T_{sky} + T_{ground})/2, \tag{1.7}$$

where q_{lw} is the long-wave radiation flux (W/m^2), ε is the emittance of surface, F_f is the view factor from surface to the sky or horizon (in case of horizontal surfaces), σ is the Stefan-Boltzmann constant (5.67 x $10^{-8} W m^{-2} K^{-4}$), T_{Sup} is the surface temperature in Kelvin and T_{Ref} is the sky temperature or an averaged temperature (Eq. 2.4) if the surface is vertical. The q_{sw} is the short-wave radiation heat flux (W/m^2) as well as q_{dir}, q_{diff} and q_{refl} are the direct, diffuse and ground reflected radiation fluxes (W/m^2) .

The outside conditions represent a temperate climate of Curitiba, South of Brazil, shown in Figures 1.8 - 1.10 and taken from Domus database. The inside temperature and relative humidity varies freely. With the intention of presenting results for more diverse situations of weather climate, two days were chosen to be presented in this work, one day on the winter solstice in the southern hemisphere, June 21, and the other one on the summer solstice, December 21.

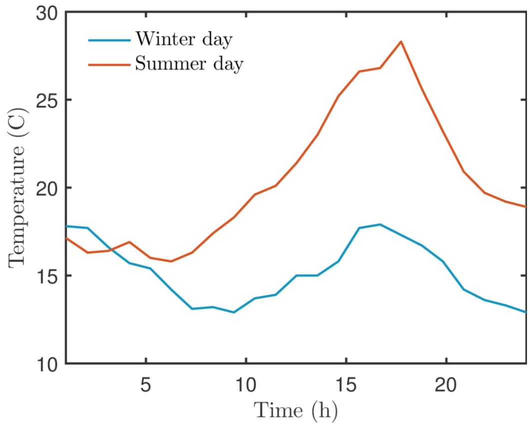


Figure 1.8: Outside conditions for temperature.

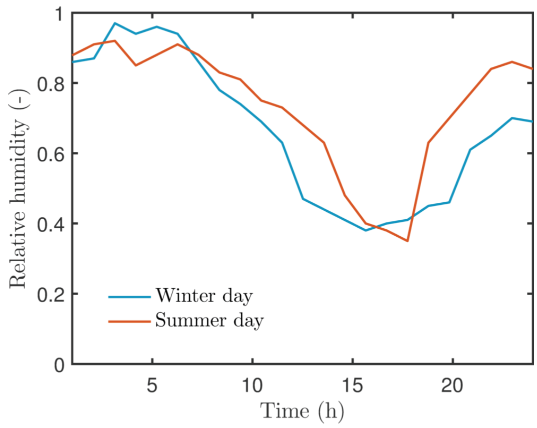


Figure 1.9: Outside conditions for relative humidity.

1.3.5 Co-simulation approach

The BPS tool (Domus) and the CFD software calculate the steps alternately by means of a ping-pong method, using a time step of ten minutes for Domus and 1 minute for CFX.

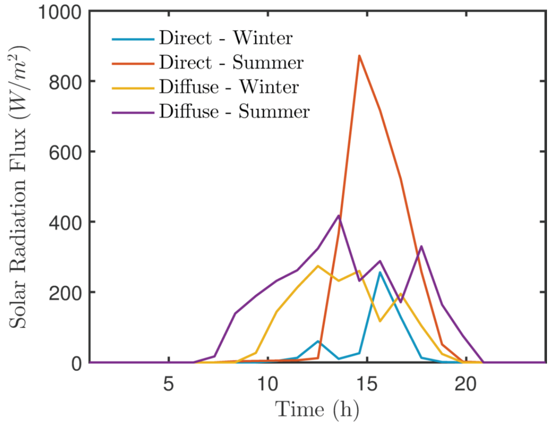


Figure 1.10: Outside conditions for solar radiation heat flux.

1.4 Discussion

The room air temperature evolutions are shown in Figure 1.11 for the winter day, using three different approaches, while Figures 1.12 and 1.13 present their errors, using the advanced approach (Domus-CFX) as the reference value, according to Eq. (1.8).

$$RMSE = \sqrt{\frac{(T_{Domus} - T_{CFX})^2}{T_{CFX}^2}} \tag{1.8}$$

The simplest approach (Domus - national standard) applies one layer of the equivalent properties (Table 1.2) from the national regulation code (RTQ-C)([30]), used when the attic is not modeled but may provide no accurate results as shown in Figures 1.11 - 1.13. Standalone solutions presented a maximum absolute difference o 2.4^oC, remembering that only the attic was co-simulated with CFX due to the high computer run time.

Table 1.2: National standard equivalent layer properties

Properties	**Value**
Thermal transmittance ($W/(m^2K)$)	2.05
Thermal capacity ($kJ/(m^2K)$)	238
Solar absortance (-)	0.8
Solar factor (-)	6.6

Figure 1.14 shows the error for the prediction of the attic temperature for the summer and winter case. The error magnitude is higher and dependent on the external boundary condition, i.e., as the corner and the attic internal airflow effect become more relevant. Figure 1.15 shows the attic temperature evolution during the winter day by means of simulations carried out with EnergyPlus, Domus and Domus-CFX. In general, the CFD results may present a spatial temperature difference heterogeneity (Figure 1.16), while both Domus and EnergyPlus, as standalone tools provide only a time-dependent temperature. ANSYS-CFX solves three-dimensionally and with high accuracy the energy, momentum and mass conservation differential equations, while Domus and EnergyPlus use a lumped approach for solving an integral formulation - based only on energy and mass balance equations - and disregard the spatial variation of the temperature and air velocity fields that promote higher temperature predictions due mainly to an increase on the convective and corner effects.

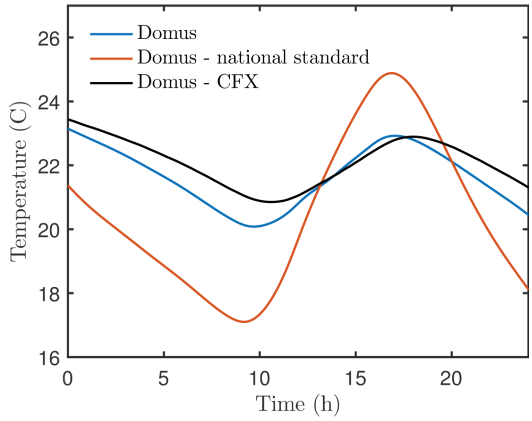


Figure 1.11: Room air temperature on the winter day

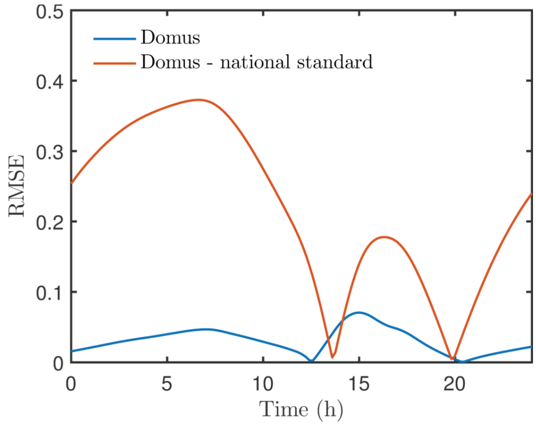


Figure 1.12: Room air temperature error on a summer day

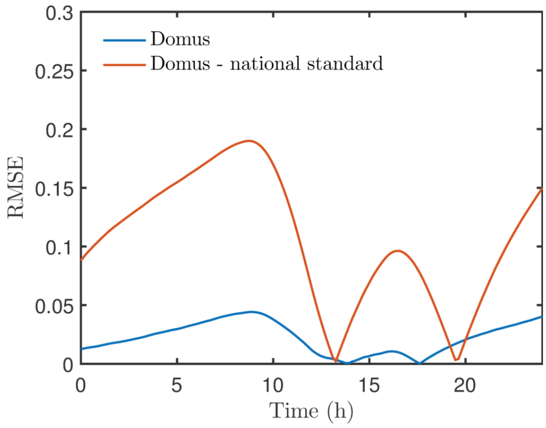


Figure 1.13: Room air temperature error on a winter day

Although the averaged attic air temperature values obtained by the software coupling (shown in Figure 1.15) is similar to the ones obtained by the standalone tool when there is no solar radiation, a difference of 2.2^oC can be noticed at 11:00 am. As the averaged values hide the heterogeneity of the air temperature profile within the attic space, the maximum air temperature was also plotted in Figure 1.15, showing a difference as high as 5.4^oC at 11:00 am.

Regarding the differences between the two BPS programs (Domus and EnergyPlus), we believe they can be attributed mainly to the methodology used for the calculation of external and internal

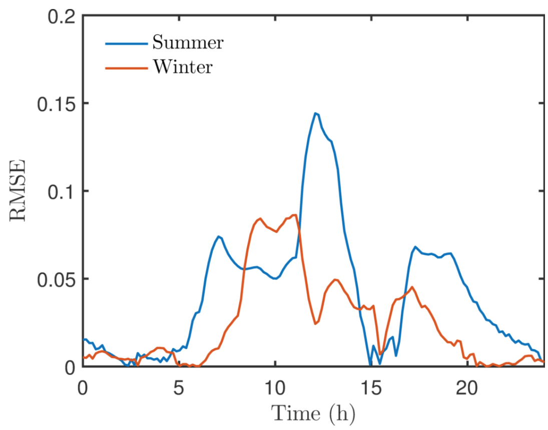


Figure 1.14: Attic air temperature error on summer and winter days

long-wave radiation heat transfer. No difference has been observed between them when all radiation mechanisms were ignored and very small differences have been noticed (not shown) associated to the boundary conditions due to short-wave radiation values on all surfaces. Therefore, as Domus provides the boundary conditions to CFX, it is expected that Domus results will be closer to Domus-CFX results shown in Figure 1.14. The opposite could be also expected if the coupling had been done between EnergyPlus and CFX.

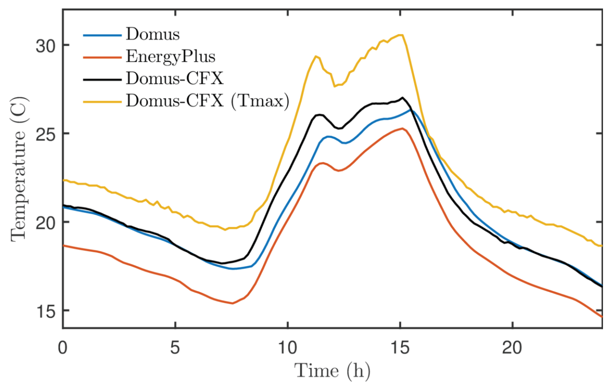


Figure 1.15: Attic temperature evolution on the winter.

Figure 1.16 shows both the attic temperature and air velocity profiles. In this result it is possible to observe the convective effect on the heat distribution inside the attic and the effect on the ceiling temperature, consequently modifying the temperature distribution within the room zone. This natural convection effect also seems to be the main cause of the higher temperatures of the attic and zone on the results of the coupled model.

In Figure 1.17, it is possible to observe, during the winter day, regions of the ceiling which are predominantly colder during the whole period of the day. The effect of the convective movement of the air inside the attic has a clear influence on the temperature variation in the ceiling of the zone in a multidimensional form, observed in Figure 1.18, which illustrates the velocity field near the attic floor during the period. An important point to be considered is the asymmetry of the radiative heat flux as the outdoor boundary condition, which may impose a strong multidimensional heat flux and magnify the natural convection within the attic.

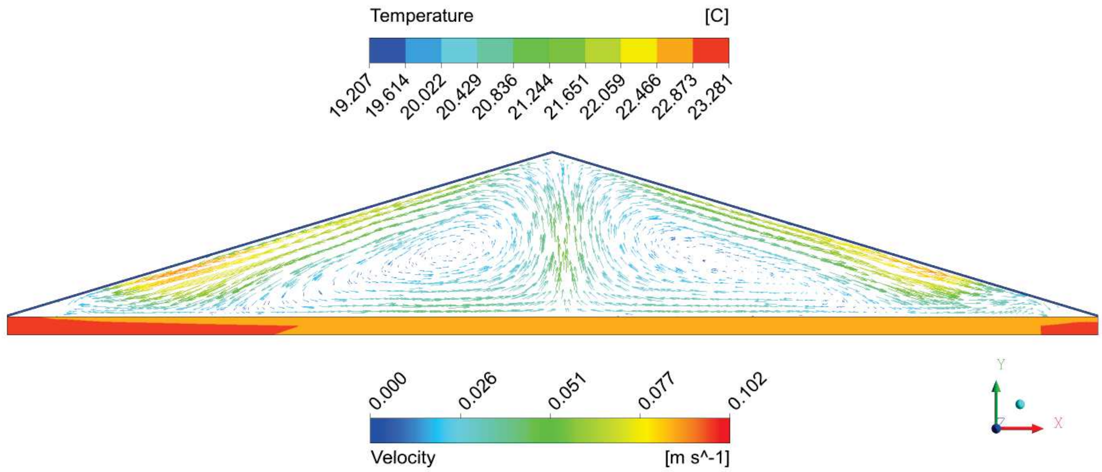


Figure 1.16: Internal attic temperature and velocity field in a winter day - 17h

1.4.1 Computational Time

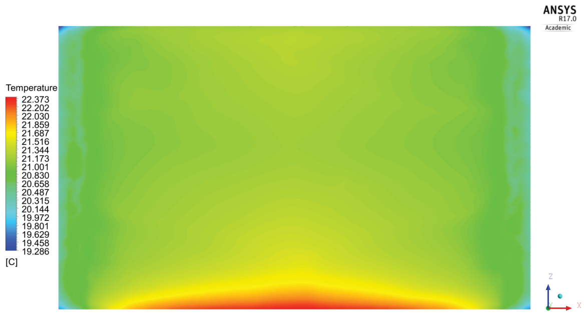
Table 2.4 shows the computing time for all performed simulations. Due to the higher computational cost of the simulation using the CFX coupling, only 8 days of simulation were performed, for each result day. The first 7 days were used as a warm-up period. In the co-simulation with CFX, the measured time was over the period of 8 days of the summer day simulation.

Table 1.3: Comparison of Computing Time for performed simulations

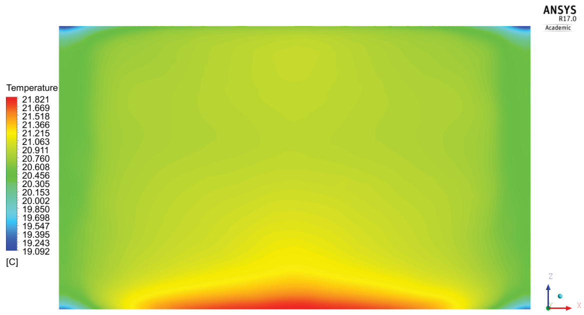
Approch	**Time (hr:min:sec)**
Domus	00:00:02
Domus - national standard	00:00:02
EnergyPlus	00:00:03
Domus-CFX	249:33:05

1.5 Conclusion

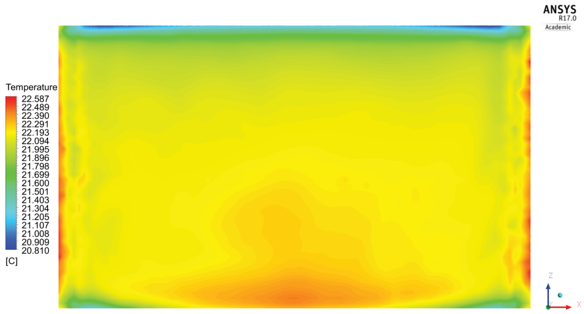
The multidimensional nature of heat transfer and the complex airflow in buildings may play an important role as shown in this paper. In addition, the lumped model might be limiting, resulting in an inaccurate representation of the physics occurring in the attic. Therefore, despite the high computational cost, it is shown the importance to bring, to building simulation tools, an advanced modeling of physical phenomena for a more precise evaluation of thermal and energy performance, which can be reasonably accomplished via co-simulation techniques. The differences shown in this paper could have been magnified under the presence of ventilation/infiltration loads or under heterogeneous boundary conditions due to complex shading on the roof. On other hand, as observed in Table 2.4, research efforts are needed to reduce the computer run time to bring advanced physics to building simulation so that it becomes a practice conducted not only by researchers but also by consultants and building designers. This might be one of the most important challenges for the development of the building simulation area in the forthcoming years, including urban physics.



(a) $t = 6$ h

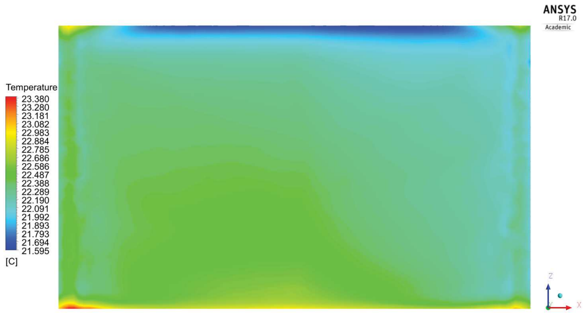


(b) $t = 9$ h



(c) $t = 12$ h

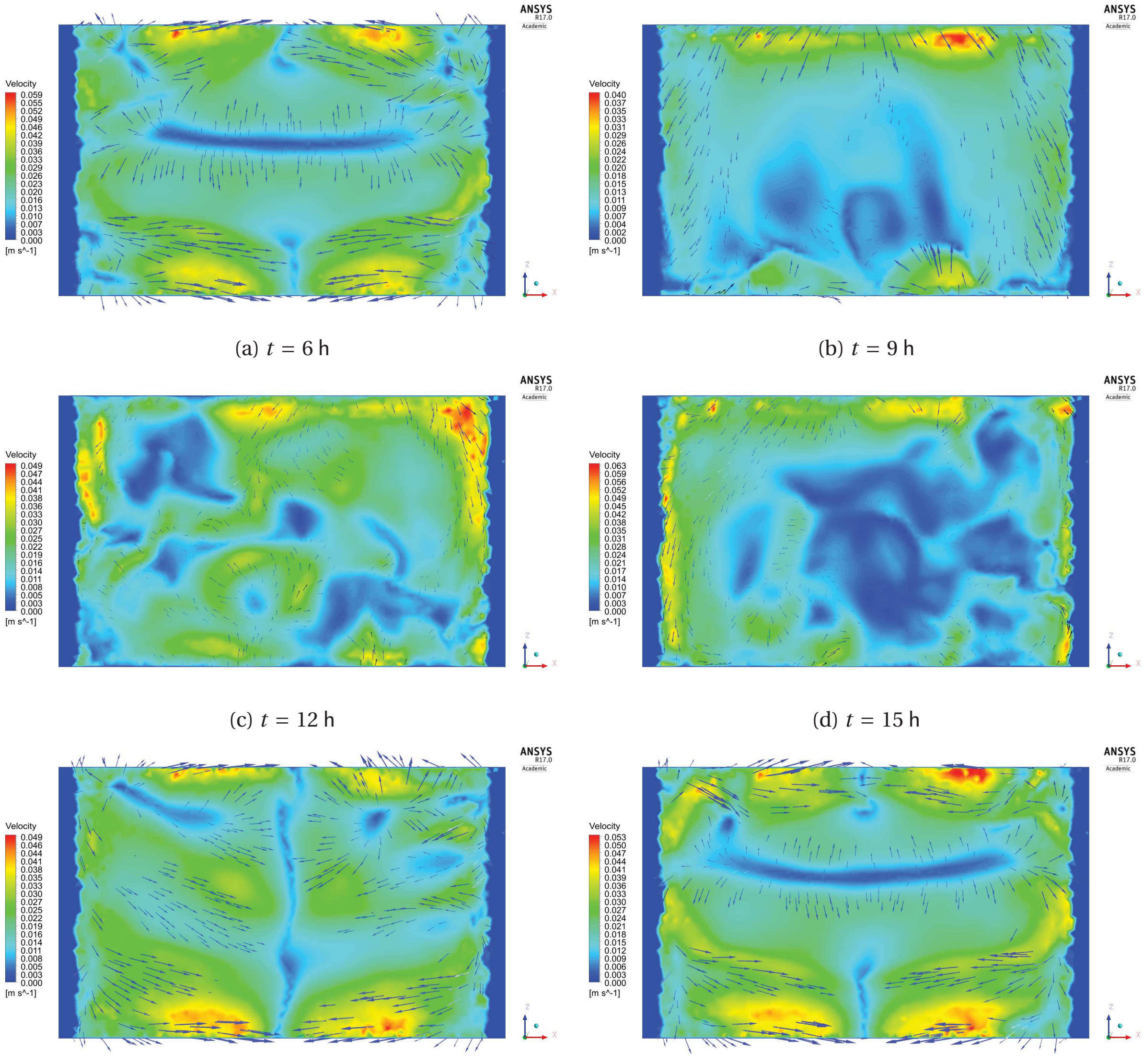
(d) $t = 15$ h



(e) $t = 18$ h

(f) $t = 21$ h

Figure 1.17: Contour plots of the room zone ceiling local range temperature (Winter)



(a) $t = 6$ h (b) $t = 9$ h (c) $t = 12$ h (d) $t = 15$ h (e) $t = 18$ h (f) $t = 21$ h

Figure 1.18: Contour of velocity magnitude, overlaid by the velocity vector field. Calculated at a vertical plane 2-cm away from the attic floor surface (Winter).

Chapter 2

An artificial intelligence based method to efficiently bring CFD to building simulation

Paper published online in the Journal of Building Performance Simulation on 12-jan-2018.
DOI: 10.1080/19401493.2017.1414880
ISSN: 1940-1493 (Print) 1940-1507 (Online) Journal homepage: http://www.tandfonline.com/loi/tbps20
Impact Factor: 2.874, Qualis: A2. Scimago: Q1

2.1 Introduction

The accurate computation of heat transfer phenomena in buildings is a topic where much attention has been paid since the seventies, when the first building simulation programs such as DOE-2 [3], BLAST [4] and ESP-R [5] emerged due to the energy world crisis.

Presently, there is a wide variety of software to simulate an entire building or a specific component. In the latter case, the software is commonly based on more complex mathematical models, where advanced physics are taken into account. On the other hand, there are the Building Performance Simulation (BPS) tools, the only ones capable to perform whole-building energy simulation, where different physical phenomena need to be simultaneously simulated. As a general characteristic of all BPS, it can be noted the adoption of simplified mathematical models, where advanced physics are neglected. For instance, most BPS tools consider only one-dimensional heat transfer through the envelope to calculate the indoor conditions and energy consumption as reported in [6, 31–37]. Although these simplifications can result in a loss of accuracy, the whole-building simulation can be performed in a reasonable computer run time.

If a more accurate assessment of building energy efficiency is desired, one solution proposed in the literature is the so called *co-simulation* [8, 9], also known as external coupling [12]. This strategy is a computational procedure that connects BPS tools with specialized software equipped with models based on advanced physics, so that it can be seen as an evolution of BPS tools.

Among the examples arising from the co-simulation, one can mention the reduction on emission of greenhouse gases and the substantial improvements on health, comfort and productivity [10]. These improvements could be achieved with the help of a BPS tool when buildings and their systems are treated as a complete optimized entity. Co-simulation enables integration among specific systems and BPS tools. According to [11], the tool development team needs to have an in-depth knowledge of the software architecture, programming language, modelling approaches and strategies. Furthermore, the fragmented development and rapid innovations in building and system technologies, makes difficult the software maintenance and their extension.

Aiming at the improvement of BPS analysis with advanced physics, including multidimensional effects and airflow, this work presents two contributions applied to the phenomenon of natural convection in an attic. First, it has been shown how the zone temperature prediction can be highly improved in terms of accuracy with the co-simulation between a BPS and a Computational Fluid Dynamics (CFD) tool, where advanced physics numerical algorithms are available. The simulation is performed on a triangular-prism-shaped attic model, and the results show that although a good improvement in terms of accuracy is achieved, the technique presents as main drawback the high computational burden. Then, the second and the main contribution of the paper is the proposition of a method to obtain more robust and faster numerical models associated to the physical phenomena. More precisely, for a specific physical phenomenon, the approach consists of designing a new model, called prediction model, capable to provide results, as close as possible to the ones provided by the complex model, with a much shorter computer run time. The synthesis of the prediction model is based on artificial intelligence, being the main novelty of the paper. Basically, the prediction model is constructed by means of a learning procedure, using the input and output data of a co-simulation where the complex model is being used to simulate the physics. Then, the synthesized prediction model replaces the complex model with the purpose of reducing significantly the computational burden with a low impact on the accuracy of the results. Technically speaking, the learning phase is performed using a machine learning technique, and the model investigated here is based on a recurrent neural network model, whose features and performance are investigated on the attic model previously mentioned. The co-simulation are performed between `CFX` and `Domus` programs, and promising results are shown in terms of both accuracy gain and run time reduction.

The manuscript is organized as follows. In Section 2.2, a brief overview of advanced physics based on the use of CFD in buildings is presented. Section 2.3 shows how co-simulation techniques enable the use of Computational Fluid Dynamics (CFD) tools coupling with a BPS tool. A case study is presented in Section 2.5. The new concept - henceforth called *intelligent co-simulation* - is presented in Section 2.4, with the same case study considered in Section 2.5. Annex 2.A presents more details about Domus and CFX tools.

2.2 Literature Review

2.2.1 Computational Fluid Dynamics (CFD)

Air lumped models and unidimensional diffusive transfer models through building envelopes have been extensively used since the seventies until nowadays. Although these models demand a low computational cost and represent well the physics in many cases, they may not be appropriate in some cases, for instance, when high gradients of temperature, air velocity or pressure in one or more dimensions cannot be disregarded. In this context, Computational Fluid Dynamics (CFD) is an appealing alternative to be brought to the building simulation area to overcome the limitations of common models extensively applied in this field. However, CFD is still barely used due to their complexity and high computational cost to run unsteady simulations of large physical domains over time periods as long as one year. Therefore, their application in the building simulation area is still very limited, although CFD simulation research in buildings has been reported since the nineties and the reader may refer to [13], [14], [15], [16], [17], [18], [19] and references therein. For instance, [18] presented a BPS tool integrated to a CFD code for the definition of convective heat transfer coefficients. [20] used a BPS to determine the boundary conditions for a CFD coupling simulation, where the CFD tool provided the data obtained from the simulation of the atrium model, such as airflow and heat transfer coefficients, to the BPS tool. [21] described coupling methods between CFD and BPS tools and when they must be used. [22] also proposed a guideline for selection of simulation tools for airflow prediction, addressing approaches representing different solutions such as Zonal Airflow Network and CFD. As a general observation about co-simulation involving CFD tools, one can say that it enables the consideration of physical phenomena normally disregarded in BPS software due to the complexity of solving Navier-Stokes formulation and turbulence modeling that are the main factors for the high computer run time ([38]). More recently, [39] developed a coupled dynamic simulation that, using a Fast Fluid Dynamics (FFD) simulation program, simulates the dynamic interaction between the room airflow, HVAC, building envelope and feedback control.

Despite the significant evolution of computer hardware and CFD tools since the seventies - when the first building energy simulation tools appeared -, the CFD application in the building simulation field is still very restricted to a few researchers and a few consultants in the building sector. The reasons for this fact are the lack of well coupled numerically robust and user-friendly tools and, more importantly, the high computer run time demanded even for buildings described for small geometries. As contributions, this paper first highlights the importance of using CFD coupled to a BPS tool in terms of accuracy and then introduces a strategy to improve the main drawback of the co-simulation approach, reducing the CFD coupling to just a small portion of the total period that must be simulated.

2.2.2 Learning techniques

Neural Network

Artificial neural networks (ANN) were created with the purpose of creating a computational learning procedure similar to the learning process of a human brain. The main objective is the generalization of complex rules and mathematical routines, learning the key information patterns within a multidimensional information domain. ANNs operate like a "black box" model, requiring no detailed information or property of what is inside the box [40]. There are two major types of neural networks, feedforward and recurrent (see [41] for a short characterization of these two types of ANN). In the feedforward network, the input data goes through the network in a one-way direction to generate an output result. A recurrent neural network (RNN), on the other hand, has one or more, cyclic path of synaptic connections. While the feedforward network has the ability to implement only static input-output mappings (functions), with recurrent neural networks it is possible to implement dynamic systems. Another important feature of neural networks is the presence of the activation functions. These functions are placed before each neuron and determine the activation of the neuron. The activation is dependent on the network input and threshold value. There are several possibilities to choose activation functions and some of the most commons are presented in [42].

The RNN is a class of ANN with the presence of cyclic connections in its neurons. This structure provides to RNNs the characteristic of having *memory*, where the past input data can influence the actual network output. There are two subtypes of RNN related to the training approach: supervised and unsupervised training. In the supervised training, there is a presence of target data to guide the training process, while, in the unsupervised one, there is no such target data.

As described in [43], neural network learning could be used as an universal function approximation, moreover adding more hidden neurons is equivalent to adding more basis functions. Other parameters like the number of layers, the number of training samples, the length of learning period, the choice of neuron activation functions, and the training algorithm could also affect the training accuracy. Interested readers may refer to [42, 44] for a better comprehension on NN and RNN.

Neural Network in building physics

The neural networks can be applied in many areas like engineering, applied mathematics and computer science. One example of application in building physics is presented in [40] that used an ANN to learn how to predict the required heating load of buildings with the minimum of input data. In order to forecast building energy consumption, [45] made a comparison between a simple model based on artificial neural network and a model based on physical principles as an auditing and predicting tool. More recently, [46] adopted a multidimensional model, very time-consuming, to calculate the coupled heat, air, and moisture transfer through a building envelope. The data provided by the analytical algorithm was then used to train a neural network model and predict temperature, vapor pressure, and relative humidity profiles.

2.3 Domus-CFX Co-simulation

2.3.1 Definition

Co-simulation can be technically defined as a type of simulation where at least two simulation tools - each one offering different numerical solutions to a physical or mathematical problem - jointly solve differential-algebraic systems of equations and exchange data while the coupling is active ([47],

[48], [49]). There are several co-simulation approaches such as the one-to-one, middleware or a standard interface such the Functional Mock-up Interface (FMI) ([50]). The one-to-one and FMI approaches are implemented in the Domus simulation program to allow its coupling with other models and tools. For the co-simulation between Domus and CFX, is used the one-to-one coupling based on scripts. The type of coupling is the Ping-Pong method ([8]), creating a weak coupling between the tools. Domus has been chosen as the BPS tool because the authors have been worked in the evolution of the software since long time, having access and full understanding of the source code. However, in principle any other BPS tool could be used to illustrate the advantages of the proposed methodology.

2.3.2 Domus-CFX co-simulation for a dwelling attic

Figure 4.12 illustrates the communication between Domus and CFX, where Domus is the master of the simulation. At the initial stage, the building envelope thermophysical properties (ρ, c, λ) and the boundary conditions are informed to CFX. The latter computes a steady-state simulation step as initial conditions. The boundary conditions consist of the internal zone temperature (T_i), external weather temperature (T_e), the outside and inside convective heat transfer coefficients (h_i,h_e) and the external and internal heat fluxes (q_e,q_i). At each time step, Domus provides the outside and inside boundary conditions (T_i,T_e,q_e,q_i,h_i,h_e) for each surface to CFX.

The CFX simulates the whole 3-D attic model, considering the natural convection in the air domain and the 3-D diffusive heat transfer in all attic solid parts. After the successful run, CFX returns to Domus the spatially averaged surface temperature ($\bar{T}_s$) of both ceiling materials. Then Domus uses these temperatures to run the room air simulation. Figure 2.2 illustrates the exchanged data between the two programs. Black arrows indicate the boundary conditions transmitted from Domus to CFX, while the red one indicates the CFX output received by Domus.

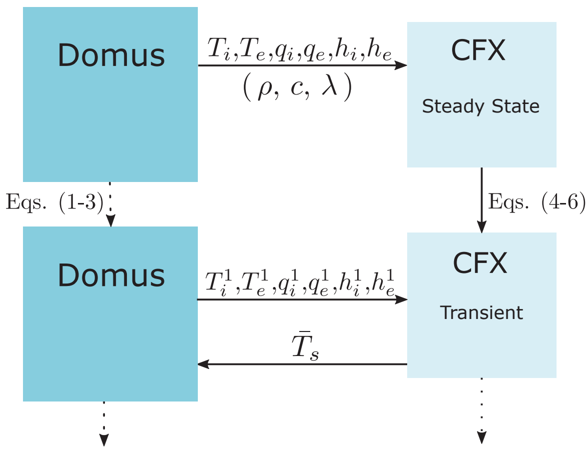


Figure 2.1: Co-simulation scheme between Domus and CFX.

The CFD coupled model receives non-uniform boundary conditions during the run-time period of simulation. This coupling requires the pre-definition of a template of the target geometry, as well as the configuration of the mesh by the user. The information transmitted to the CFD tool, such as air temperature, incident radiation flux and convective heat transfer coefficients, are from the previous time-step simulation. After converging the CFD model, the results are transmitted back to Domus, that uses the data to complete the time step as well as the energy and mass zone balances.

The following radiative heat fluxes calculated by Domus are used by CFX to be added to the convective heat flux boundary condition:

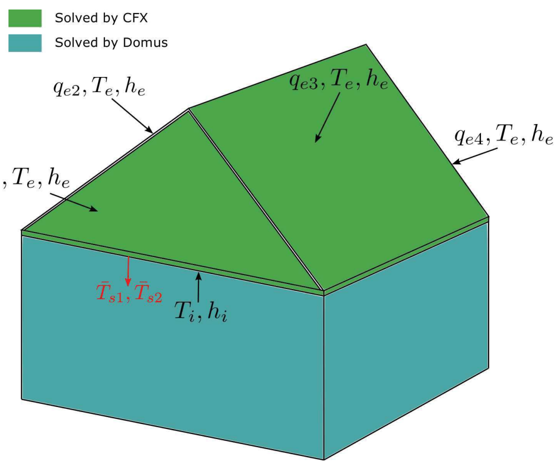


Figure 2.2: Diagram indicating Domus and CFX solving parts and the exchange data between the two programs.

$$q_{lw} = -\varepsilon\sigma F_f(T_{Sup}^4 - T_{Ref}^4), \tag{2.1}$$

$$q_{sw} = q_{dir} + q_{diff} + q_{refl}, \tag{2.2}$$

$$q_e = q_{lw} + q_{sw}, \tag{2.3}$$

where q_{lw} is the long-wave radiation flux (W/m^2), ε is the emittance of surface, F_f is the view factor from surface to the sky or horizon (in case of horizontal surfaces), σ is the Stefan-Boltzmann constant (5.67 x 10^8 kg$s^{-3}K^{-4}$), T_{Sup} is the surface temperature in Kelvin and T_{Ref} is the sky temperature or an average temperature if the surface is vertical, and is given by:

$$T_{Ref} = (T_{sky} + T_{ground})/2. \tag{2.4}$$

The q_{sw} is the short-wave radiation heat flux (W/m^2) as well as q_{dir}, q_{diff} and q_{refl} are the direct, diffuse and ground reflected radiation fluxes (W/m^2), respectively.

2.3.3 Co-simulation Verification

The Co-Simulation between `Domus` and `CFX` using the one-to-one approach has been verified by the analysis of a simple monolithic homogeneous wall. The validation procedure consists in using the `Domus` wall heat transfer model for simulation of a single zone with all adiabatic walls except the one used in the co-simulation. Then, the results are compared with the co-simulation between `Domus` and `CFX`.

The CFD model consists of a rectangular block composed only of concrete material (Table 2.1), 250 cm^2 area (50 cm x 50 cm) and 20 cm thickness. Due to the simple characterization of this geometry, a hexagonal mesh of approximately 10 thousands nodes guarantees an excellent quality of elements of the mesh. The boundary conditions used in the co-simulation are the projected external solar radiation flux, the external air temperature, the indoor temperature and the convective heat transfer coefficients at both sides of the wall. The output of the model is the average internal surface temperature, which is transmitted to `Domus` and used to determine the variation of the internal temperature of the thermal zone. The weather file used is from the city of Curitiba, Brazil, and the indoor temperature varies freely.

Table 2.1: Properties of the material used in the verification.

Parameter	**Material**
Density (kgm^{-3})	2200
Specific heat ($Jkg^{-1}K^{-1}$)	1000
Thermal conductivity ($Wm^{-1}K^{-1}$)	1.75

The building model in Domus is a 20.7 m^3 (3 m x 3 m x 2.3 m) single thermal zone with a 6.9 m^2 surface over analysis. The internal and external convective heat transfer coefficients are, respectively, 3 W/m^2K and 12 W/m^2K. The material properties and thickness of the wall are the same in the Domus standalone simulation and in the CFD model. The Domus time-step is 20 minutes, that is also the co-simulation time-step where the information between Domus and CFX is exchanged. The CFD tool time-step is 60 seconds.

The results show a good agreement as noticed in Figure 2.3a. Figure 2.3b shows the root-mean-square (RMS) comparing the two approaches. The mean error in the evaluated period is 1.5e-03, which is very reasonable considering that the two numerical models are very distinct. Therefore, the coupling between Domus and CFX appears as reliable.

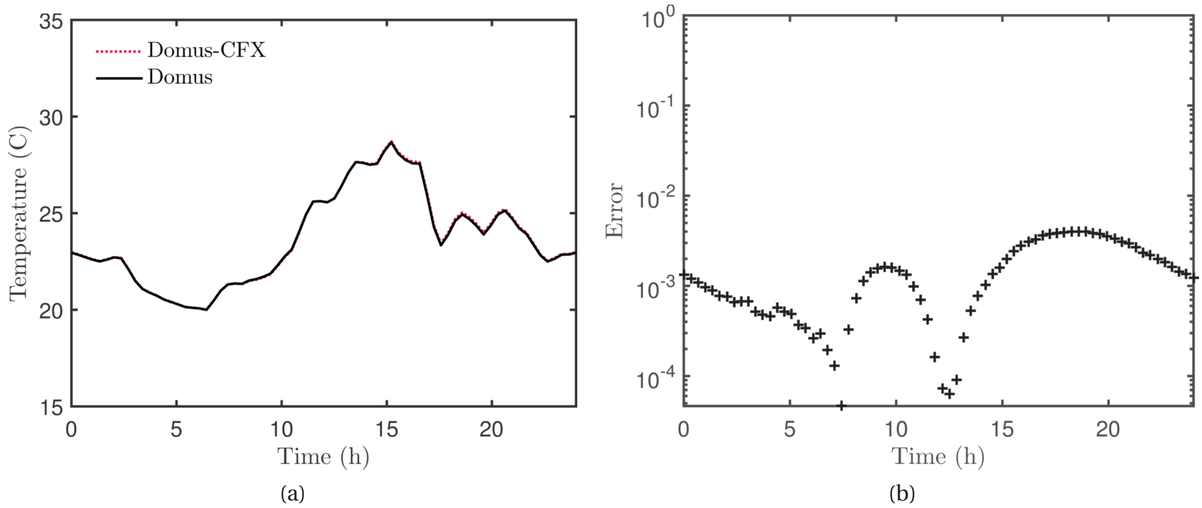


Figure 2.3: Comparison of room air temperature (a) and the error (b) between Domus and Domus-CFX.

2.4 Intelligent co-simulation strategy

2.4.1 General statement

The results of the investigations presented so far made clear the effect of using the simplified models available in the current BPS tools to represent the physical phenomena, even considering a simple case study, with only natural convection and heat conduction. Differences near to 5 °C were observed in comparison with the results of the co-simulation with the CFD tool. When evaluating such information, it is notorious the importance of using co-simulation approaches to correctly represent the phenomena involved in buildings physics. Although the co-simulation seems to be the most accessible alternative currently available to obtain precise results (thanks to advanced physics), on the other hand, however, the high computational cost is certainly an important issue that must be addressed.

With the purpose of alleviating the high computational cost demanded by the co-simulation, at this point it is introduced the intelligent co-simulation, that is the second and main contribution of the paper. In the context under investigation, the strategy relies on performing the co-simulation in its standard setup (`Domus` connected to `CFX`) only during a short period of the whole simulation. In this phase, called learning (or training) period, a new mathematical model is synthesized to represent the physical phenomenon that is being simulated by `CFX`. The synthesis is performed using a machine learning technique, that basically observes the input and output data generated by the co-simulation in this short period and trains a neural network model, called prediction model. Thus, `CFX` is disconnected from the co-simulation and Domus performs the rest of the simulation using the prediction model, being much faster to simulate with a low impact on the accuracy of results. One of the highest points of this approach is that the whole process can be concentrated in a single building performance simulation tool. Moreover, the intelligent co-simulation can be transparent to the user, being performed without requiring a great interference of the user. In the study described below, a Recurrent Neural Network (RNN) is used as the prediction model and the case study discussed so far is evaluated again using the intelligent co-simulation.

2.4.2 Co-simulation training a RNN

The RNN model used in this work is a supervised single hidden neuron layer with a context layer that gives feedback from the previous outputs. This RNN structure and the feedforward method was implemented in C++ programming language, however, the training is performed by the Matlab Neural Network toolbox, using the Bayesian Regularization (BR) training algorithm. The BR updates the weight and bias values according to Levenberg-Marquardt optimization. Furthermore it determines the correct combination of minimized squared errors and weights, so delivering a network with better quality of generalization [51, 52].

Training phase

The training phase is performed during the co-simulation procedure between the tools. As a master, `Domus` collects the data exchanged during the co-simulation, stores the input parameters provided for the slave tool (named as input dataset) and returns the calculated data (named as target dataset).

The size of the training phase must be defined by the user at the beginning of the simulation. This definition is based on days of simulation, with 2 days being the minimum required period. With the data obtained from the CFD co-simulation, the last co-simulated day is used to test the RNN and the remaining data is used to train the model.

An algorithm was developed to estimate the best number of hidden neurons for each input set of data. The structure of the training algorithm proceeds with 2 loops where 70 training sessions of the desired network is performed. The first loop corresponds to the number of nodes in the network, ranging from 1 to 7. The second loop is the number of iterations, set to 10, which is the number of trainings that are performed for each number of nodes. Briefly, 10 new networks are trained for each number of nodes from 1 to 7. At the end of each training, the neural network generated is tested in the prediction of the data over the test set. Of the 70 new networks created, the one with the smallest RMS error with respect to the test set is preserved, while the others are discarded. The `Matlab` training phase is concluded with the transmission of the weights calculated to the network with the best results, as well as the number of network nodes, to `Domus`, by means of .txt file. With this data, `Domus` generates its own neural network with the received weights, completing the training model ready to perform the outgoing predictions. The illustration of the intelligent co-simulation is shown in Figure 2.4a for the training process and in Figure 2.4b for the prediction phase. In Table 2.2, it is possible to observe the link between the tools for each phase.

Prediction phase

In the prediction phase, the co-simulation is then deactivated, and no new access to the CFX tool is performed. The Domus RNN model, updated with the calculated weights, replaces the CFD model in the simulation. For each time step, the input parameters are given to the prediction model, which returns the desire prediction target.

Before feeding the prediction model with the input data, these data need to be scaled, to the -1 to 1 range, due to two reasons. The first one because Matlab uses to automatically scale the data for training the model. If the data in the prediction phase is not scaled, they would be out of the range used during the model training, resulting in a bad prediction capacity. The second reason, as addressed in [53], is the good practice to scale the actual data into dimensionless data or into required space (range). The activation function chosen to the neural network can define the choice of the range of the data scaling. As the Domus-RNN model has the hyperbolic tangent sigmoid transfer function (Equation 2.5), as the function to calculate the hidden layer output, it requires the range -1 to 1 for the input data. It is also necessary to scale in the opposite direction for the prediction model output data.

One advantage of the neural network prediction model is that after completing the training it is possible to store the trained model (weights) and use it in future simulations, without the need of new calls to the co-simulated tool.

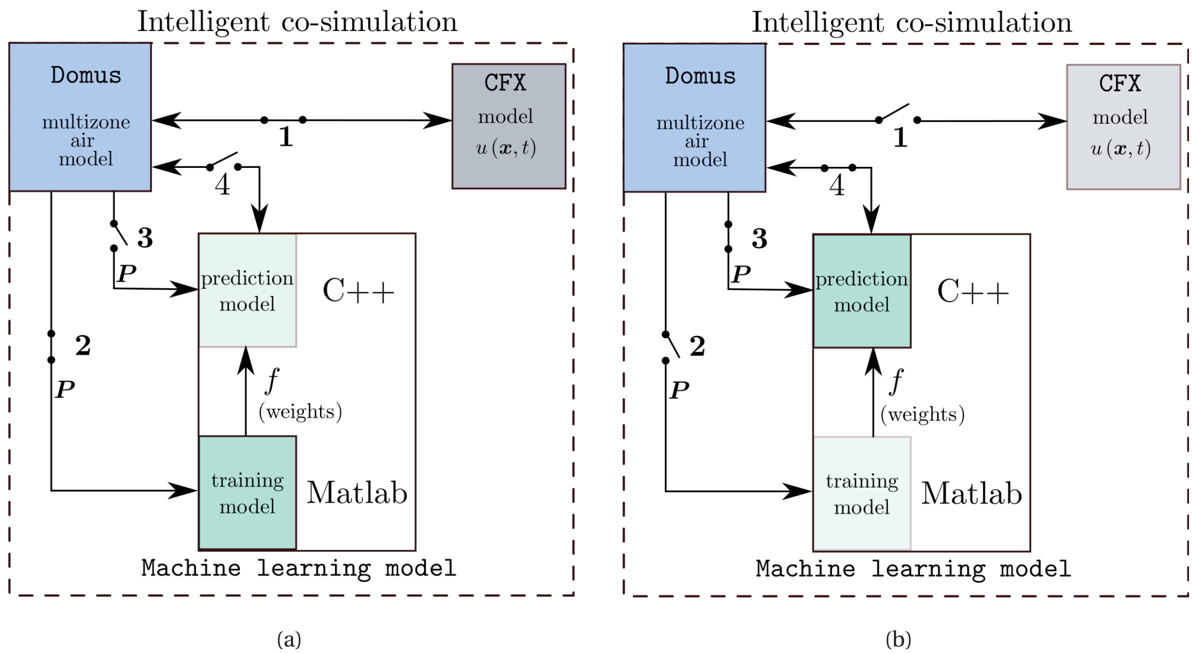


Figure 2.4: Training (a) and prediction (b) phases.

The number of inputs and outputs is directly related to the CFD model used to train the RNN.

Two activation functions have been selected, Hyperbolic Tangent Sigmoid (Eq. (2.5)) for the output of the Hidden Layer and Pure Linear (Eq. (2.6)) function for output of the Output Layer.

Table 2.2: Link between tools and model in each phase.

	Training phase	Prediction phase
1	ON	OFF
2	ON	OFF
3	OFF	ON
4	OFF	ON

$$tansig: u = \frac{1-e^{-2n}}{1+e^{-2n}}, \tag{2.5}$$

$$pureln: u = n. \tag{2.6}$$

Within the Recurrent Neural Network, the solution of the problem is approximated by:

$$uh_k(t+dt) = f^1\big(b_k^1 + \sum_{i=1}^{M} W_{ki}^1 X_i + b_k^c + \sum_{i=1}^{N} W_{ki}^c uh(t)\big), \tag{2.7}$$

$$u(t+dt) = f^2\big(b^2 + \sum_{i=1}^{N} W_i^2 uh_i(t+dt)\big), \tag{2.8}$$

where u is the model output, N is the number of neurons in the hidden and context layers, M is the number of inputs, X_M is the input vector, f^1 is the transfer function in the hidden layer (tansig), W_{NxM}^1 are the weights in the hidden layer, W_{NxN}^C are the weights in the context layer, f^2 is the output layer transfer function (pureln), W_{Nx1}^2 are the weights in the output layer and b_{Nx1}^1, b^2 and b_{Nx1}^c are the bias for the hidden, output and context layers, respectively. As initial condition, $uh(0)$ is equal to the input vector.

2.5 Case study

The case study comprises a building with two thermal zones (Figure 2.5), a conventional 96m^3 room and a 33.32 m^3 triangular-prism-shaped attic with a height between the base and the top of the attic of 2.7 m , as depicted in Figure 2.6. The aim is to evaluate the effect of natural convection in the attic on the temperature of the room zone. To analyze the isolated effect of the attic on the temperature of the room, an adiabatic condition was assumed for all the other walls and floor of the room. The ceiling and the attic temperatures were evaluated by sensible heat transfer models using a co-simulation between `Domus` and `CFX`. The CFD model was responsible for calculating the entire attic and the transient heat flux to the room zone by means of the average inner temperature of the ceiling surfaces. The thermal parameters of different materials used in the simulation are listed in Table 4.1. Simulations were performed using a computer equipped with an Intel Core i7 4790 Processor with a four-core CPU at 3.6 GHz. The `CFX` used two cores for the simulation.

It is expected a reasonable influence of natural convection due to the asymmetric behavior generated by the considerable differences in the boundary conditions.

A CFD adaptive mesh was generated for the attic geometry. The Grid Convergence Index (GCI) calculation was performed within the limitations imposed by the computational capacity available. Eight mesh models with 10, 30, 60, 100, 200, 300, 450 and 600 thousand nodes were generated and evaluated. The GCI results (Figure 2.7) enable to conclude that a mesh of 450 thousand nodes presents good mesh independent results, since a greater refinement does not bring considerable difference in the results of the model. Figure 2.8 shows the 450 thousand node mesh generated by ANSYS Mesh.

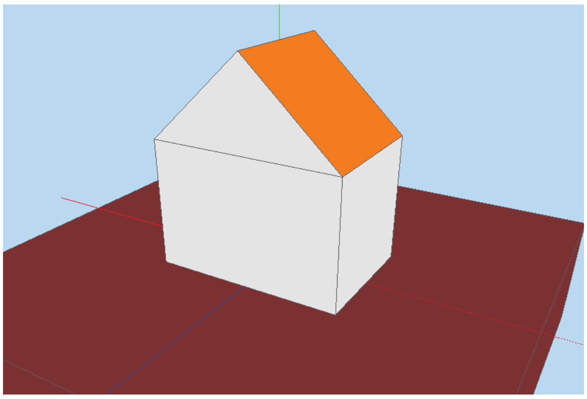


Figure 2.5: Case study in the Domus graphical interface.

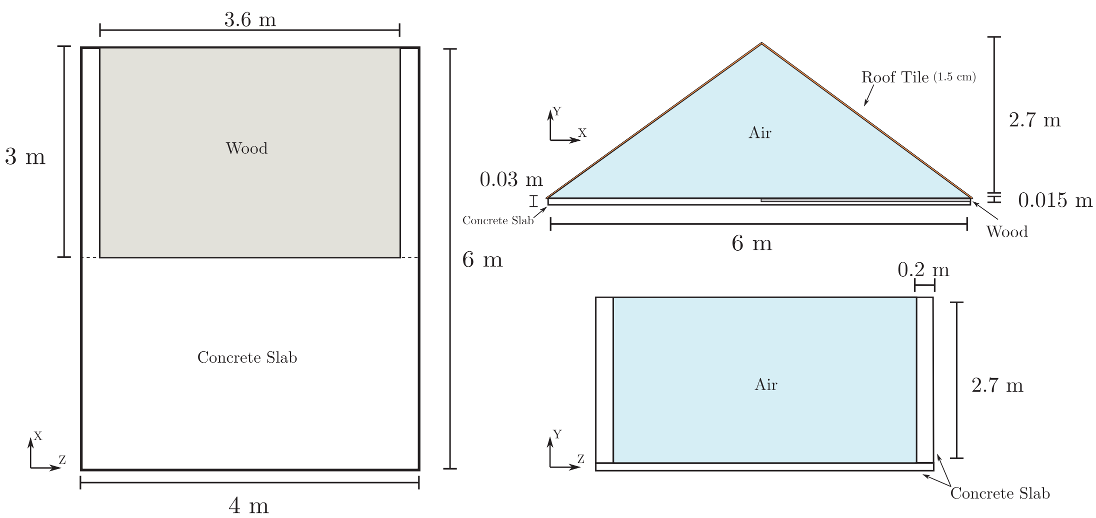


Figure 2.6: Diagram of the attic model in the case study.

Table 2.3: Range of parameters used in simulation.

Parameter	**Material**			
	Air	Roof Tile	Concrete Slab	Wood
Density (kgm^{-3})	1.200	2000	2200	720
Specific heat ($Jkg^{-1}K^{-1}$)	1007	920	1000	1255
Thermal conductivity (Wm^{-1}K^{-1})	0.024	1.05	1.75	0.16

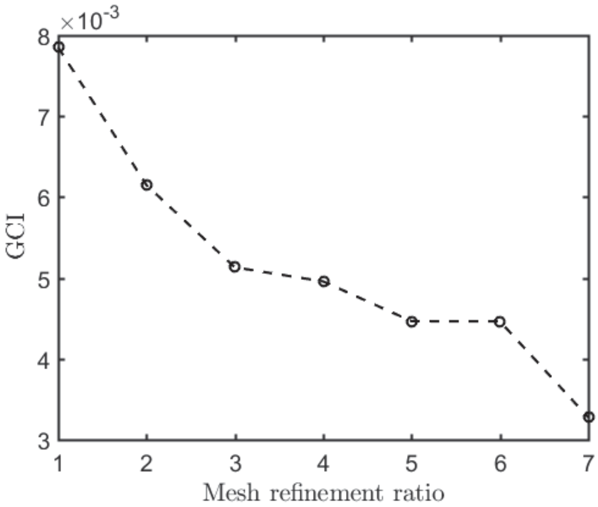


Figure 2.7: GCI value as a function of the mesh refinement ratio.

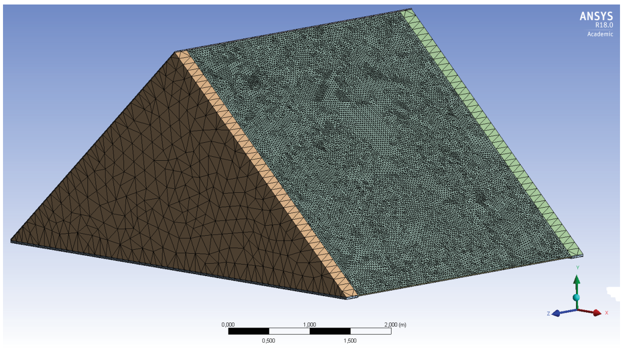


Figure 2.8: Attic mesh generated by ANSYS Mesh.

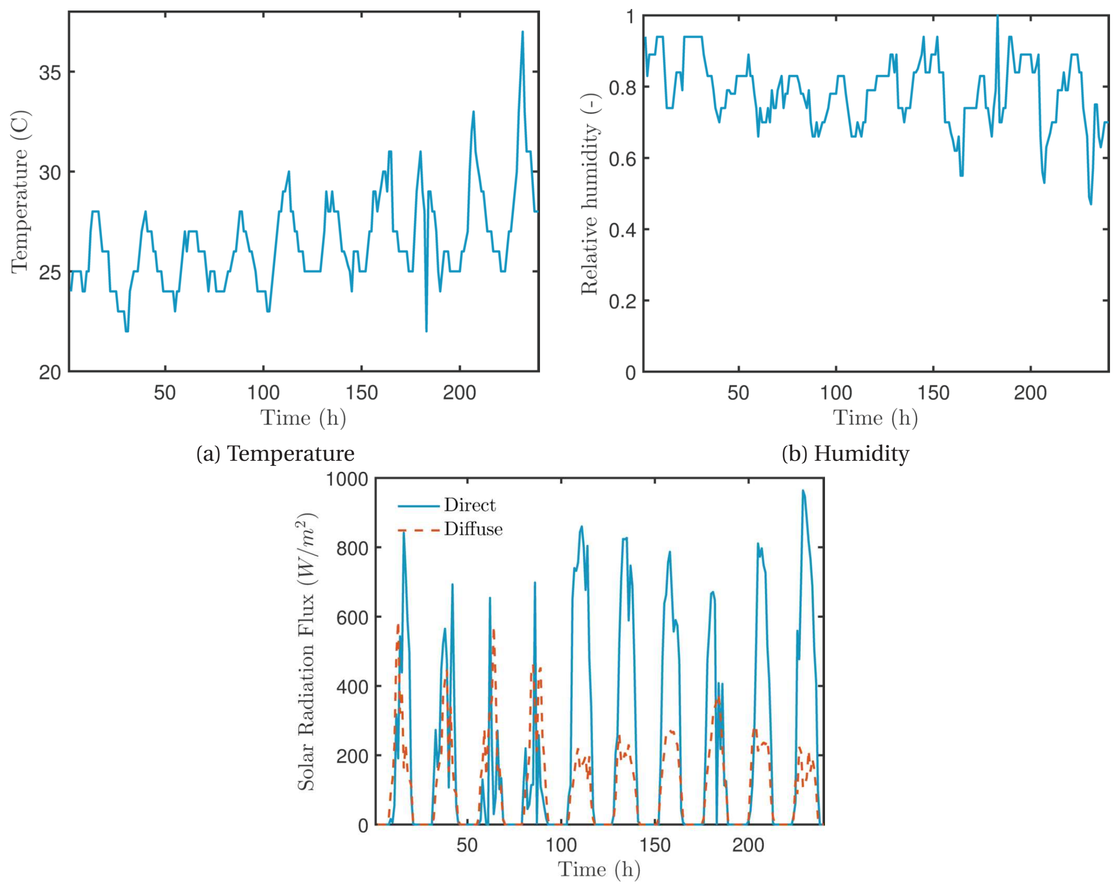
2.5.1 CFD Attic model

The CFD model convergence criterion was defined as a residual level of 10^{-3}, considered as a loose convergence criterion but sufficient to fulfill the objective of this case study, as the attic model represents mainly the natural convection phenomenon and there is no air exchange or infiltration into the attic.

The initial temperature in both zones is $T_i = 20\ °\mathrm{C}$. The convective heat transfer coefficients are set to $h_i = 3\ \mathrm{W/(m^2.K)}$ and $h_e = 12\ \mathrm{W/(m^2.K)}$. The h_i inside the attic is calculated by the CFD model via the co-simulation approach. The outside conditions represent a temperate climate of Rio de Janeiro, Brazil. The inside temperature and relative humidity vary freely. Due to the much high computational cost, a ten-day simulation was performed. Climatic data of the first ten days of January, taken from the `Domus` database, were used and are shown in Figures 4.6a - 4.6f.

2.5.2 Co-simulation results

The room air temperature evolution is shown in Figure 2.10a using two different approaches, while Figure 2.10b presents the RMS errors using the advanced approach (`Domus-CFX`) as the reference value. The standalone solutions presented a maximum difference of 4.2°C with respect to `Domus-CFX` model, remembering that only the attic was co-simulated with `CFX` due to the high computer run time. `CFX` solves three-dimensionally and with high accuracy the energy, momentum and



(a) Temperature

(b) Humidity

(c) Radiation

Figure 2.9: Outside conditions for temperature, relative humidity and solar radiation flux.

mass conservation differential equations, while Domus uses a lumped approach for solving an integral formulation - based only on energy and mass balance equations - and disregard the spatial variation of the temperature and air velocity fields that promote higher temperature predictions due mainly to an increase on the convective and corner effects.

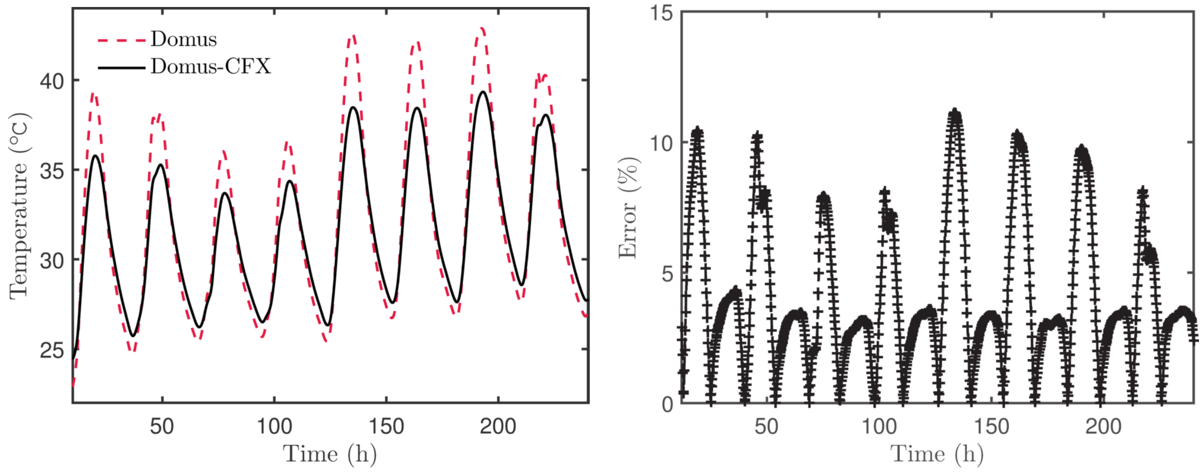
In Figure 2.11, the airflow effect on the temperature field within the attic can be observed due to the multidimensional thermal asymmetry of two different materials on the attic floor - the wood lining and the concrete slab - that causes visible differences in the air temperature field within the attic, emphasizing the contours presented in Figures 2.11b and 2.11d.

Figure 2.12 illustrates the contour plots of air temperature and velocity fields, on a layer at half the attic length. An important point to be considered is the asymmetry of the radiative heat flux due to the solar radiation, which imposes a high multidimensional effect on the heat flux and magnifies the natural convection within the attic.

Table 2.4 shows the computer run time for all performed simulations. Due to the higher computational cost of the simulation using the CFX coupling, only 10 days were simulated. The computer time, for both approaches, include the pre-processing steps like *pixel counting* [54] for definition of external solar radiation and view factor for calculation of internal long-wave radiation.

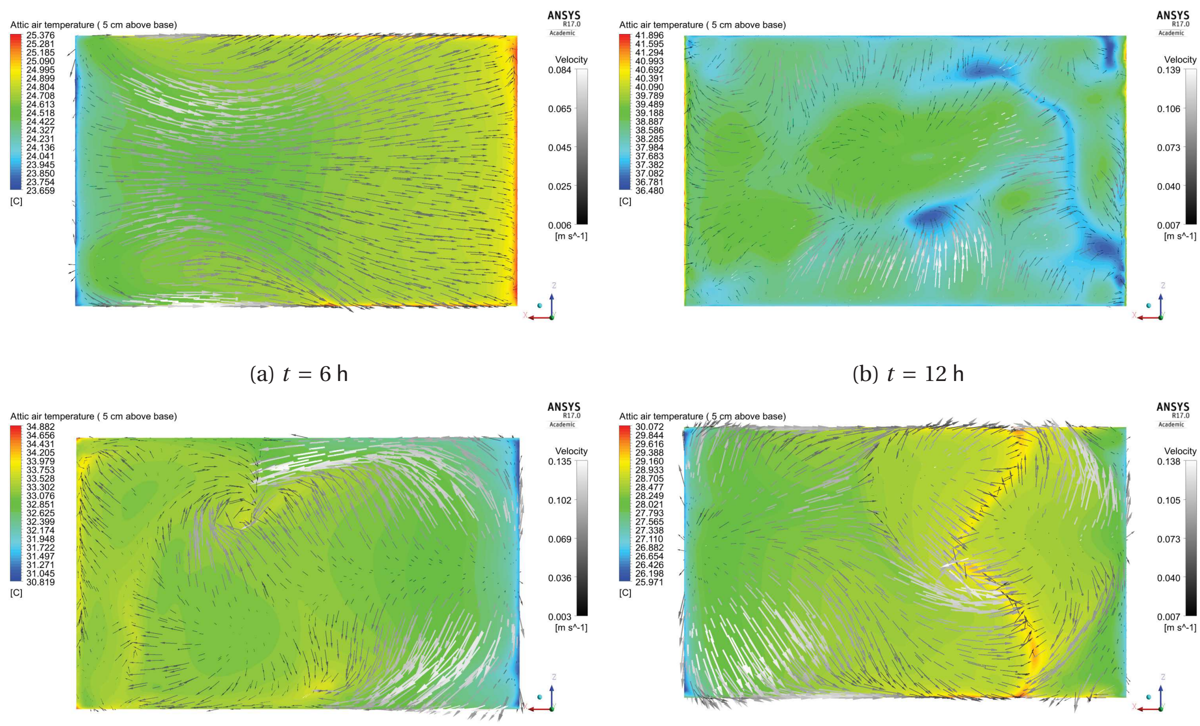
2.5.3 Applying Intelligent co-simulation methodology

This subsection aims to apply the concept of *intelligent co-simulation* in the co-simulation case study, focusing on the attic CFD model as the target to be replaced by the neural network model. The



(a) (b)

Figure 2.10: Room air temperature (a) and Room air temperature error (b).



(a) t = 6 h (b) t = 12 h

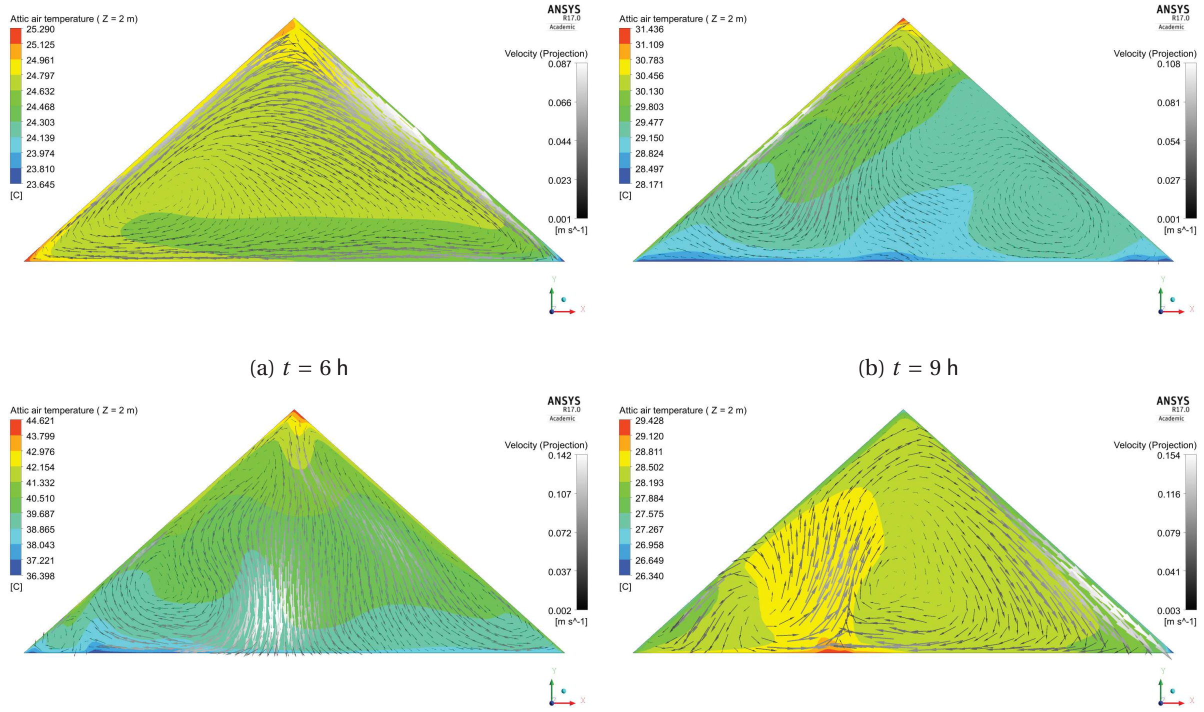
(c) t = 18 h (d) t = 21 h

Figure 2.11: Contour plots of the air local range temperature, overlaid by the velocity vector field. Calculated at a vertical plane 5-cm above the attic floor surface.

properties of materials, geometry and boundary conditions are the same. The simulation procedure has now two phases, training and prediction. The former performs data collection and training of the model, while the latter replaces the co-simulated model by the neural network prediction model.

Training phase

The training period is defined as the first days of simulation for the case study, more precisely 432 time-steps (72h) are chosen. This training period is defined by means of a numerical investigation,



(a) t = 6 h

(b) t = 9 h

(c) t = 12 h

(d) t = 21 h

Figure 2.12: Contour plots of the air local range temperature, overlaid by the velocity vector field. Calculated at a horizontal plane in the middle of the attic.

Table 2.4: Computational times for the performed simulations.

Approach	**Time (hr:min:sec)**
`Domus`	00:00:52
`Domus-CFX`	656:00:00

being the minimum period that presents a satisfactory result for the prediction phase. Eight inputs are selected in the attic boundary conditions. The training inputs are the weather temperature, zone temperature, sky temperature, ground temperature and four solar short-wave radiation fluxes, one for each external attic surface (indicated in Figure 2.5). The targets were defined as the two ceiling surfaces temperatures, knowing that the attic model has two different materials between the evaluated zone and the co-simulated space. As the case study uses a time step of 10 min, the test dataset (the last training day) has the size calculated as 144 data samples.

Prediction phase

In the prediction phase, the Domus RNN model, with the calculated weights, replaces the attic object in the simulation. For each time step, the eight input parameters are given to the prediction model, which returns the temperature of each ceiling region.

2.5.4 Intelligent co-simulation results

The room air temperature evolution is shown in Figure 2.13a using three different approaches: `Domus-CFX` as the reference value, `Domus` only and `Domus` with a RNN model representing the attic model. Figure 2.13b presents the RMS errors, using the advanced approach (`Domus-CFX`) as the reference value.

It is possible to notice that the error for the RNN model is lower than the error presented by the standalone simulation with `Domus` only. The standalone model provides a mean error of $4.4e^{-2}$, with a maximum difference of 4.2 °C in the zone temperature. On the other hand, the RNN model presents a 10 times lower error, with approximately $3.4e^{-3}$ of mean error, and a maximum zone temperature difference of only 0.39 °C in the prediction phase. In this case study, the training algorithm chose 6 neurons for the hidden layer.

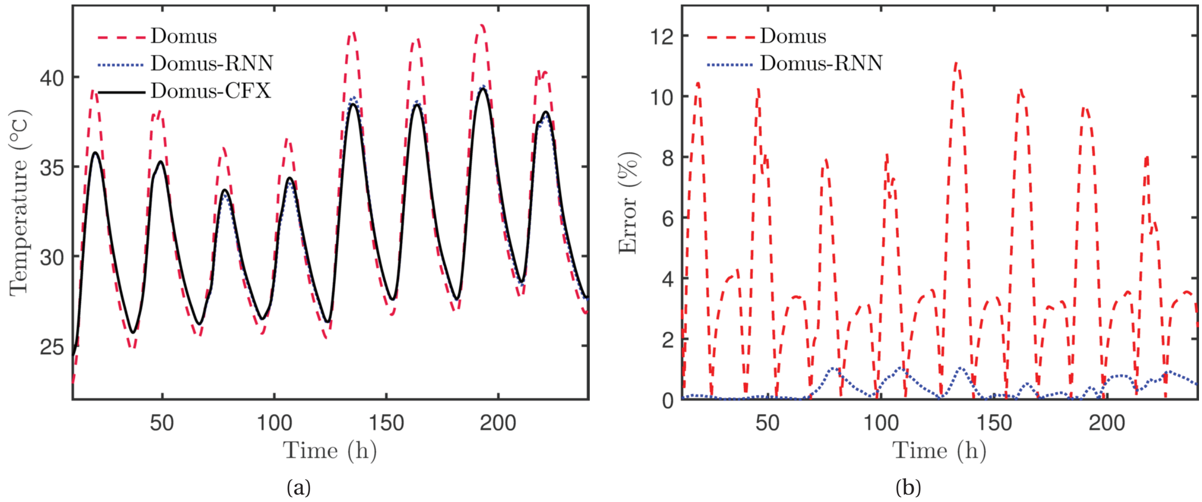


Figure 2.13: Room air temperature (a) and Room air temperature error (b).

Table 3.1 shows the computational time for the 10 days simulation. The main point is the comparison between the `Domus-CFX` and `Domus` with the RNN model, where three days of `Domus-CFX` co-simulation were used to train the RNN model. This computer run time information demonstrates how the employment of the RNN model can reduce the run time by a factor higher than six. In practical terms, more than 500 hours of simulation are saved. After performing the three-day training period in approximately 100 hours (Table 3.1 - `CFX` column), the RNN model completed the remaining simulation period in less than one minute (Table 3.1 - Prediction Model column).

After the training, with the prediction model trained and stored in the BPS tool, a new simulation was performed using only the prediction model (without the co-simulation). A whole-year period was simulated to obtain the computer run times and to compare them with the lumped model. Table 2.6 shows this comparison, where it is possible to observe that with a RNN trained model, the run time is even lower than the one obtained by the lumped model and with a great gain of accuracy. This result shows that the prediction model, constructed with the aid of a tool capable to perform a detailed simulation using advanced physics models (based on unsteady Navier-Stokes formulation), is not only useful to finish the simulation with great accuracy, but also to replace the lumped model in future simulations if similar geometry and external conditions are used.

2.6 Conclusion

Since the seventies, the accuracy of building performance simulation has been a topic of great concern. Building energy simulation programs usually present simplified mathematical models, such as lumped models for calculating room air temperatures, which might create considerable differences

Table 2.5: Computer run time for a 10 days period.

Approach	Total time		CFX		Domus	Prediction Model
	(h)	(min)	(h)	(min)	(min)	(min)
Domus lumped model	0.01	< 1	−−	−−	< 1	−−
Classical co-simulation	656	39360	656	39360	< 1	0
RNN-based intelligent co-simulation	101.71	6103	101.68	6101	< 1	< 1

Table 2.6: Computer run time for a one-year period.

Approach	Total time	CFX	Domus	Prediction Model
	(sec)	(sec)	(sec)	(sec)
Domus lumped model	11	−−	11	−−
RNN-based prediction model	10	0	6	4

in the representation of physical phenomena. As a matter of fact, this paper presented the simulation of a single-zone building with a triangular-prism-shaped attic, comparing the results of a purely lumped model and a CFD-coupled model. The differences appeared to be as high as 4.2°C for the building room air temperature. Such magnitude cannot be neglected, especially considering that the investigated case is well-behaved, with only natural ventilation. Higher differences can be expected if more complex phenomena such as natural ventilation, infiltration or air-conditioning systems are taken into account.

It was also shown that using co-simulation to obtain more accurate results is not a simple task, mainly due to the high computational cost. For the case study presented in this paper, the CPU time was increased 45000 times, which makes the use of unsteady CFD simulation not affordable in the field of building performance analysis, at least for the computational capacity of the computers available nowadays.

To provide a solution to the drawback associated to the high computational cost of detailed complex models such as unsteady CFD simulation, this paper introduced a strategy - called intelligent co-simulation -, which consists of constructing a new model, called prediction model, capable to provide results, as close as possible to the ones provided by the complex model, with a much lower computer run time.

In terms of accuracy, while the standalone lumped model provided high errors and room air temperature difference as high as 4.2°C, as mentioned above, the intelligent co-simulation model provided a RMS error as low as 0.34% (13 times lower than the one obtained with the standalone lumped model) and a maximum temperature difference of only 0.39°C. Once the model is trained, the accurate model can be used for yearly simulation, running even faster than the simplified purely lumped model. Therefore, those results indicate this new strategy seems to be very promising to accurately and rapidly bring some advanced modelling - such as those to represent multidimensional diffusive and convective heat transfer, natural ventilation and detailed HVAC systems - to building simulation tools.

However, a drawback still remains for the training period, where a complex model must be simulated to obtain the data set necessary to construct the fast and accurate prediction model. Although the training period is very short - a couple of days - unsteady 3D CFD simulations are costly. Some concerns also arise regarding the efficiency of prediction models to represent more complex phenomena than those addressed in this paper, such as airflow and forced convection.

As prospective themes for further investigations of the intelligent co-simulation technique, one can mention the use of representative climates in a pre-simulation training period, using artificially created climates to reduce training time. Other machine learning methods should also be investigated, comparing for instance, the technique based on model reduction - Proper Orthogonal Decomposition (POD) - with the RNN presented in this paper, with the purpose of reducing the training time period. The use of a standard interface such as FMI should also be considered as soon as a CFD tool that supports such functionality be available.

To finally conclude, the authors consider that, although the computational barrier certainly can be further reduced with more research efforts, the results presented in this paper show that the proposed intelligent co-simulation can be a significant step towards a new generation of algorithms and BPS tools.

2.A Simulation tools: Domus and CFX

2.A.1 The building energy simulation program `Domus`

The whole-building hygrothermal and energy simulation program `Domus` is used as the BPS tool, being able to compute the room air temperature using the lumped multizone model [7, 55] and to provide the boundary conditions to the CFD tool. The energy balance of the zones is performed considering heat sources from the walls, the windows, the occupants and the HVAC systems. The `Domus` lumped energy balance for a zone submitted to loads of conduction, convection, short-wave solar radiation, inter-surface long-wave radiation, infiltration, ventilation and HVAC system related loads, is given by the equation:

$$E_t + E_g = \rho_{air} c_{air} V_{air} \frac{dT_{int}}{dt}, \tag{2.9}$$

where E_t is the energy flow that crosses the room (W), E_g is the internal energy generation rate (W), ρ_{air} is the air density ($\mathrm{kg/m^3}$), c_{air} is the specific heat of air ($\mathrm{J/kg-K}$), V_{air} is the room volume ($\mathrm{m^3}$) and T_{int} is the room air temperature (°C).

Equation (2.10) shows the heat released by the internal porous surfaces of the building envelope for the sensible conduction load and, in Equation (2.11), for the latent conduction load.

$$Q_{wall,S}(t) = \sum_{i=1}^{m} h_{c,i} A_i \left[T_{i,x=L}(t) - T_{int}(t) \right], \tag{2.10}$$

$$Q_{wall,L}(t) = \sum_{i=1}^{m} L(T_{i,x=L}(t)) h_{m,i} A_i \left[\rho_{v,n,j}(t) - \rho_{v,int}(t) \right], \tag{2.11}$$

where A_i represents the area of the i-th surface, h the convection coefficients for heat (hc) and mass (hm), $T_{i,x=L}$ the temperature at the i-th internal surface of the considered zone, L the vaporization latent heat and ρ_v the water vapor density.

`Domus` has been used in a variety of studies as in [56] that presented integrated calculation of the hygrothermal behavior of indoor climate, building porous envelope and central HVAC system. [57] validate several cross and single-sided natural ventilation models implemented in the `Domus`. Photovoltaic panels are numerically coupled in `Domus` in the work presented by [58]. One last example is the study presented by [59] to analyse the modelling level needed to successfully evaluate the heat transfer through window glazing material in whole-building simulation.

Domus has databases of material thermophysical properties and weather files. It also has a one-dimensional model of heat and moisture transfer in porous materials (that has been disabled for this work). The whole program is written in C++ with an OpenGL based graphical interface, available at http://www.domus.pucpr.br/.

2.A.2 Computational Fluid Dynamics(CFD)

As described in [60], Computational Fluid Dynamics is a computer-based tool for simulating the behaviour of systems involving fluid flow, heat transfer, and other related physical processes. CFD works solving the unsteady Navier-Stokes equations of fluid flow over a region of interest, with specified conditions on the boundary of that region.

Navier-Stokes equations describe the processes of momentum, heat and mass transfer and are solved numerically solved. For Ansys-CFX, the solution of the Navier-Stokes equations is based on the finite-volume method ([60]).

As presented in [20], the formulation for mass, momentum and total energy balances for each control volume node are shown in Equations (2.12) to (2.14), respectively:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0, \tag{2.12}$$

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right) = \rho g_i - \frac{\partial p}{\partial x_i} + \mu_t\left(\frac{\partial^2 u_i}{\partial x_j^2}\right), \tag{2.13}$$

$$\frac{\partial(\rho h_{tot})}{\partial t} - \frac{\partial p}{\partial t} + \rho h_{tot} \frac{\partial u_i}{\partial x_i} = \lambda \frac{\partial^2 T}{\partial x_i^2} + \tau_{ij} \frac{\partial u_i}{\partial x_j} + u_i \frac{\partial \tau_{ij}}{\partial x_j}, \tag{2.14}$$

where ρ is the density, u_i the velocity in index notation, T the temperature, g the gravity acceleration, p the pressure, μ_t the turbulent viscosity, h_{tot} the total enthalpy, λ the thermal conductivity and τ the sub-grid scale stress.

Setup

The numerical solution method is defined as a high-resolution transient scheme that alternates between the use of second-order backward Euler scheme and the first-order backward Euler scheme. The convergence criterion is defined as a RMS error. Focusing mainly in the natural convection phenomenon, it is adopted the k-Omega turbulence model, that is represented by the k-equation (Equation (4.19)) and the ω-equation (Equation (4.20)):

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j k) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \beta' \rho k \omega + P_{kb}, \tag{2.15}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j \omega) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + \alpha \frac{\omega}{k} P_k - \beta \rho \omega^2 + P_{\omega b}. \tag{2.16}$$

As described in [60], in addition to the independent variables, the density ρ, and the velocity vector U, are treated as known quantities from the Navier-Stokes method. P_k is the production rate of turbulence and the constants have the following values: $\beta' = 0.09$, $\alpha = 5/9$, $\beta = 0.075$, $\alpha_\omega = 2$ and $\alpha_k = 2$. The effect of gravity is taken into account in the buoyancy term. For the boundary conditions, a non-slip condition is imposed at solid surfaces with a third-type boundary condition.

Space discretization

The CFD mesh is generated by ANSYS Mesh generator, and to assure better independence of the CFD simulation with respect to computational grid, a grid-convergence study is performed adopting the Grid Convergence Index (GCI) method, proposed by [28]. This method provides an approach for uncertainty assessment of grid convergence based on a grid refinement error estimator. As GCI was originally proposed to be applied in uniform grids, an alternative option for adaptive grids was suggested in [29] where a number of mesh nodes could be used to calculate refinement ratio. Based on [29], the equation to calculate GCI is expressed as:

$$GCI = 3|\varepsilon|/(r^p - 1), \tag{2.17}$$

$$\varepsilon = (f_2 - f_1)/f_1, \tag{2.18}$$

$$r = (N_{fine}/N_{coarse})^{1/3}, \tag{2.19}$$

where f_1 is the monitored value at a point with a fine grid; f_2 is the monitored value at the same point with a coarse grid; N_{fine} and N_{coarse} are the grid numbers of fine and coarse meshes, respectively, and p denotes the accuracy order of the numerical scheme.

Chapter 3

Intelligent co-simulation: neural network vs. proper orthogonal decomposition applied to a 2D diffusive problem

Paper published online in the Journal of Building Performance Simulation on 06-feb-2018.
DOI: 10.1080/19401493.2017.1414879
ISSN: 1940-1493 (Print) 1940-1507 (Online) Journal homepage: http://www.tandfonline.com/loi/tbps20
Impact Factor: 2.874, Qualis: A2. Scimago: Q1

3.1 Introduction

Most building energy simulation (BES) tools, developed since the 1970's, are based on one-dimensional (1D) diffusive heat transfer, [3–6, 31–37]. The main reasons are due to the excessive computational cost to perform two- or three-dimensional (2D–3D) calculations through whole-building envelopes. Sometimes, the ratio between the computer run time and the real physical time can be even higher than one for the computers available nowadays.

However, 1D diffusive models may lead to a lack of accuracy to represent complex physical phenomena associated to non-uniform convective and radiative boundary conditions, thermal bridges and complex geometries. As a solution to this problem, complex models representing physical phenomena, generally available in specialized software, as CFD, have been coupled to the BES tools by means of a computational procedure known as *co-simulation*. Basically, BES commands the simulation employing its simplified models for the majority of physical phenomena. However, for a specific physical phenomenon, the complex model available in the specialized software is used, creating exchange of data during the simulation. Figure 3.1a illustrates this procedure. At each time iteration t^n, each model (available in the BES or in the specialized software), exchanges parameters $\boldsymbol{P}$ with the other. Then, during the interval $t \in \left[t^n, t^{n+1} \right]$, each model computes the field of interest (u and v). It is important to note that during this interval the parameters $\boldsymbol{P}$ are considered as constant in time. A more detailed review of co-simulation examples can be seen in some recent works in the frame of the IEA Annex 60 [47] as well as [61] and [9]. For instance, [39] proposed a co-simulation to the interactions between room airflow, heat, ventilation and air conditioning (HVAC) systems and building envelope, within the outlooks of systems management. In [62], a co-simulation is carried out between building simulation and urban-level programs to investigate the energy consumption of two university campuses.

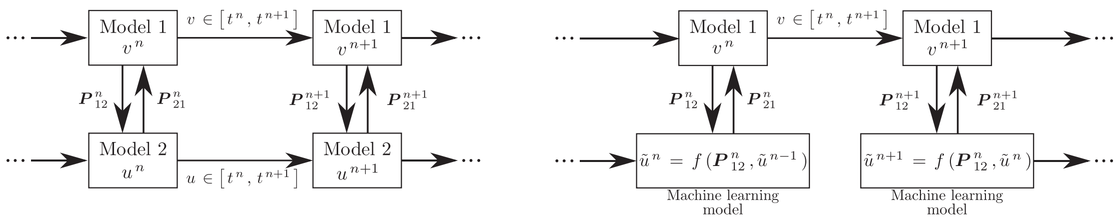
The benefits of the co-simulation in terms of improved accuracy is certainly unquestionable, as can be noted for instance in [63], where a building simulation software is coupled with an in-house model to take into account heat and moisture transfer in soils. However, as discussed in [61], improved accuracy comes with a price: a significant computational cost. For instance, in [64], a co-simulation was performed between a building simulation program and an in-house reduced order model to decrease the computational cost, for simulating 2D heat and moisture transfer through a porous wall. The latter was packaged in `Matlab` executable file. The co-simulation involved two numerical solvers with data exchange through `.txt` files and consequently the reduction of the computational cost was not the one expected with the use of model reduction methods.

As a first attempt to solve this drawback, in [61], an interesting approach, called *intelligent* co-simulation, was proposed to reduce the computational cost of the co-simulation procedure based on the coupling of two building simulation programs to accurately represent some physical phenomena. This new co-simulation scheme consists of designing a new model, called *prediction model*, capable to provide results, as close as possible to the ones provided by the complex model, with a lower run time. The synthesis of the prediction model is based on artificial intelligence, constructed by means of a machine learning procedure, using input and output data of a co-simulation run on a complex model used to simulate the physics. The synthesized prediction model replaces the complex model with the purpose of reducing significantly the computational effort with a low impact on the accuracy. Therefore, in this intelligent co-simulation approach, the model learns from the first results of the co-simulation to disconnect later one of the programs. At this time, as illustrated in Figure 3.1b, the machine learning model replaces the disabled program (here model 1) with a prediction model f, depending on the parameters $\boldsymbol{P}_{12}$. The prediction model computes an approximation of the field

$\tilde{u}^{\,n+1} = f(\boldsymbol{P}_{12}^{\,n}, \tilde{u}^{\,n})$, at a lower computational cost until the simulation is finished.

Nevertheless, as discussed in [61], research is still needed to reduce the training time period. In this way, this paper proposes to investigate the intelligent co-simulation approach, aiming to reduce the cost of the training phase. In this sense, two machine learning models are explored, based on Artificial Neural Network (ANN) and Proper Orthogonal Decomposition (POD) model reduction method. These two machine learning models have been selected mainly for their promising efficiencies reported in literature. In [65], the Navier–Stokes equation is computed using ANN built approximation function. Other examples of ANN applications are also given in [43]. Results have also been recently proposed for the POD approach with examples in literature, particularly in building physics, as for instance [66, 67] (further references are given in Section 3.2.2).

After presenting the general approach of intelligent co-simulation in Section 4.3.3, the advantages of both machine learning models are investigated. First, a simple case study of heat transfer is considered in Section 3.3. It enables to analyze deeper the influence of the parameters and to draw intermediary conclusions. In Section 3.4, a complex case of co-simulation is considered involving two simulation programs, `Domus` and `CFX`, to evaluate the performance of the intelligent co-simulation approach and the gain in terms of accuracy and computational cost.



(a) Classic co-simulation (b) Intelligent co-simulation

Figure 3.1: Schematic description of the intelligent co-simulation.

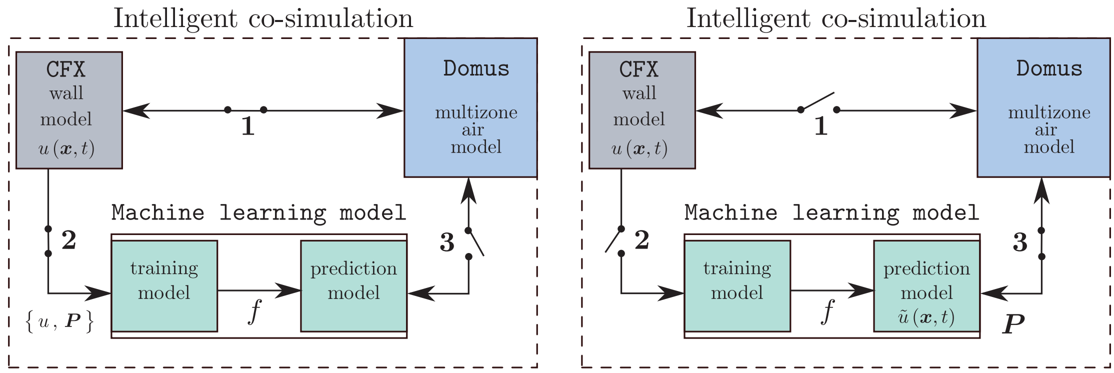
3.2 Intelligent co-simulation approach

3.2.1 General statement

The intelligent co-simulation approach was introduced in [61], considering Domus, as a building performance simulation tool, and CFX as a CFD specialized software, which relies on the construction of a *machine learning model* that can stop the communication between the two programs after a short period of simulation and replace one of them until the end of the simulation. As illustrated in Figure 3.2, the machine learning model is composed of two sub-models: i) the training model and ii) the prediction model. The field of interest computed by `CFX`is denoted by $u(\boldsymbol{x}, t)$.

The training model, $t \in [0, \tau]$**:**
During the first phase, the co-simulation occurs normally between `Domus` and `CFX`, while the training model receives the results $u(\boldsymbol{x}, t)$ and the corresponding parameters $\boldsymbol{P}$ (as boundary conditions) from the two software. During this phase, `CFX` runs continuously exchanging data at each synchronization point. As noticed in Figure 3.2a, during this phase, switches 1 and 2 are ON and switch 3 is OFF, the co-simulation is operating as usual. The training model receives the results from the `CFX` program. The training phase is defined for a fixed period τ. The boundary conditions of this phase are called the *training climate*. Its definition and influence are investigated in next sections.

The prediction model, $t \in [\tau, \Gamma]$**:**
After a long enough training phase, the training model provides to the prediction model an approximation function f. The latter model is used to co-simulate with `Domus` and compute an approximation $\tilde{u}(\boldsymbol{x}, t)$ of the field u as a function of the parameter $\boldsymbol{P}$. Thus, the `CFX` program is no longer used. As noticed in Figure 3.2b, during this second phase switches 1 and 2 are OFF, while switch 3 is ON. The prediction model is implemented inside `Domus` (in `C++` language) in order to avoid the increase of computational time. Thus, during a whole simulation, the intelligent co-simulation is divided into two steps, commanded by a coupling/uncoupling algorithm.



(a) Training phase

(b) Prediction phase

Figure 3.2: Schematic description of an example of intelligent co-simulation.

It should be remarked that in the case of the classic co-simulation, `CFX` runs continuously during the whole simulation, exchanging data with `Domus` at each data synchronization point. Moreover, it is important to note that in this study the so-called *quasi-dynamic coupling* strategy was used between `CFX` and `Domus`. This strategy was selected for practical reasons, because of the high computational time demanded by the normal co-simulation. Indeed, as it will be illustrated in Section 3.4, the co-simulation takes around $14\,\mathrm{h}$ to compute the solution for one year. It should be mentioned that a *fully-dynamic coupling* strategy could be used probably improving accuracy of the results but at the price of a higher computational effort.

3.2.2 The Proper Orthogonal Decomposition as a machine learning model

To perform the intelligent co-simulation, an interesting approach is the Proper Orthogonal Decomposition (POD). Also referenced as Karhumen-Loève decomposition or Principal Component Analysis, it was proposed by KARHUNEN in 1946 [68] and by LOÈVE in 1955 [69]. This method extracts the main information from a large set of data by projecting them into a smaller subspace. It was developed and applied in several domains such as image processing, oceanography or weather forecasting and it is one of the most widely used method to reduce the order of dynamic models. In the framework of building applications, the method was firstly applied on computational fluid dynamics problems of turbulent flows by Lumley [70]. Then, other studies were carried out on similar cases [71, 72], on the control of unstationary flows [73, 74], on the particle dispersion in ventilated cavities [66] or on the temperature distribution [75]. The POD method was applied on heat diffusion [76, 77], on convection [78] and on heat and moisture [67] transfer problems. As mentioned in [79, 80], an important negative drawback of the POD is the inherent extra-computational cost. Even if the order of reduction is lower than other approaches such as proper generalized decomposition

(PGD) or Spectral [80], a previous computation (reported in literature as *learning* or *warming* steps) of the solution of the problem is required to build a reduced order model. This preliminary computation is a non-negligible restriction when considering only one physical model. Here, models coupled through co-simulation are under investigation and the POD drawback is turned into an efficient convenience. During the training phase, the results from co-simulation between `Domus` and `CFX` are used to compute as a prediction model.

The POD approach searches a set of basis functions Φ, for which the projection of $u(\boldsymbol{x}, t)$ best approximates the mean field [66]. It is an *a posteriori* model reduction method. The basis Φ is built during the training phase. During the prediction phase, the solution of the problem is approximated by:

$$\tilde{u}(\boldsymbol{x}, t) = \sum_{i=1}^{N} \Phi_i(x, y)\zeta_i(t), \tag{3.1}$$

where ζ is the new unknown of the POD reduced order model, used during the co-simulation with `Domus`.

The training model

Using the `CFX` program, the solution $u(\boldsymbol{x}, t)$ is known for $N_x \times N_t$ discrete points in the domains $\Omega_x \times [0, \tau]$. Using these results, the correlation matrix γ of the field is computed, according to the snapshots approach:

$$\gamma_{ij} = \left\langle u(\boldsymbol{x}, t_i), u(\boldsymbol{x}, t_j) \right\rangle .$$

Then, the *a posteriori* basis Φ is computed by solving the spectral problem:

$$\gamma\Phi = \lambda\Phi. \tag{3.2}$$

Once the spectral problem given in Eq. (3.2) is solved, a truncation is performed in the POD basis Φ, choosing N of the M eigenvectors. To obtain an optimal decomposition, in the energetic sense [81], functions $f(N)$ corresponding to the energy of the mode N, are defined:

$$e(N) = \sum_{k=1}^{N} \lambda_k \left(\sum_{i=1}^{M} \lambda_i \right)^{-1} .$$

In this work, the order of the reduction N is chosen as $e(N) \geqslant 0.999$. In practice N is the order of ten and M of thousands for 2D problems. According to the description of the intelligent co-simulation and Figure 3.2, the parameter $\boldsymbol{P}$ corresponds to the discrete *snapshot* matrix $u(\boldsymbol{x}, t)$ and the approximation function f is the basis Φ.

The prediction model

Using an explicit time stepping scheme and the finite-difference approach, the solution $u(\boldsymbol{x}, t)$ is obtained by solving an algebraic system of the form:

$$\mathbf{K}\mathbf{u}^{n+1} = \mathbf{M}\mathbf{u}^{n} + \mathbf{Q},$$

where matrices $\mathbf{K}$ and $\mathbf{M}$ result from the discretisation and $\mathbf{Q}$ from the boundary conditions of the problem. It is important to note that matrices $\mathbf{K}$ and $\mathbf{M}$ may change during the time stepping, particularly if the physical problem is nonlinear or if the numerical method uses adaptive in time and/or in space schemes. In this study, involving linear heat transfer and constant discretisation for time and space domains, the matrices remain unchanged over the simulation.

Once the basis Φ is computed during the training phase, the POD prediction model computes the results of the 2-dimensional heat transfer using approximation of Eq. (3.1). The temporal coefficients ζ are calculated by:

$$\Phi^T \mathbf{K} \Phi \zeta^{n+1} = \Phi^T \mathbf{M} \Phi \zeta^{n} + \Phi^T \mathbf{Q}.$$

It can be noticed the effect of model order reduction as the new unknown ζ of the problem to be solved has the dimension $N << M$.

3.2.3 The Recurrent Neural Network as a machine learning model

Artificial neural networks were created with the purpose of being similar to the learning process of a human brain. The main objective is the generalization of complex rules and mathematical routines, learning the key information patterns within a multidimensional information domain. As described in [43], neural network learning could be used to represent universal function approximation. Other parameters like the number of layers, the number of training samples, the length of learning period, the choice of neuron activation functions, and the training algorithm could also affect the training accuracy. There are two major types of neural networks: (i) feed-forward and (ii) recurrent. In the feed-forward network, the input data goes through the network in a one-way direction to generate an output result. A recurrent neural network (RNN) has one, or more, cyclic path of synaptic connections and is used in this study. The RNN was selected in this work thanks to its internal memories, that enable to approximate the current solution in function of the previous results. Since we are dealing with time dependent problems, the RNN is more appropriate. In addition, it should be noted that there are several computational software libraries available for simple implementation in [42] with simple examples in [44]. Interested readers are invited to consult [61] for a better review on the use of artificial neural networks, particularly for building physics applications.

The training model

The RNN model used in this work is a supervised single hidden neuron layer with a context layer that provides feedback from the previous outputs. This RNN structure and the feedforward method was implemented in `C++` programming language, however the training is performed by the `Matlab` Neural Network toolbox, using the Bayesian regularization (BR) training algorithm. The BR updates the weight and bias values according to LEVENBERG–MARQUARDT optimization. Furthermore it determines the correct combination of minimized squared errors and weights, so delivering a network with better quality of generalization [51, 52].

An algorithm was developed to estimate the best number of hidden neurons for each input set of data. With the data obtained from the CFD co-simulation, results from the last day are often used to test the accuracy of the RNN prediction model. Thus, the remaining data is used by the training model. The algorithm executes the training procedure several times, testing at each time the performance under the test dataset. At the end, the RNN with the lowest error ε_{inf} on the testing data is selected for use.

The prediction model

After the training phase has been concluded in the `Matlab` environment, the calculated weights are transmitted to the selected RNN implemented in `C++`. The RNN prediction model is ready to perform the computation during the prediction phase. Two activation functions have been selected, Hyperbolic tangent sigmoid for the output of the hidden Layer and pure Linear function for the output layer [61]. Within the Recurrent Neural Network, the solution of the problem is approximated by:

$$v_k(t+dt) = f_1\Big(b_{1,k} + \sum_{i=1}^{M} W_{1,k,i}\, X_i + b_{c,k} + \sum_{i=1}^{N} W_{c,k,i}\, v(t)\Big),$$
$$\tilde{u}(t+dt) = f_2\Big(b_2 + \sum_{k=1}^{N} W_{2,k}\, v_k(t+dt)\Big),$$

where $\tilde{u}$ is the approximated model output, N is the number of neurons in the hidden and context layers, M is the number of inputs, X_i is the input vector, f_1 is the transfer function in the hidden layer, $W_{1,k,i}$ are the weights in the hidden layer, $W_{c,k,i}$ are the weights in the context layer, f_2 is the output layer transfer function, $W_{2,k}$ are the weights in the output layer and b^1_{Nx1}, b^2 and b^c_{Nx1} are the bias for the hidden, output and context layers, respectively. As initial condition, $v(0)$ is equal to the input vector.

3.2.4 Comparing the numerical solutions

To compare and validate the numerical solution, the error between the solution $u(x,t)$ and a reference solution u^{ref}, is computed as a function of t by the following formulation:

$$\varepsilon_2(t) \overset{\text{def}}{:=} \sqrt{\frac{1}{N_x} \sum_{j=1}^{N_x} \Big(u_j^{\text{num}}(x,t) - u_j^{\text{ref}}(x,t)\Big)^2},$$

where N_x is the number of spatial discretised points. The global uniform error ε_∞ is given by the maximum value of $\varepsilon_2(x)$:

$$\varepsilon_\infty \overset{\text{def}}{:=} \sup_{t \in [0,\Gamma]} \varepsilon_2(t).$$

As detailed in next Sections, the reference solution $u^{\text{ref}}(x,t)$ is given by the results of the classical co-simulation.

3.3 Numerical investigation of a simple case study

As both POD and RNN machine learning models involve many parameters influencing the performance of the intelligent co-simulation, a primary simple case is considered. The physical phenomenon involves a building envelope wall, where 1-dimensional heat transfer phenomenon occurs.

3.3.1 Definition of the problem

The physical problem for the wall model can be formulated as:

$$\rho c \frac{\partial T}{\partial t} - \lambda \frac{\partial^2 T}{\partial x^2} = 0 \qquad \forall x \in [0, L], t \in [0, \Gamma]. \tag{3.3}$$

The wall can be multilayered and a perfect contact is assumed at the interface between two materials. The following convention is adopted: $x = 0$ corresponds to the surface in contact with the inside room and, $x = L$, corresponds to the outside surface. The boundary conditions are expressed as:

$$\lambda \frac{\partial T}{\partial x} = h_e \big(T - T_e \big) - q_e \,, \qquad x = 0 \,, t > 0 \,,$$
$$-\lambda \frac{\partial T}{\partial x} = h_i \big(T - T_i \big) \,, \qquad x = L \,, t > 0 \,,$$

where h_e and h_i are the convective heat transfer coefficients, T_e and T_i are the outside and inside temperatures, respectively, and q_e the outside heat flux due to long-wave radiations. Their variations are defined as:

$$T_e = T_{e,0} + \delta_{e,1} \sin\big(2\pi \omega_{e,1}\, t\big) + \delta_{e,2} \sin\big(2\pi \omega_{e,2}\, t\big) \,, \tag{3.4a}$$
$$T_i = T_{i,0} + \delta_{i,1} \sin\big(2\pi \omega_{i,1}\, t\big) \,, \tag{3.4b}$$
$$q_e = q_{e,0} \sin\big(2\pi \omega_{qe,1}\, t\big)^{20} \,. \tag{3.4c}$$

An uniform initial distribution temperature is considered:

$$T = T_0 \,, \qquad \forall x \in \big[0, L \big] \,, t = 0 \,.$$

The issue is to compute the surface temperature $T_s = T(x = L, t)$ for $t \in \big[0, \Gamma\big]$ by solving the problem defined above using POD and RNN machine learning models. To accomplish this, the computation is divided into two steps. During the training step, the problem is solved for $t \in \big[0, \tau\big]$ using the improved explicit DUFORT–FRANKEL scheme. Interested readers may consult [82] for a description of the numerical scheme. For the sake of simplicity and without loosing the generality of the analysis, the numerical study has been carried out using the `Matlab` environment. Results from this simulation are used by the POD and RNN training models to build a prediction model. Then, during the prediction phase, the problem defined by Eq. (3.3) is computed using the POD and RNN prediction models. Results from both approaches are compared to the reference solution computed with the DUFORT–FRANKEL scheme.

It should be mentioned that the whole problem is written using a dimensionless formulation when performing the computations, enabling to determine important scaling parameters such as the (BIOT and FOURIER numbers). Henceforth, solving one dimensionless problem is equivalent to solve a whole class of dimensional problems sharing the same scaling parameters. Then, dimensionless equations can be used to estimate the relative magnitude of various terms, and thus, eventually to simplify the problem using asymptotic methods [83]. Finally, the floating point arithmetics is designed such as the rounding errors that become minimal if one manipulates the numbers of the same magnitude [84]. Moreover, the floating point numbers have the highest density within the interval $(0, 1)$ which decays exponentially when we move further away from zero. So, it is always better to manipulate numerically the quantities at the order of $\mathcal{O}(1)$ to avoid severe round-off errors and to likely improve the conditioning of the problem in hands.

3.3.2 Numerical application

The case study considers a wall length $L = 0.1\ \mathrm{cm}$. The initial temperature in the zone and the wall is $T_i = 20\ ^\circ\mathrm{C}$. The convective heat transfer coefficients are set to $h_i = 8.7\ \mathrm{W/(m^2.K)}$ and $h_e = 23.3\ \mathrm{W/(m^2.K)}$. The material properties of the wall correspond to those of concrete: $\lambda = 1.75\ \mathrm{W/(m.K)}$, $c = 1000\ \mathrm{J/(kg.K)}$ and $\rho = 2200\ \mathrm{kg/m^3}$ and those of brick: $\lambda = 0.745\ \mathrm{W/(m.K)}$,

$c = 920\ \mathsf{J/(kg.K)}$ and $\rho = 1900\ \mathsf{kg/m^3}$. The following numerical values are considered for the outside and inside boundary conditions:

$$\begin{aligned}
&T_{e,0} = 20\ ^\circ\mathsf{C}, \quad \delta_{e,1} = -4.4\ ^\circ\mathsf{C}, \quad \omega_{e,1} = \frac{1}{8}\ \mathsf{h}^{-1} \quad \delta_{e,2} = -11.7\ ^\circ\mathsf{C}, \quad \omega_{e,2} = \frac{1}{50}\ \mathsf{h}^{-1}, \\
&T_{i,0} = 20\ ^\circ\mathsf{C}, \quad \delta_{i,1} = 1.46\ ^\circ\mathsf{C}, \quad \omega_{i,1} = \frac{1}{100}\ \mathsf{h}^{-1} \quad q_{e,0} = 500\ \mathsf{W/m^2}, \quad \omega_{qe,1} = \frac{1}{10}\ \mathsf{h}^{-1}.
\end{aligned}$$

The total simulation horizon is $\Gamma = 100\ \mathsf{days}$.

Training phase
The problem is first solved for $\tau = 34\ \mathsf{days}$ using the DUFORT–FRANKEL explicit scheme. The training climate corresponds to the boundary conditions from Eq. (3.4) for $t \in [0, \tau]$, as illustrated in Figure 3.3a. Results are used by the training models to compute a prediction model. For the POD training model, a reduced model of order $N = 3$ is built. With the RNN training model, a network with number of neurons $M = 3$ is defined.

Prediction phase
Using the prediction models, the problem is solved for $t \in [0, \Gamma]$ using the boundary conditions shown in Figure 3.3b. The surface temperature variations are given in Figure 3.3c. A very good agreement can be observed between the POD and RNN solutions, compared to the reference one computed using the DUFORT–FRANKEL approach. The error ε is of the order $\mathcal{O}(10^{-4})$ as shown in Figure 3.3d. The error of the RNN prediction model is lower for $t \in [0, \tau]$, as this period corresponds to the same simulation carried out during the training phase.

A parametric study on the duration of the training phase has been carried out. For each value of parameter τ, the $\mathcal{L}_2$ error has been computed with the reference solution, as shown in Figure 3.4. The error of the POD prediction model is relatively constant at $\mathcal{O}(10^{-3})$, which is consistent with the energy criterion defined in Section 3.2.2. For this model, a short duration of the training phase $\tau = 2\ \mathsf{days}$ is sufficient to provide an accurate prediction model. It is important to note that a lower error could have been obtained by defining a higher energy criterion and therefore building a reduced model with a higher order. The error of the RNN prediction model becomes satisfactory for a duration of the training phase $\tau = 32\ \mathsf{days}$. After this value, the error is lower than the one for the POD prediction model. Even if the error remains completely acceptable, it can be noted that the error has important variations due to the random initial guess of the RNN weight basis used when building the prediction model.

The POD and RNN prediction models are approximation functions built using the output T_s. If we look carefully at the training climate for $t \in [0, 32]\ \mathsf{days}$, it corresponds to a period for which the training models receive almost the whole range of values of the T_s field. After the value $\tau = 32\ \mathsf{days}$, the error decreases and the prediction models become more accurate. During this numerical study, it has been noted that the order of the POD prediction model and the number of nodes of the RNN prediction model remained stable at $N = M = 3$.

Two intermediary results may be highlighted from this simple case study. First, both approaches enable to build accurate prediction models of the T_s field, based on a training phase shorter than the simulation horizon of the considered problem. The POD prediction model requires a shorter training phase than the RNN. Second, the accuracy of the machine learning models increases if the training models receives a wider range of values of the field T_s to build the prediction models.

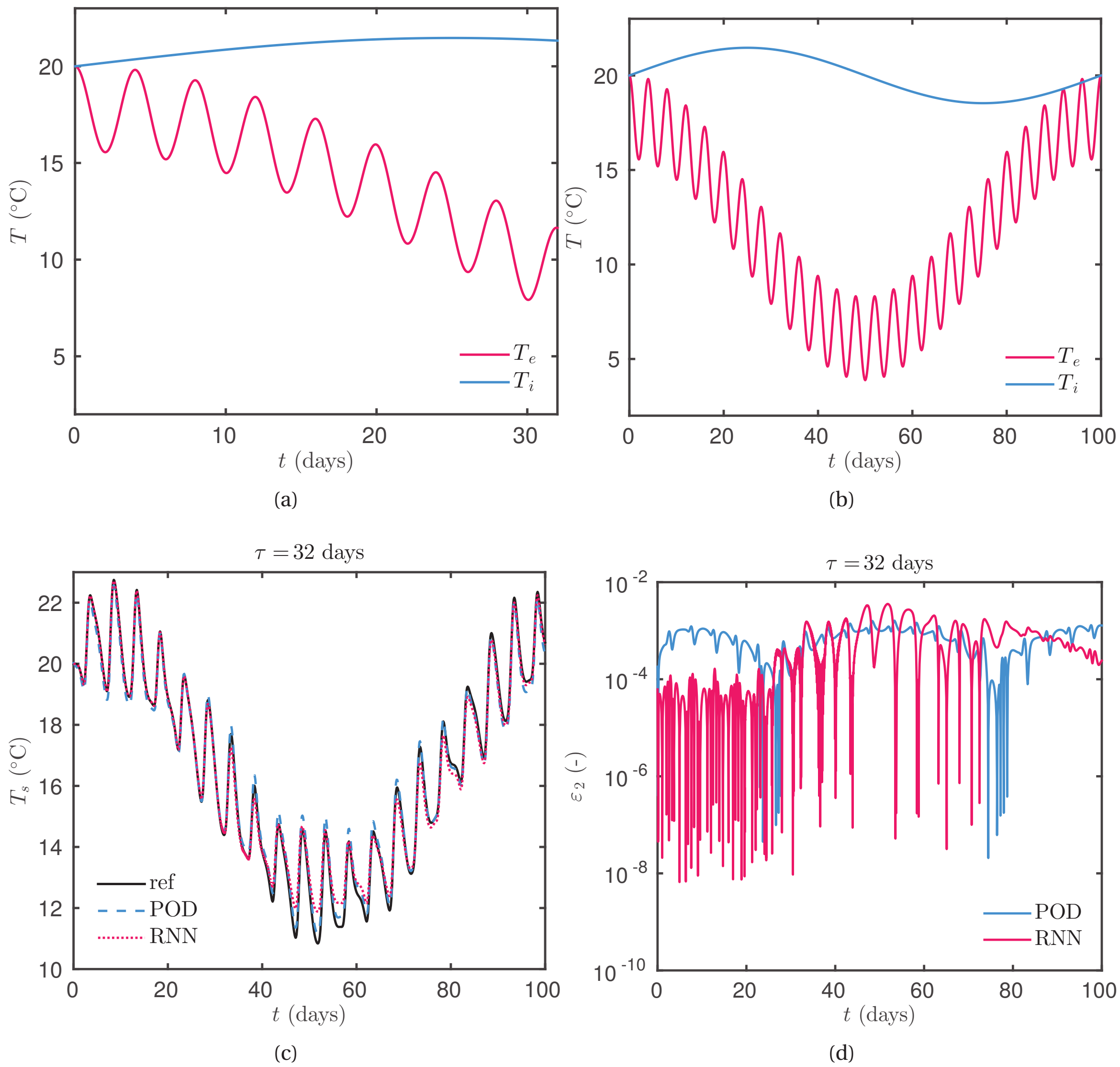


Figure 3.3: Temperature boundary conditions of the training phase (a), of the prediction phase (b) and evolution of the temperature surface for the RNN and POD prediction models (c) with their respective errors (d).

3.4 Intelligent co-simulation for whole-building energy simulation with 2-dimensional diffusive problem

The previous case study considered a single wall with 1-dimensional heat transfer. The purpose was to investigate and understand, for a simple case, the influence of the numerical parameters of both POD and RNN machine learning models. It has been highlighted that both approaches enable to compute an accurate function approximation of the interested field based on a training phase shorter than the simulation horizon of the considered problem. In this Section, the issue is to focus on a more complex problem of co-simulation involving the `Domus` and `CFX` programs, to solve a purely 2D diffusive problem as illustrated in Figure 3.5.

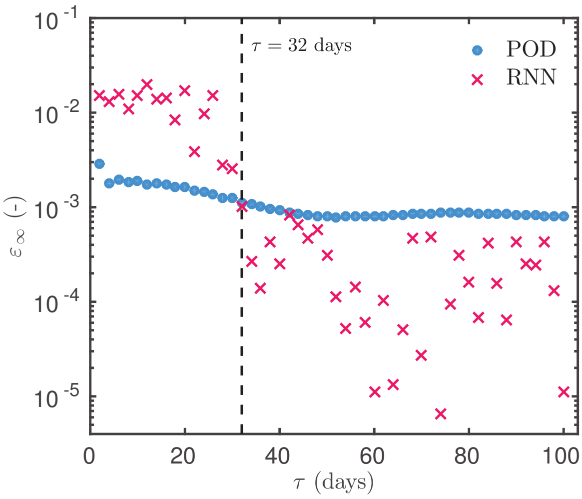


Figure 3.4: Error ε as a function of the number of the training days.

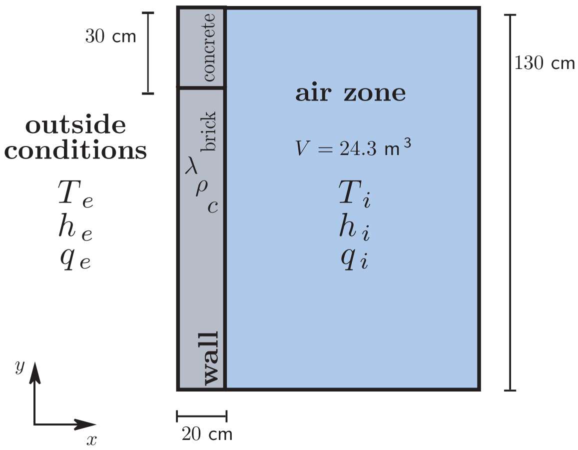


Figure 3.5: Schematic description of the case study.

3.4.1 The whole building energy program `Domus`

`Domus` is used as the whole-building energy model, as it has already been prepared to co-simulate with CFX. It enables to compute the air temperature in the zone using the lumped multizone model. `Domus` has one-dimensional models of heat and moisture transfer through porous materials, that have been disabled for this work. The whole program is written in `C++` and is available at `http://www.domus.pucpr.br/`. Interested readers may refer to [55] for further documentation.

3.4.2 The 2D Wall Model

The wall model computes the evolution of temperature considering 2-dimensional unsteady heat transfer and can be described by:

$$\rho\, c\, \frac{\partial T}{\partial t} - \lambda \frac{\partial^2\, T}{\partial x^2} - \lambda \frac{\partial^2\, T}{\partial y^2} = 0, \qquad \forall\, (x, y) \in \Omega_x \times \Omega_y, t > 0, \tag{3.5}$$

The spatial domains are defined as $\Omega_x = \left[0, L_x\right]$ and $\Omega_y = \left[0, L_y\right]$. The wall can be multilayered and a perfect contact is assumed at the interface between materials. The inside and outside temperatures and heat fluxes influence the facade of the wall through the boundaries conditions:

$$\lambda \frac{\partial T}{\partial x} = h_e \left(T - T_e \right) + q_e, \qquad \forall\, y \in \Omega_y,\, x = 0,\, t > 0,$$
$$-\lambda \frac{\partial T}{\partial x} = h_i \left(T - T_i \right) + q_i, \qquad \forall\, y \in \Omega_y,\, x = L_x,\, t > 0.$$

The temperatures T_e and T_i, the convective heat transfer coefficients h_e and h_i, and the heat fluxes q_e and q_i stands for the outside and inside boundary conditions of the walls. The other boundaries are considered as adiabatic: $\lambda \frac{\partial T}{\partial y} = 0, \quad \forall\, x \in \Omega_x,\, y = \left\{0, L_y\right\},\, t > 0$. The initial temperature in the wall is considered as uniform: $T = T_0, \quad \forall\, (x, y) \in \Omega_x \times \Omega_y,\, t = 0$. Problem of Eq. (3.5) is solved using CFX program, based on the finite-element method and a sufficiently converged grid using a second-order backward Euler scheme.

3.4.3 The co-simulation between CFX and Domus

The co-simulation is carried out using software CFX for the wall 2-D heat transfer and Domus for the inside air zone. The programs are coupled using a direct approach written in C++, *i.e.*, the convergence of the whole model is checked at each time step, being Domus the master of the simulation. At the initialization stage, Domus defines the material properties of the wall (λ, ρ, c), the outside and inside convective heat transfer coefficients (h_i, h_e) and the coupling time step between both programs. At each time step, Domus provides the outside and inside boundary conditions (T_i, T_e, q_i, q_e) to CFX. After computing the field within the wall, CFX returns to Domus the internal surface averaged temperature $\bar{T}_{\mathrm{s}}$, defined as:

$$\bar{T}_{\mathrm{s}} = \frac{1}{L_y} \int_0^{L_y} T\left(L_x, y\right) \mathrm{d}y.$$

Interested readers are invited to consult [61] for further information on the coupling between Domus and CFX. The POD and RNN machine learning model are integrated in Domus in C++ code. The algorithm of the first one is generated using Matlab platform. For the RNN approach, the training model is performed using Matlab platform whereas the prediction model algorithm has been written by the authors directly in Domus.

3.4.4 Numerical results and discussions

Description of the case

The issue is to use the intelligent co-simulation approach to perform a whole-building energy simulation for the case study illustrated in Figure 3.5. For the numerical application, the material properties of the wall are: $\lambda = 1.75$ W/m/k, $c = 1000$ J/kg/K, $\rho = 2200$ kg/m^3 for the concrete and $\lambda = 0.75$ W/m/k, $c = 920$ J/kg/K, $\rho = 1900$ kg/m^3 for the brick. The outside climate is the one for Curitiba, Brazil (Latitude: 25°S, Longitude: 49°W) from the Domus database. The inside condition is set as a sinusoidal variation given in Figure 3.6. The inside and outside convective heat transfer coefficients are $h_i = 7\ \mathrm{W/m^2/K}$ and $h_e = 25\ \mathrm{W/m^2/K}$, respectively. The time step is set to $1\mathrm{h}$ and the total horizon time simulation is $1\ \mathrm{year}$.

Figure 3.10a shows the reference solution, which has been obtained by simulating the case with a normal co-simulation between Domus and CFX. The last week results are zoomed in and shown in Figure 3.10b. The spatial average temperature at the inside surface is compared to the one obtained by performing a simulation with only Domus, which considers only 1-dimensional transfer through

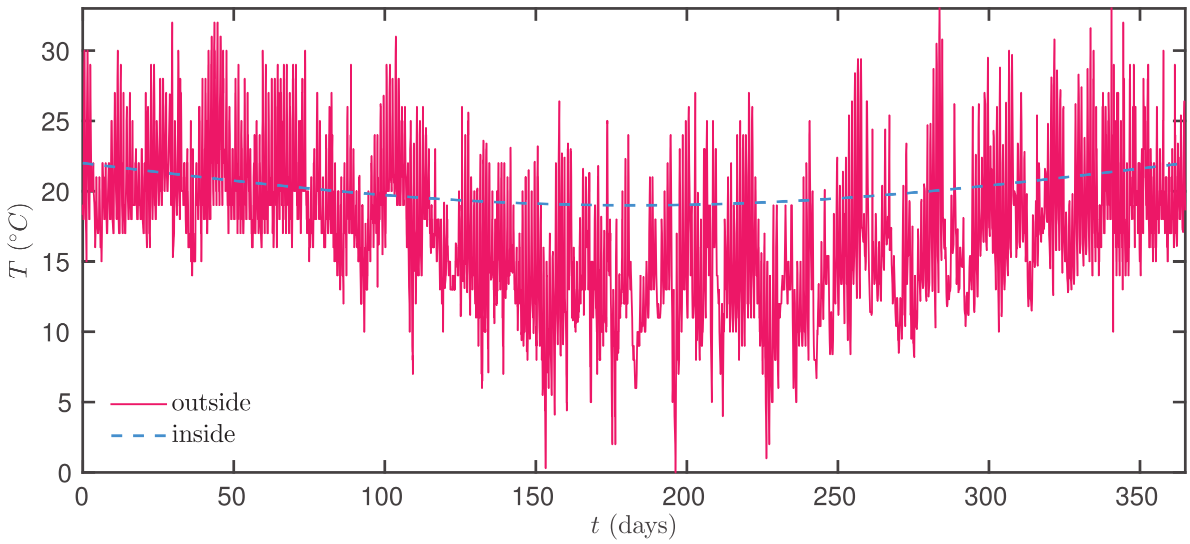


Figure 3.6: Inside and outside boundary conditions.

the wall. The error between the 1D simulation and the 2D co-simulation is given in Figure 3.11. These results highlight the importance of considering 2-dimensional transfer in the wall and justify the use of co-simulation between the two programs. Nevertheless, as reported in Table 3.1, the co-simulation requires more than 14h using Intel i7 CPU and 8GB of RAM to compute the solution. About 80% of this time is devoted to CFX and 20% only for the communication between both programs. The 1D simulation using only Domus took less than 2 min. These results state clearly the problem of co-simulation, which enables to predict with a better accuracy the physical phenomenon, but at the price of a huge computational cost. Analyses of the new results show the benefits of the intelligent co-simulation approach to decrease the computational cost without loosing the accuracy of the results.

Training phase

The space discretisation corresponds to $\Delta x = 4$ mm and $\Delta y = 10$ mm. Thus, for the CFX model, at each time step an algebraic system of 6681 equations has to be solved. The definition of the training climate is an important point. The spectrum analysis of the equivalent outside temperature is illustrated in Figure 3.7. The 24h period is responsible for 10% of the total energy spectrum, while the periods for 1 year and 12h account for less than 1.5%. According to this analysis, the training climate is defined on daily sequences of outside temperature and heat flux. The training phase is first set to $\tau = 15$ days. To provide a robust training, it includes the hottest and coldest days of the year, calculated with a daily average. The sequence is built by including the days before and after those extreme days. All in all, the training climate has eight hot days and seven cold days (08^{th}-16^{th} of December and 09^{th}-16^{th} of June), as shown in Figure 3.8. For the city of Curitiba - Southern Hemisphere, the daily variation appears clearly for the cold days. There are important variations for the heat flux between the cold and hot days. The hottest and coldest days of the year define the limits of the interval of temperature variations within the wall. By defining the training climate as a function of those days, it enables to produce data (here the average surface temperature) for the upper and lower range of the input data for the prediction models. Therefore, it ensures a more robust prediction model when computing new data from input data within that range.

As mentioned before, during the training phase, the POD and RNN training models collect the results of the field of temperature within the wall, obtained by the `CFX` program. Figures 3.9c-3.9f provide the 2D temperature fields at specific time instants of the training climate. These instants are also reported in Figures 3.9a and 3.9b, showing the variation of the outside temperature and heat flux. The inner surface ($x = 0.2$ m) is warmer as the inside zone temperature arises 20 °C. At $t = 13$ h, there is an important gradient of temperature as the outside conditions reach almost their higher magnitude. The different thermal behavior between the concrete and the brick is highlighted. Similarly, at $t = 16$ h, the temperature gradient has a high magnitude as the snapshot is taken after the peaks of temperature and heat flux. At the end of the day, the temperature is almost homogeneous and the wall starts cooling at $t = 20$ h as the outside heat flux equals to zero, and the temperature decreases. The different thermal behavior between the concrete and the brick is highlighted. The concrete has a higher diffusivity implying that the heat diffusion through the wall is faster.

At the end of the training phase, to build the prediction model, the POD training models use the wall temperature *snapshot* matrix composed of of 6681×360 elements. Using the energetic criterion described in Section 3.2.2, the prediction model of order $N = 8$ is built. The RNN prediction models are built with 5 nodes.

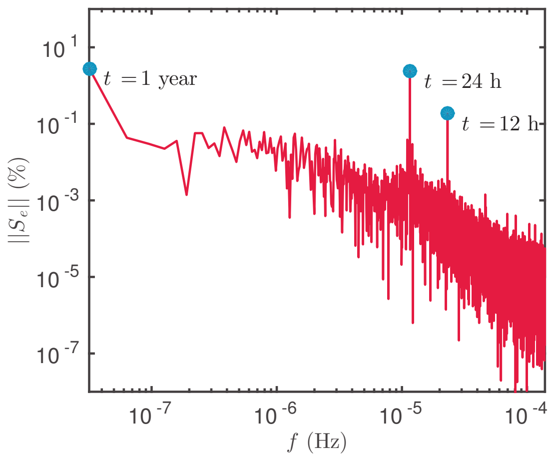


Figure 3.7: Power spectrum of the yearly outside equivalent temperature T_e^{eq} of Curitiba.

Prediction phase

During the prediction phase, the co-simulation between `Domus` and `CFX` is interrupted. The co-simulation is only operated between the RNN or POD prediction models. Figure 3.10c shows the variation of the spatial average temperature for the last week of the simulation. A very fine agreement can be noticed for both POD and RNN prediction models. The time evolution of the error is also provided in Figure 3.11. The error of the POD prediction model remains lower than 10^{-2}, which is consistent with the energy criterion considered to generate the POD prediction model. The error of the RNN prediction model slightly increases during the winter period ($t \in [120, 250]$ days). However, the error remains totally acceptable compared to the 1D simulation approach.

As the training phase is of capital importance in the accuracy of the prediction models, a parametric study has been carried out on the duration τ of the training phase. Figure 3.12a shows the variation of the error as a function of τ. It can be seen that the accuracy of the POD prediction model does not vary much. The error remains at the order $\mathcal{O}(10^{-2})$. A training phase of $\tau = 2$ days would be

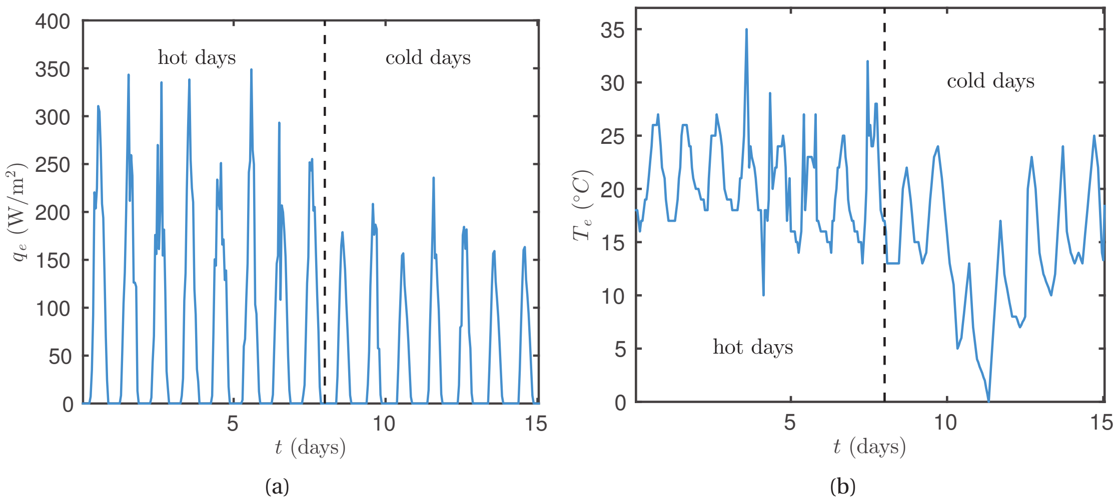


Figure 3.8: Evolution of the outside incident radiation heat flux (a) and temperature (b) of the training climate.

enough to provide an accurate simulation. On the contrary, the error of the RNN prediction model decreases with the duration of the training phase. A training phase of minimum $\tau = 8$ days is necessary to have an accurate RNN prediction model. The order of the POD prediction model varies between $N \in [7, 11]$. The number of nodes of accurate RNN prediction model varies between $M \in [5, 7]$. During this parametric study, it has been noted that the computational run time of the RNN prediction model is multiplied by almost two, when passing from a node number $M = 6$ to $M = 7$. Therefore, the maximum number of nodes has been set to $M = 7$.

According to these results, the computer run time has been evaluated for each approach and reported in Table 3.1. As mentioned before, the CPU time using the classical co-simulation approach requires 14 h to compute the solution, which is significant. Using the intelligent co-simulation, the total CPU time decreases to 88 min and to 136 min, for the POD and RNN based approaches, respectively. The main advantage of the proposed strategy is to reduce exponentially the CPU time of the co-simulation. The reduction is mainly due to two aspects. First, at each time step, the spatial averaged temperature distribution is computed using a fast (POD or RNN) prediction model, instead of using `CFX` to compute the temperature field in 6681 spatial points. Furthermore, the co-simulation between `CFX` and `Domus` programs is cut. As the prediction model is included into `Domus`, no exchange data is required after the training phase. According to Table 3.1, almost three hours of the classical co-simulation are saved.

To reach a satisfying accuracy, the RNN intelligent co-simulation requires a longer training phase than the POD approach ($\tau = 2$ days against $\tau = 8$ days). Thus, as noticed in Table 3.1, the training phase of the RNN intelligent co-simulation is almost four times longer. Nevertheless, an interesting point is that the RNN prediction model is much faster than the POD ones (2 min against 54 min). Two reasons may explain this significant difference. First, the POD prediction model computes the field in the whole spatial domain of the wall, whereas the RNN one only computes the spatial averaged surface temperature $\bar{T}_s$. Moreover, at each time step, the POD algorithm must perform the operation $\mathbf{\Phi} \cdot \zeta$, according to Eq. (3.1), to recompose the physical field in the wall (and then compute the spatial average surface temperature). This computation implies a full big matrix Φ (6681×6681). Then, the implementation of the POD in `Domus` program has been done by generating a `C++` code using `Matlab` coder whereas our own RNN prediction model algorithm has been written. Some improvement of the POD algorithm may be still done to decrease the total CPU time of this approach.

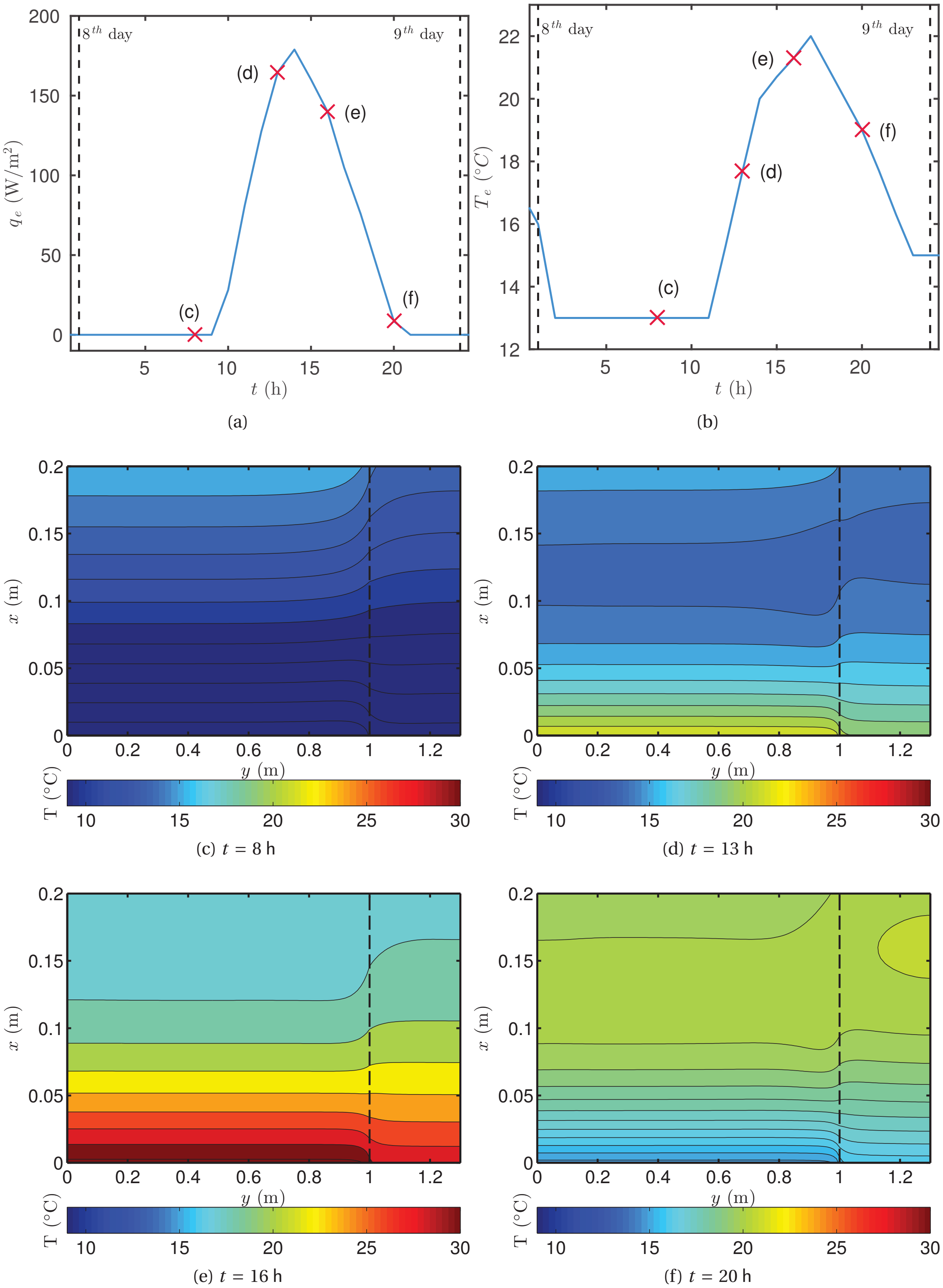
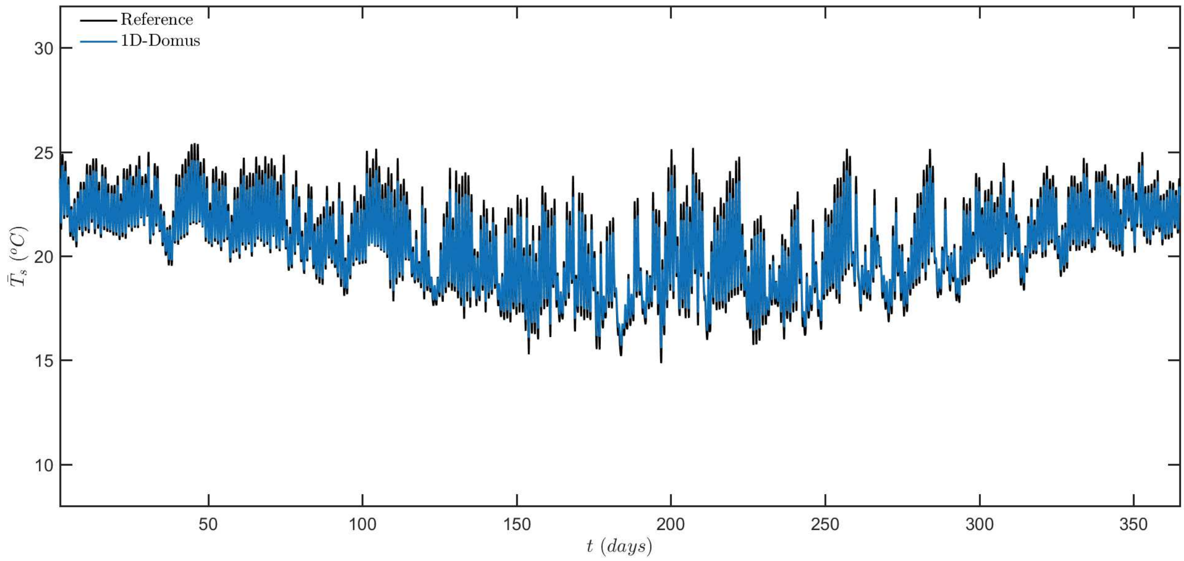
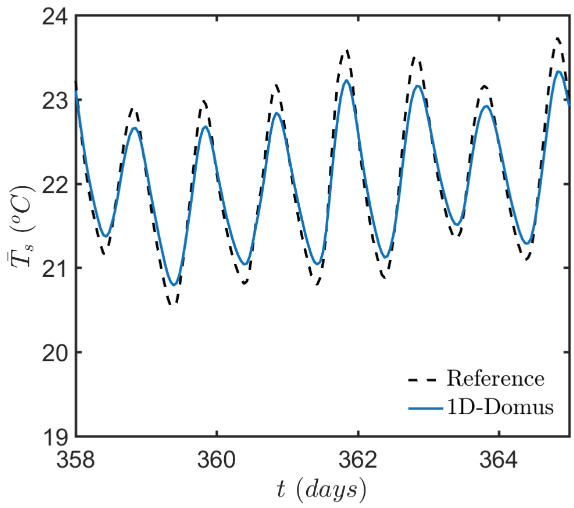


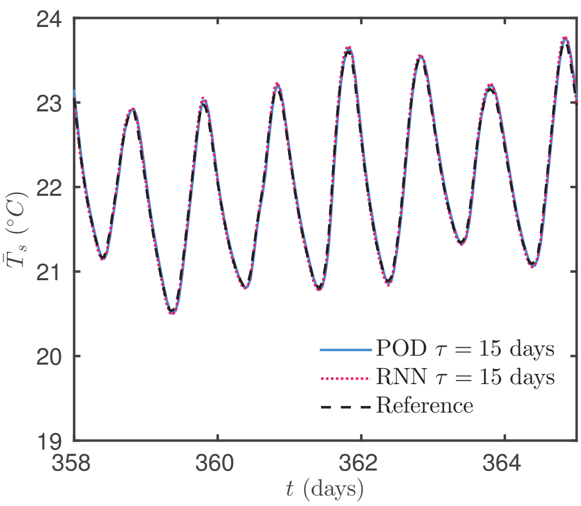
Figure 3.9: Outside heat flux (a) and temperature (b) evolution during the 8^{th} day with the temperature fields within the wall computed with the co-simulation between `CFX` and `Domus`, during the training phase (c–f).



(a)



(b)



(c)

Figure 3.10: Evolution of the surface temperature during the whole year (a) and during the last week (b-c).

Table 3.1: Computational time of the co-simulation.

Case	Total time			CFX			Domus			Data exchange		
	(h)	(min)	(%)	(h)	(min)	(%)	(h)	(min)	(%)	(h)	(min)	(%)
1-D simulation with Domus	0.03	2	< 1	–	–	–	0.03	2	< 1	–	–	–
Classical co-simulation	14.8	888	100	12	720	80	0.001	0.06	< 1	2.8	168	20
POD-based intelligent co-simulation (τ = 2 days)	1.46	88	9.9	0.56	34	3.8	0.90	54	6	0.02	1.2	< 1
RNN-based intelligent co-simulation (τ = 8 days)	2.25	135	15.2	2.21	133	14.9	0.03	2	< 1	0.08	4.8	< 1

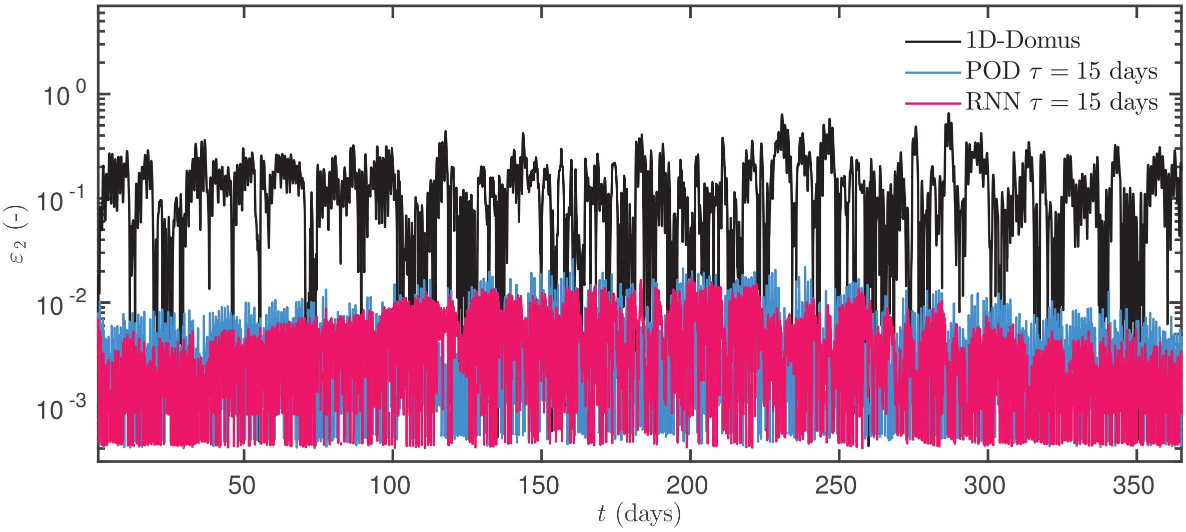


Figure 3.11: Time evolution of the error ε with the reference solution.

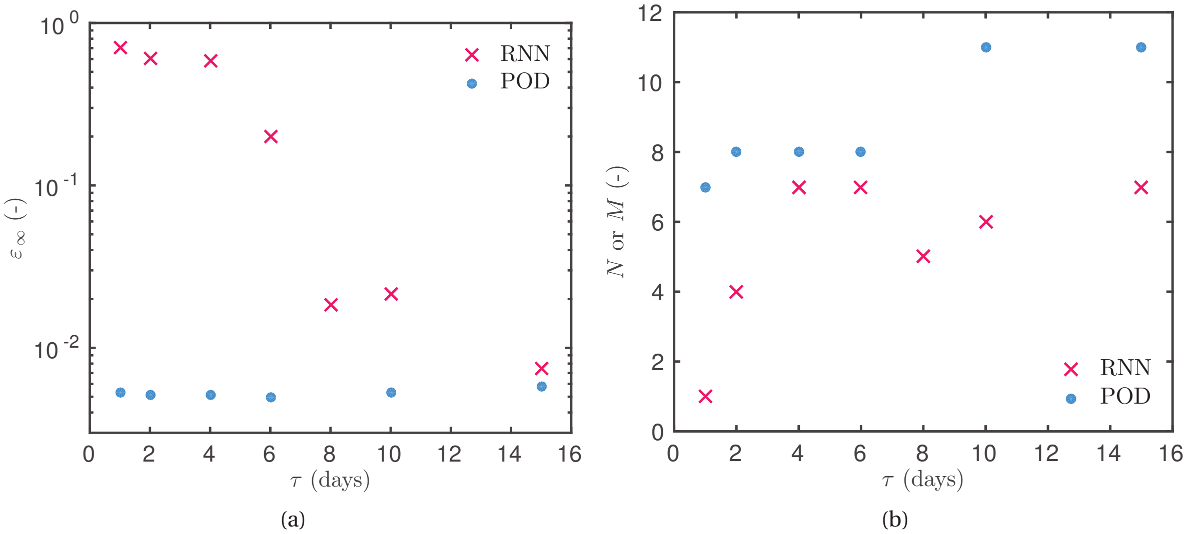


Figure 3.12: Variation of the error ε (a) and of the POD order and RNN nodes number (b) as a function of the duration τ of the training phase.

Complementary tests: improving the training climate

One important point of the intelligent co-simulation is the definition of the training climate in order to build a robust and accurate prediction model. In the previous case, the training climate was defined according to the hottest and coldest days. However, the outside radiation flux was not considered in this approach and might have an important influence on the accuracy of the prediction models. The improvement proposed here consist of creating a training climate considering the equivalent outside temperature:

$$T_e^{\mathrm{eq}} = T_e + \frac{q_e}{h_e}.$$

The equivalent outside temperature is calculated for the whole climate of the case study. The hottest and coldest days, based on a daily-averaged computation, are selected and ordered. Then, the training climate sequence is built according to the sequence order defined in Figure 4.13. The same procedure is adopted for the coldest days. This sequence enables to simulate the problem for the hottest

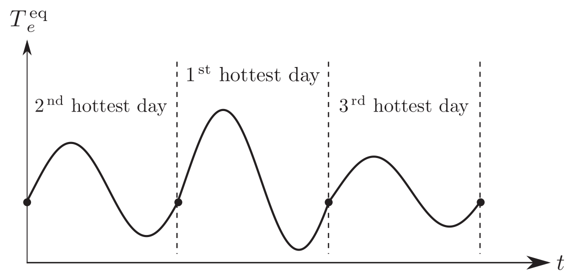


Figure 3.13: Definition of the improved training climate for the hottest days.

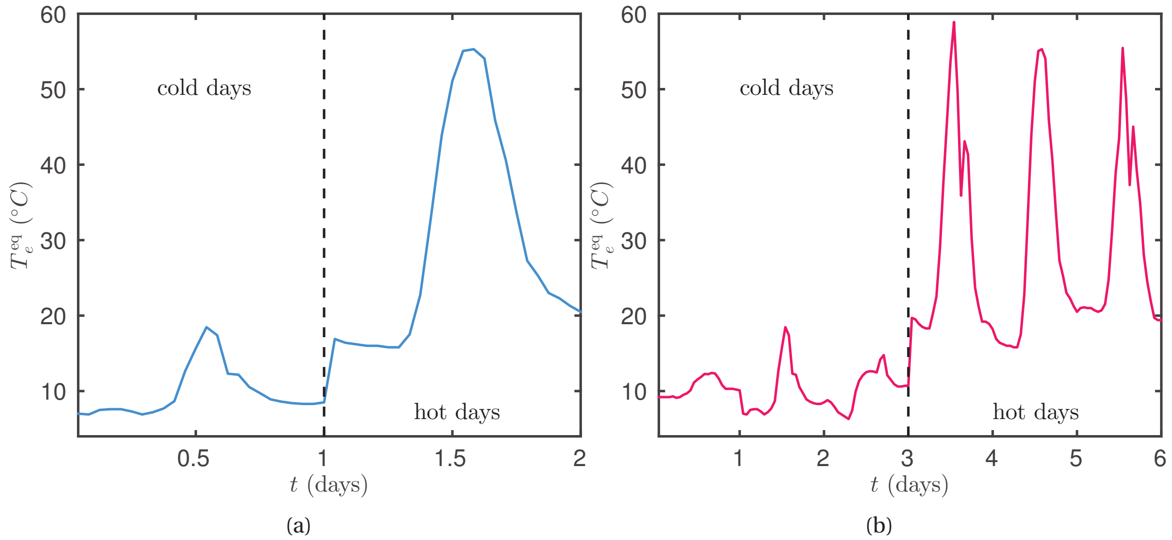


Figure 3.14: Variation of the outside equivalent temperature T_e^{eq} for $\tau = 2$ days (a) and $\tau = 6$ days (b).

days increase and coldest decrease of temperature, respectively. According to this procedure, the variation of the outside equivalent temperature is shown in Figure 3.14a, for $\tau = 2$ days and in Figure 4.14, for $\tau = 6$ days. It should be underlined that the equivalent outside temperature is only used to built the sequence of the training climate composed of the outside variations of temperature and solar radiation heat flux. The equivalent outside temperature is not used as boundary conditions for the training phase.

The intelligent co-simulation approach, with a training phase $\tau = 2$ days for the POD prediction model and $\tau = 6$ days for the RNN prediction model, is used to compute the solution of the case study. The time variation of the error is given in Figure 3.15. As for the previous case, the error remains at the order $\mathcal{O}(10^{-2})$. By comparing Figure 3.11 and Figure 3.15, the use of this improved training climate enables to reach a better accuracy to represent the physical phenomenon. A very good agreement can be noticed in Figure 3.16 between the solutions. Moreover, the computational time of the intelligent co-simulation is reduced due to the shorter training phase.

The error as a function of the duration of the training phase τ is given in Figure 3.17. With this improved training climate, only $\tau = 6$ days are now required to build an accurate RNN prediction model. For the POD prediction model, the minimal training period remains $\tau = 2$ days.

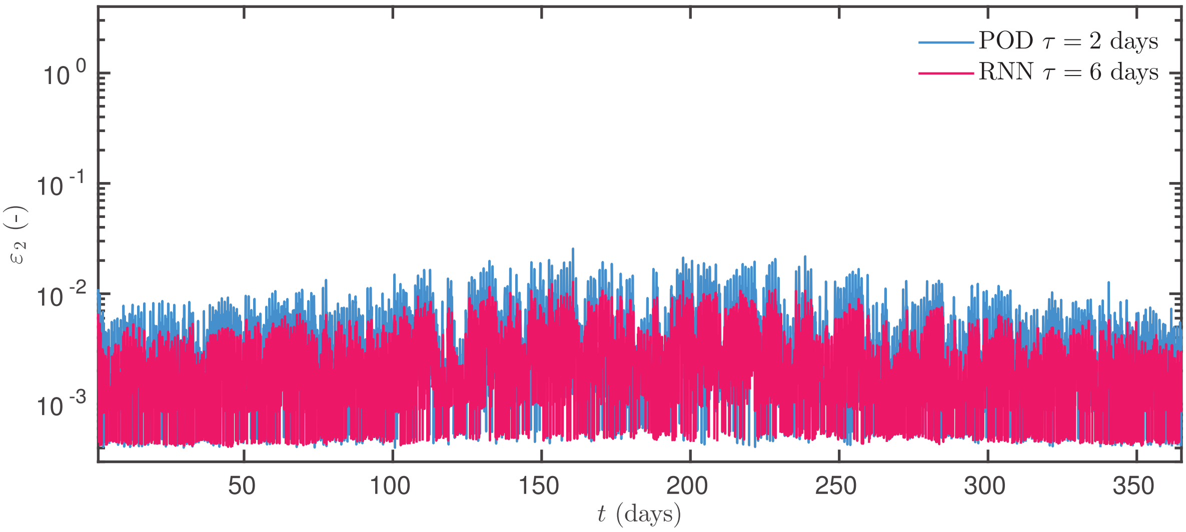


Figure 3.15: Time evolution of the error.

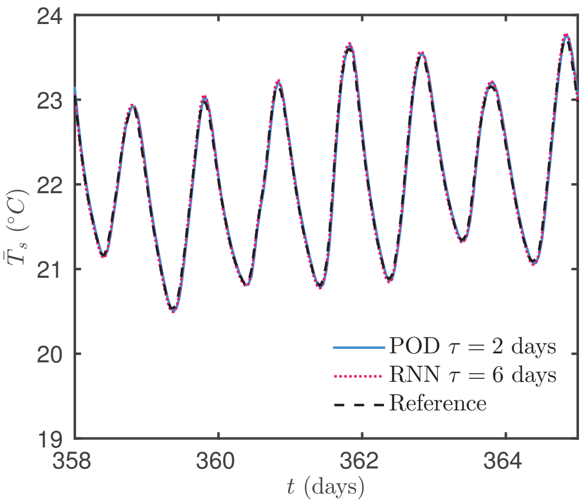


Figure 3.16: Evolution of the inside spatial average surface temperature during the last week of the year.

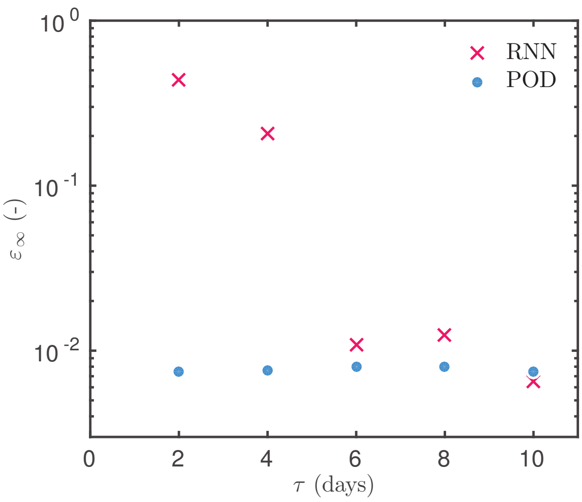


Figure 3.17: Variation of the error as a function of the duration τ of the training phase.

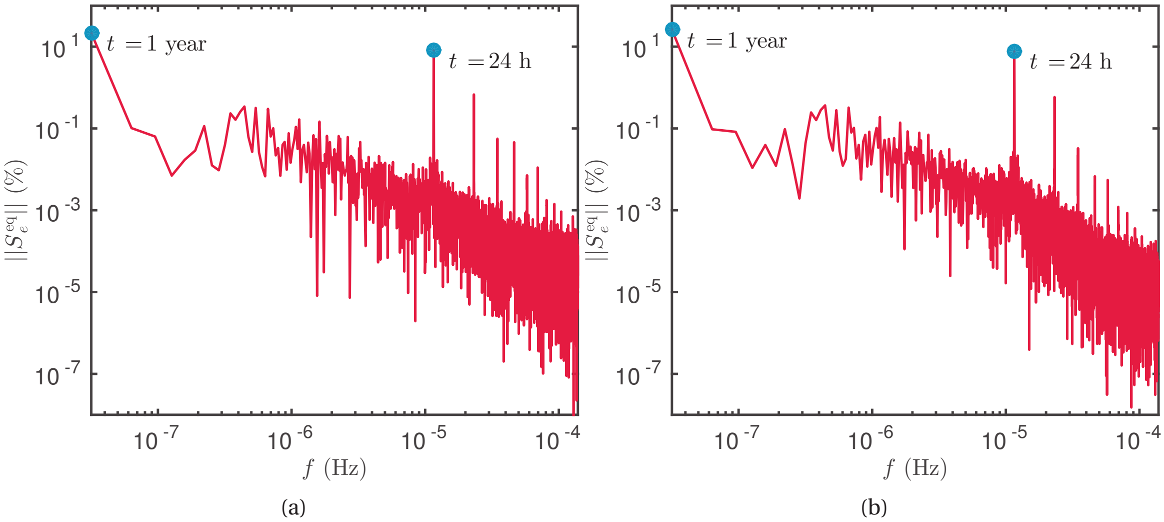


Figure 3.18: Power spectrum of the yearly outside equivalent temperature T_e^{eq} of Miami (a) and Paris (b).

Complementary tests: other climates

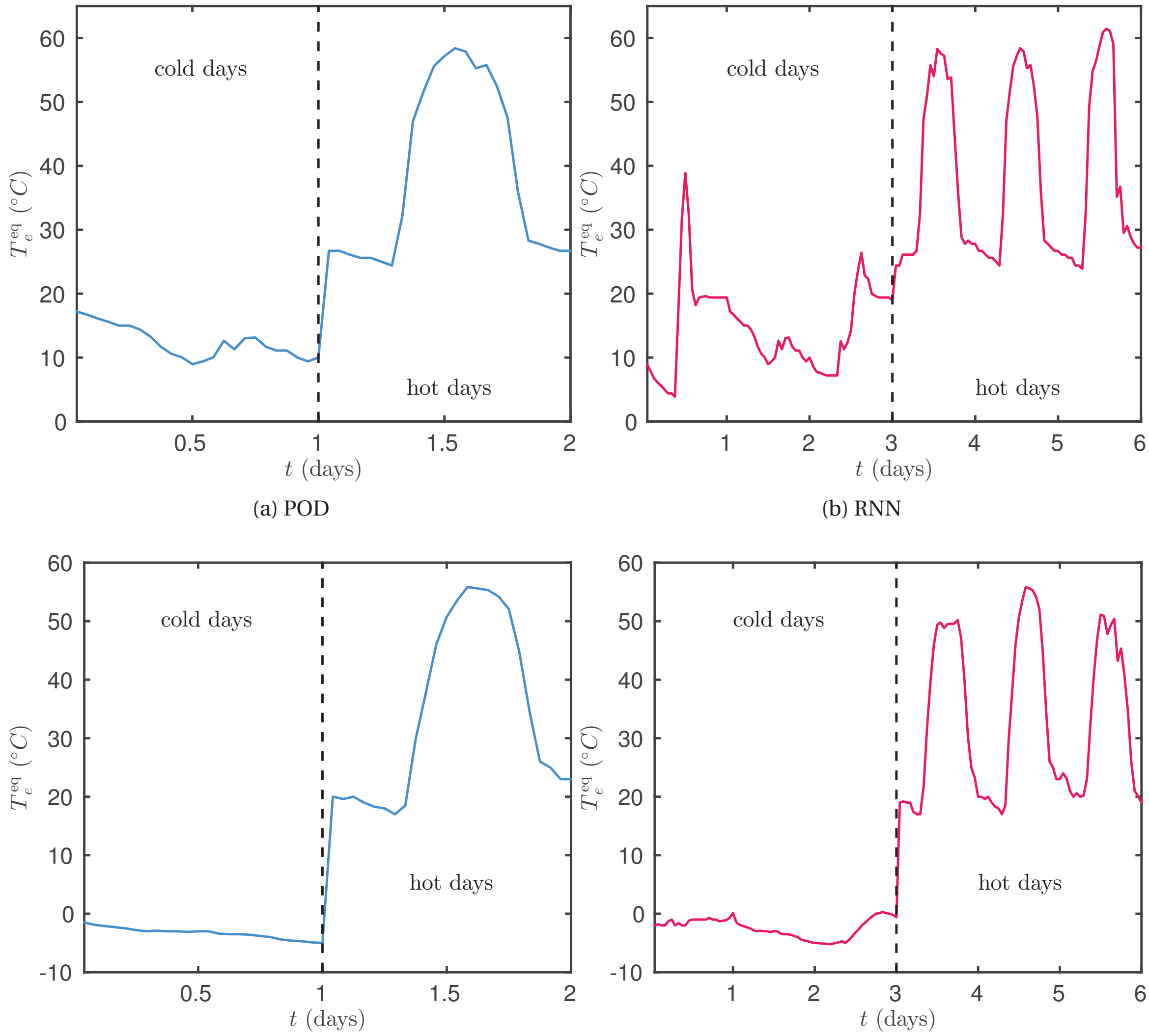
To evaluate the robustness of the intelligent co-simulation approach, complementary tests are carried out. The same setup as the one described in Section 3.4.4 is considered. The simulations are performed for two climates, Miami (Latitude: 25°N, Longitude: 80°W) and Paris (Latitude: 48°N, Longitude: 2.2°E), which are rather different from Curitiba. Miami and Paris have, respectively, low and high thermal amplitudes over the year.

As shown in Figures 3.18a and 3.18b, the frequency of 24h is responsible for 10% of the energy spectrum of the equivalent outside temperature for both climates. The training climate is defined according to the procedure described in Section 3.4.4. The variation of the equivalent temperature is given in Figures 3.19a– 3.19d for both climates. Therefore, from this analysis and from the previous results, the training phase is set to $\tau = 2$ days and $\tau = 6$ days for the POD and RNN machine learning models, respectively.

Figures 3.20a and 3.20b show the error of both prediction models with the reference solution computed with the normal co-simulation between `CFX` and `Domus`. For both cases, the error remains at the order $\mathcal{O}(10^{-2})$, highlighting a very satisfying accuracy. Moreover, it can be observed that the co-simulation enables to improve the representation of the physical phenomenon compared to the 1-dimensional simulation performed with the standalone `Domus` program. A good agreement is noticed between the solutions as shown in Figures 3.21a and 3.21b. In addition, the gains observed in terms of computational time are similar as the previous case. The intelligent co-simulation requires 10% and 15% of the classical co-simulation for the POD and RNN approaches, respectively. These results highlight the intelligent co-simulation, that is capable to compute an accurate solution of the problem, for different climates and enables to be confident according to the proposed methodology.

3.5 Conclusion

Building simulation tools have been coupled through a computational procedure known as co-simulation, in order to extend their capabilities and to perform accurate whole-building energy simulations. However, the total computational effort may increases exponentially, particularly when using



(a) POD (b) RNN

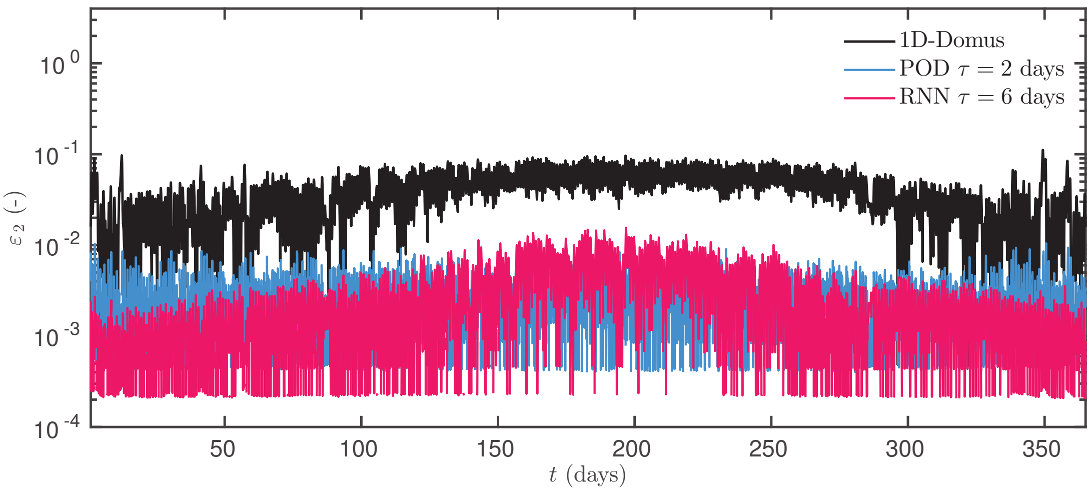
(c) POD (d) RNN

Figure 3.19: Variation of the utside equivalent temperature T_e^{eq} of Miami (a-b) and Paris (c-d) for τ = 2 days (a-c) and τ = 6 days (b-d).

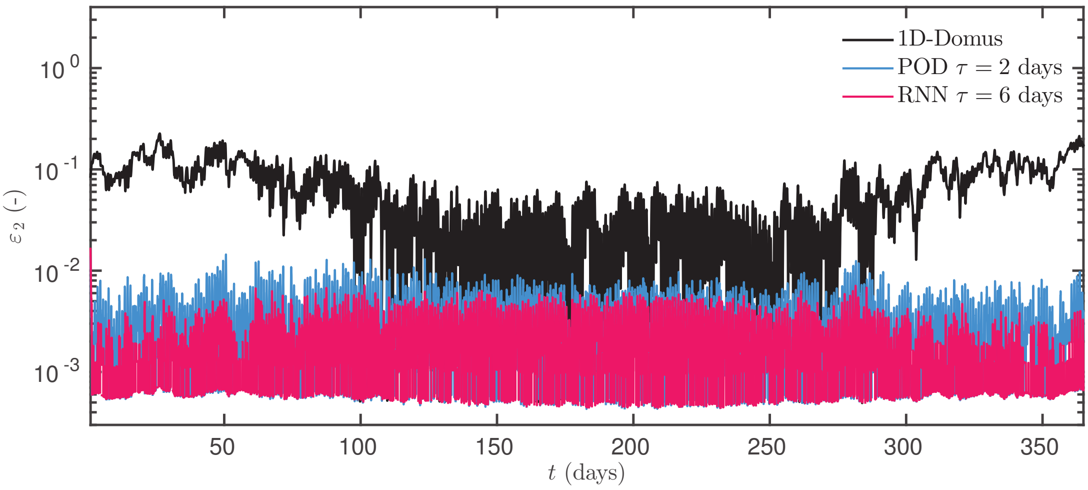
CFD software. In the investigated case study, the co-simulation required 14 h to perform a whole building simulation with 2-dimensional heat transfer.

In this context, this manuscript proposed to explore deeper an innovative approach, called intelligent co-simulation [61]. The method is based on the use of a machine learning model aiming at reducing the computational efforts. It is divided into two steps. During the training phase, the co-simulation operates normally between the two programs, while a training model learns from the results. After a long enough training period, the co-simulation is modified, replacing one of the programs by a prediction model, capable of computing accurate results, at a lower computational cost. This paper explored the features of two machine learning models, based on artificial neural network and POD model reduction method within the context of a co-simulation between `Domus` and `CFX` programs.

First, both machine learning models have been tested on a simple case study, to rapidly run and highlight intermediary and easy-to-understand results that could benefit, in principle, more complex building models. Both approaches enable to build accurate prediction models, based on a training

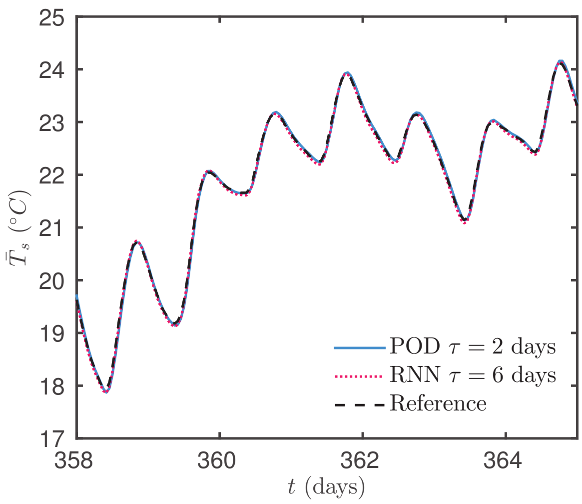


(a)

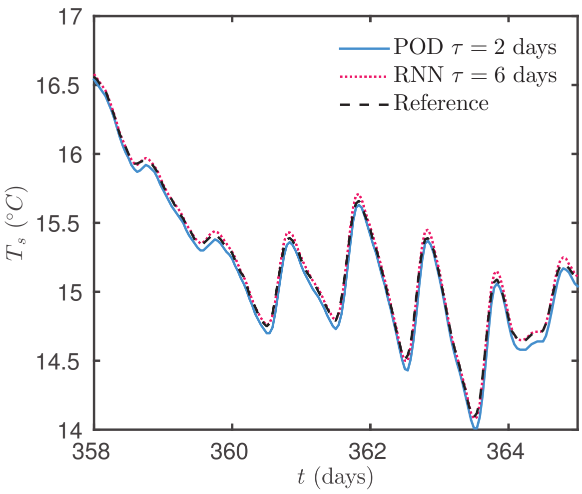


(b)

Figure 3.20: Time evolution of the error for Miami (a) and Paris (b) cases.



(a)



(b)

Figure 3.21: Evolution of the inside spatial average surface temperature for Miami (a) and Paris (b) cases during the last week of the year.

phase shorter than the simulation horizon of the considered problem. Moreover, the accuracy of the machine learning models increases when the training models receive a wide range of values of the interested field to build the prediction models.

From this intermediary results, a more complex case has been studied, based on the co-simulation between `Domus` and `CFX` programs, to perform a whole building energy simulation of 2-dimensional diffusive heat transfer. The POD and RNN machine learning models have been implemented in the `Domus` program, reducing the computational cost due to data exchange, during the training phase, while the co-simulation between `Domus` and `CFX` normally runs. At each iteration, `Domus` provides the boundary conditions to `CFX` that returns the internal surface spatially averaged temperature. The machine learning models learn from `CFX` results. The definition of the training climate is a crucial point in the methodology. Here, its definition is based on a sequence of hot and cold days, defined on an analysis of the daily average equivalent outside temperature.

At the end of this phase, the RNN training model builds an approximation function to compute the spatially averaged temperature. The POD training model builds a reduced order model that enables to compute the temperature distribution within the wall. Then, during the prediction phase, the machine learning models disconnect `Domus` and `CFX` and replaces `CFX`. The co-simulation is thus performed between `Domus` and either POD or RNN prediction models for the whole simulation time. This innovative approach of co-simulation has been tested for several configurations, in terms of outside climate and duration of the training phase, showing promising results. The intelligent co-simulation enabled to reduce by ten the computer run time compared to the classical co-simulation. Where the classical approach required 14h, the POD and RNN machine-learning-based models reduce the total time to 88 min and 135 min, respectively. The accuracy of the physical phenomenon is respected with an error, with the reference solution, at the order of $\mathcal{O}(10^{-2})$ for both approaches.

If the results are very satisfying for both approaches, the POD and RNN machine learning models have important differences. For the studied case, the POD based intelligent co-simulation seems to be more accurate and requires a shorter training phase (around 2 days for a total horizon simulation of 1 year). This advantage is due to the fact that the POD machine learning is based on a reduced order model built upon the governing differential equations of the problem. However, it limits the POD machine learning model to the physical phenomenon investigated, *i.e.*, the 2-dimensional diffusive heat transfer. If the RNN based intelligent co-simulation needs a longer training phase (around 6 days for a total horizon simulation of 1 year), the machine learning model is more generic. In other words, it does not depend on the physical phenomena considered. As it is implemented in the `Domus` program, it could be directly used for another case study. As an example, the same RNN based intelligent co-simulation was used in [61] to analyze 3-dimensional heat transfer by diffusive, radiative and convective mechanisms.

This article opens interesting opportunities for further works. This innovative approach of co-simulation may be tested for other physical phenomena such as the convective heat transfer in building enclosures. Other approaches should be tested, as for instance, radial basis functions [85] or time series forecasting [86] to compute approximation functions. An important aspect that deserves more research effort is the definition of the training climate to build a robust machine learning model. Nevertheless, the simulation time of the training phase has to be shorter to limit the computer run time of the co-simulation. To improve the training climate, some sampling techniques of the input parameters for the machine learning model should be also investigated as well as accuracy control algorithms.

Chapter 4

Intelligent co-simulation: complex 3D analysis considering non-uniform radiative boundary conditions

Paper to be submitted to the Journal of Building Performance Simulation on aug-2018
ISSN: 1940-1493 (Print) 1940-1507 (Online) Journal homepage: http://www.tandfonline.com/loi/tbps20
Impact Factor: 2.483, Qualis: A2. Scimago: Q1

4.1 Introduction

Presently, the Building Performance Simulation (BPS) tools are the only ones capable to perform whole-building energy simulation, where different physical phenomena need to be simultaneously simulated. Although developed since the 1970s, the BPS tools available nowadays are still commonly based on simplified mathematical models. For instance, most BPS tools consider only one-dimensional heat transfer through the envelope to calculate the indoor conditions and energy consumption, as reported in [3–6, 31–37]. Most of these tools are also based on the air lumped model to estimate the building zone air temperature. Not only within the building and its constructive elements, the simplifications also occur in the boundary conditions. For example, as presented in [87], the most common methods implemented in BPS tools for calculating the sunlit area on surfaces have great limitations with respect to the shape and complexity of the geometry. In addition, the computational time is usually high for the complex geometries and shadowing elements [88]. In behalf of these simplifications, the whole-building simulation can be performed in a reasonable computer run time.

However, simplification in the BPS physical models may lead to lack of accuracy to represent complex phenomena associated to non-uniform convective and radiative boundary conditions, thermal bridges and complex geometries. For instance, in [89] a discussion is provided about how complex three-dimensional heat transfer computations are important to properly describe the thermal behaviour of hollow blocks, widely used in structure of residential buildings and factories. In [90], the importance of sun patch and 3-D heat transfer on the accuracy of building thermal prediction and thermal bridges assessment is presented.

There is a wide variety of software to simulate a specific physical phenomenon with complex mathematical models providing, in general, accurate and reliable results, such as Computational Fluid Dynamics (CFD) tool. Although the tools are available, the main challenge is to explore these specific tools solving capabilities in BPS tools. One solution proposed in the literature is the so-called *co-simulation approach* [8, 9], also known as external coupling [12]. This strategy is a computational procedure that connects BPS tools with specialized software equipped with models based on advanced physics, so that it can be seen as an evolution of BPS tools [61]. In the field of building simulation, co-simulation can be described as the BPS tool, operating as a master, commanding the simulation and applying its simplified models for most of the physical phenomena covered by the simulation. However, for a specific phenomenon, a specialized tool with a complex physical model is used. As a consequence, there is a flow of data exchange between the tools during the entire simulation period.

The benefits of the co-simulation are certainly remarkable in terms of improved accuracy in building performance simulation. However, as discussed in [49] and [91], improved accuracy comes with a significant increase in the computational cost. As a first attempt to solve this drawback, in [49], an approach, called *intelligent co-simulation*, was proposed to reduce the computational burden of the co-simulation between a building simulation program and a CFD tool, while keeping a great level of accuracy. In short, the approach consists of designing an artificial intelligence-based model - called prediction model - for a given specific phenomenon, capable to provide results as close as possible to the ones provided the complex model implemented in the CFD tool, but much faster to simulate. The co-simulation strategy is further explored in [91], where the training time is reduced by applying features like improved training climates. Further references about co-simulation and *intelligent co-simulation* are given in Sections 4.2 and 4.3.

Aiming to bring more improvements of BPS analysis with advanced physics, including multidimensional effects, airflow and sun patch description, this chapter presents three contributions. First, a state-of-the-art simulation is presented considering the accuracy of a CFD tool to calculate the 3-D heat transfer and natural convection inside a two-storey building. Regarding the solar radiation, the pixel counting (PxC) technique implemented in the `Domus` software is used to locate with great accuracy the shape and surface region where the direct solar beam hits the building internal and external surfaces. To explore all the potential of the PxC technique, two trees described in terms of a complex geometry are used as shading elements, thus showing that even the most complex shadows can be included in the simulation of 3-D buildings by using the co-simulation technique. Related to the first contribution, the second one provides a new model for radiation in building enclosures that has been implemented in the `Domus` software to include the complete radiation evaluation at the internal surfaces. Besides the use of PxC technique, the radiation distribution model uses `Domus` view factor algorithm for distribution of reflected radiation and long-wave radiation heat exchange. The third contribution is the application of the *intelligent co-simulation* technique to this state-of-the-art case study, being the most complex application performed so far by means of this innovative technique, that enables to accurately and rapidly predict 3-D heat transfer and airflow in a more realistic way. Therefore, the present chapter demonstrates more deeply the predictive potential that *intelligent co-simulation* can bring to the field of building simulation.

The chapter is organized as follows. In Section 4.2, a brief literature review about the different subjects involved in this work is presented. The methodology is provided in Section 4.3, while co-simulation and *intelligent co-simulation* results are presented in Section 4.4.

4.2 Literature Review

In this section, a brief literature review about the different subjects involved in this work is presented. First details are presented on the importance of radiation models and on how the sunlit area is taken into account. Then, a review is conducted in terms of the importance of detailed 3-D thermal simulation on the accuracy of building performance simulation. Further details on the co-simulation procedure and applications in the field of building simulation are presented. Finally, it is presented an overview of the concept of Machine Learning and its applications.

4.2.1 Radiation Models

The accurate evaluation of solar radiation in building is essential to predict the buildings thermal behavior and energy consumption. Comparisons among BPS tools were performed in [92], revealing large differences in the solar gain calculation results. The main reason is the lack of calculation of reflections by most of the methods. Also described by [92], an appropriated method must take into account that a large part of the transmitted radiation can be lost through re-transmissions, via reflections or directly, through the glazed surfaces. As an example, related studies such as [93] conclude that the location where the solar beam radiation is projected can impact on the heating demand. However, according to [94], beam solar radiation through a window is generally taken into account by the available algorithms in an overly simplified way, considering that all the flux hits the floor. It is also common to assume time-invariant distribution coefficients: for example, 60% of the radiation is assumed to reach the floor and the rest is user-defined [90].

The most employed methods in BPS tools for calculating the sunlit area on surfaces are based on trigonometric and projection-clipping operations [95–99]. Programs such as ESP-r, BLAST, DOE-2, TRNSYS and EnergyPlus adopt those methods that may present numerical limitations in terms of accuracy and computational time when dealing with complex geometries [87]. According to [100], it is a common practice to model the geometry of the building in a more simplified way to generate compatible geometric models with the simulation tool, which can lead to large differences between simulation and reality. Themselves proposed an approach for direct solar shading calculations, called *pixel counting technique*, based on computer graphic methods (OpenGL) with hardware support. First proposed by [101], the pixel counting technique can calculate, for each time-step of simulation, the sunlit area on the building envelope. Basically the technique renders the building scene using orthogonal projection from the vantage point of the Sun and the sunlit area on each surface is computed by determining the number of visible pixels. For a detailed discussion of the technique, the reader may refer to [26, 87].

Pixel counting technique has been recently implemented in `Domus` software (Figure 4.1), that also presents a graphic interface for building modelling based on OpenGL. PxC was implemented in `Domus` following the original approach proposed by [101] in 1994 and the extended approach proposed by [26] in 2012. Currently, the pixel counting is used to calculate the sunlit fraction and direct solar energy on exterior surfaces. Though the implementation of PxC in `Domus` enables to evaluate the sunlit area on internal surfaces, this information has not yet been considered in the building energy balance, and this improvement is one of the contributions of the present work.

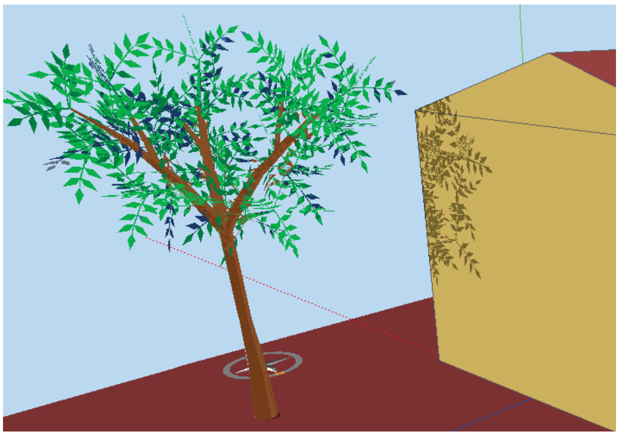


Figure 4.1: `Domus` using Pixel Counting technique for complex shading calculation for a tree represented by a very complex geometry.

4.2.2 Detailed Building Simulation

Contemporary BPS tools are typically used to predict performance of buildings for long periods such as a whole year. As a result, the representations of certain phenomena are simplified, or even neglected, for instance, the 1-D treatment of heat conduction through the building envelope. These simplifications certainly reduce the precision to represent complex physical phenomena associated to non-uniform conditions of convective and radiative boundaries, thermal bridges and complex geometries. Such subjects have been studied in many publications, for example, in [90] the effect of a more complex 3-D thermal analysis in low energy buildings is described. As presented in [90], the 3-D thermal envelope model is more accurate to describe the air, the surface temperature field and the heating load requirement. Comparisons with a 1-D thermal model and on-site experiments presented important discrepancies for the assessment of surface temperatures and heating power when using a simplified model. Moreover, a 3-D model is more adapted than 1-D models when the objective is to evaluate the behavior of surfaces composed of different materials with different thermal masses [102].

In [89], a study related to hollow blocks shows that complex 3-D heat transfer are indispensable to properly describe the thermal behavior, with 9% to 14% of total losses of a system due to thermal bridges. However, due to the prohibitive computational time necessary to use these complex models, reduced models have been proposed as an alternative to improve the evaluation of hollow block walls. In [64], a reduced model was also proposed, using Proper Generalized Decomposition (PGD) to predict the 2-D heat and moisture transfer through a building envelope. However, computer run time even with a reduced model for a purely diffusive problem has still been shown to be too expensive.

In addition, the evaluation of heat losses through thermal bridges and their effect on the overall building performance represent a difficulty frequently faced in building simulation [103]. The dynamic effect of thermal bridges on the building energy performance is evaluated in [104] using WUFI Plus software, modeling the thermal bridges using a 3-D dynamic method and the equivalent U-value method. Simulation results show that the annual heating load is underestimated by 12.5-14.8% using the equivalent U-value method compared to the 3-D dynamic method.

Computational Fluid Dynamics (CFD) is currently an option to perform more complex predictions in buildings physics. According to [105], the major difficulty with CFD - especially in three dimensions - is that the calculations are very slow and require large amounts of memory. An intermediate approach that allows to determine the airflow pattern inside a room without the computational investment of CFD is the “zonal method” which is adapted to a building domain characterized by great volumes. However, this method has shown a reasonable accuracy only for simple rectangular geometries.

Therefore, innovative and fast methods are needed to enable detailed simulation of buildings, including 3-D heat/moisture transfer and airflow simulation, considering non-homogeneous distribution of radiative boundary conditions.

4.2.3 Co-simulation

Co-simulation can be technically defined as a type of simulation where at least two simulation tools - each one offering different numerical solutions to a physical or mathematical problem - jointly solve differential-algebraic systems of equations, exchanging data while the coupling is active ([47], [48], [49]).

Some examples of improvements in building analysis and life quality can be found in [10, 61, 64]. These improvements could be achieved with the help of a BPS tool when buildings and their systems are treated as a complete optimized entity. It is a fact that the development team of a BPS tool needs to have an in-depth knowledge of the software architecture, programming language, hardware support, modelling approaches and strategies, and due to the rapid innovations in building and system technologies, to keep a BPS tool continuously updated is a hard task. In this context, co-simulation can be an interesting alternative [11].

There are several co-simulation techniques and combination of methods such as the *one-to-one approach* and the FMI standard. In the *one-to-one approach*, a specific protocol regulates the exchange of data during run time. It can be considered as the most flexible approach and can support any kind of numerical solution method [106]. For instance, an one-to-one coupling was adopted to connect `Domus` to the commercial CFD tool ANSYS-CFX. The type of coupling used was the Ping-Pong method [8], creating a weak coupling between the tools. The *one-to-one approach* requires a specific implementation in each of the tools to be coupled. Such demands requires implementation efforts over time to keep the compatibility between the tools. To overcome these limitations, an open-source standard for coupled simulation, known as FMI Standard, was proposed by Modelisar project in 2008. The use of the FMI standard provides greater scope for co-simulation between tools that adopt the same standards such as EnergyPlus, Modelica Models and many others available at `https://fmi-standard.org`.

As an interesting example of co-simulation, one can mention the work with multidimensional effects, natural convection airflow, heterogeneous convective heat transfer coefficients presented in [61]. It is shown how the zone temperature prediction can be highly improved in terms of accuracy with the co-simulation between a BPS and a Computational Fluid Dynamics (CFD) tool. The results of a co-simulation of a triangular-prism-shaped attic model showed that a good improvement in terms of accuracy can be achieved, although the technique presents as a drawback a high computational cost. In [64], a co-simulation was performed between a building simulation program and an in-house reduced order model for simulating 2-D heat and moisture transfer through a porous wall. Among other examples arising from the co-simulation in building physics, one can mention the one presented in [39] that proposes a co-simulation to deal with the interactions among room airflow, heat, ventilation and air conditioning (HVAC) systems and building envelope, within the outlooks of management of systems. In [10], the reduction on emission of greenhouse gases and the substantial improvements on health, comfort and productivity based on co-simulation between tools is mentioned. A more detailed review of co-simulation examples can be found in some recent works in the frame of the IEA Annex 60 [47] as well as [9] and [91].

4.2.4 Machine Learning

Currently, with the growth of the capacity to obtain and store data, the opportunity and need to work with data substantially rise. This availability of data calls for automated methods of data analysis, which is precisely what *machine learning* provides. Machine learning can be defined as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty [107]. There are two main types of machine learning. The supervised and the unsupervised learning approaches. As defined in [108], the former is characterized by the concept of teacher or supervisor, whose main task is to provide the measure of prediction errors directly comparing with the output variables. In the supervised scenario, the goal is training a generalised system that must work with the training and test data set but also with samples never seen before. According to [108], the unsupervised approach is based on the absence of any supervisor and, therefore, of absolute error measures. It is useful when it is necessary to group a set of data according to their similarity.

Neural Network

One of the well known classical learning techniques is the Artificial Neural Network (ANN), that was proposed for creating a computational learning procedure similar to the learning process of a human brain. The main objective is the generalization of complex rules and mathematical routines, learning the key information patterns within a multidimensional information domain. ANNs operate like a "black-box" model, requiring no detailed information or property of what is inside the box [40]. There are two major types of neural networks, feedforward and recurrent (see [41] for a short characterization of these two types of ANN). In the feedforward network, the input data goes through the network in a one-way direction to generate an output result. A recurrent neural network (RNN), on the other hand, has one or more cyclic path of synaptic connections. While the feedforward network has the ability to implement only static input-output mappings (functions), with recurrent neural networks it is possible to implement dynamic systems. Another important feature of neural networks is the presence of the activation functions. These functions are placed before each neuron and determine the activation of the neuron. The activation is dependent on the network input and threshold value. There are several possibilities to choose activation functions and some of the most commons are presented in [42].

The RNN is a class of ANN with the presence of cyclic connections in its neurons. This structure provides to RNNs the characteristic of having *memory*, where the past input data can influence the actual network output. There are two subtypes of RNN related to the training approach: supervised and unsupervised training. In the supervised training, there is a presence of target data to guide the training process, while, in the unsupervised one, there is no such target data.

As described in [43], neural network learning could be used as an universal function approximation. Moreover, adding more hidden neurons is equivalent to adding more basis functions. Other parameters like the number of layers, the number of training samples, the length of learning period, the choice of neuron activation functions, and the training algorithm could also affect the training accuracy. Interested readers may refer to [42, 44] for a better comprehension on NN and RNN.

Machine Learning in building simulation

Machine learning can be applied in many areas like engineering, applied mathematics and computer science. Some examples in the building simulation context are the works presented in [49] and [91], where a BPS and a specialized tool are coupled through co-simulation and the data exchange produced by this co-simulation was used to design a prediction model based on both RNN and Proper Orthogonal Decomposition (POD). In both works, the results show great capacity of accurate prediction of the physical phenomena. Other examples of applications in this field are presented in [40] that used an ANN to learn how to predict the required heating load of buildings with the minimum of input data. In order to forecast building energy consumption, in [45], a comparison between a simple model based on ANN and a model based on physical principles as an auditing and predicting tool is presented. More recently, in [46], a multidimensional model, very time-consuming, to calculate the coupled heat, air, and moisture transfer through a building envelope is adopted. The data provided by the analytical algorithm was then used to train a neural network model to predict temperature, vapor pressure, and relative humidity profiles.

4.3 Methodology

To show the potential of the co-simulation for detailed analysis of thermal and energy performance, this section first presents the radiation distribution model implemented in `Domus` to include a more complete and accurate information of radiation as a boundary condition at the internal surfaces. Then, a detailed building model is described. The last part of the methodology presents the necessary development for application of the co-simulation and *intelligent co-simulation* approaches in the detailed building model.

4.3.1 Radiation Model Improvements

As part of the development of this work, some improvements have also been done in the internal radiation calculations in software `Domus`. In the previous formulation, all the radiation received by the internal walls was distributed on the floor. Now, using the PxC technique, it is possible to identify precisely the sunlit area on each inner surface of the zone and accurately and rapidly include it into the Domus wall energy balance. Another functionality improved is the use of the view factor numerical calculation among all internal surfaces of the zone. Until now this information was only used to calculate more accurately the long-wave radiative heat exchange among surfaces of the zone. These modifications have been definitively included into Domus, bringing more accuracy to the simulation. This section describes the equations implemented in Domus for the improvement of the radiation model, however, only the information regarding direct, diffuse and reflected radiation is used in the case study presented in this work, since the long-wave calculation is performed by the CFD tool itself, which has a fast and accurate algorithm for long-wave radiation, based on the Discrete Transfer Model [60].

The radiative heat fluxes implemented in Domus as part of the building envelope and internal walls boundary conditions - considering the surfaces as diffuse and gray - are shown below.

First, the radiation absorbed by an external surface k, is given by:

$$q_{SW,k} = q_{dir,k} + q_{diff,k} + q_{refl,k}. \tag{4.1}$$

The direct solar radiation on external surfaces is calculated as:

$$q_{dir} = \frac{A_{sunlit}}{A}\cos(\theta)\alpha_k I_{dir}, \tag{4.2}$$

where I_{dir} is the intensity of the beam radiation, A_{sunlit}, the sunlit area of the surface, A the surface total area, θ, the solar beam incidence angle and, α, the surface absorptivity coefficient. The term $\frac{A_{sunlit}}{A}\cos(\theta)$ is obtained by the pixel counting technique and is valid only for 1-D analysis by Domus. In the evaluation of radiation in a 3-D approach, as described in section 4.3.2, the value of $A = A_{sunlit}$ for the sunny part of the surface and $A_{sunlit} = 0$ elsewhere.

The diffuse radiation on an external surface k can be calculated as:

$$q_{diff,k} = \sum_{j=1}^{N_{ext}}\left[(1+\delta_j)V_{F,k\to j}\right]\alpha_k I_{diff}, \tag{4.3}$$

where δ_j is the reflectivity of the neighboring surface j and $V_{F,k\to j}$ is the view factor from surface k to the neighboring surface j, N_{ext} is the number of external surfaces that are considered.

When surrounding surfaces are not taking into account, Eq. (4.3) becomes:

$$q_{diff,k} = \frac{(1+\cos(\beta_k))}{2}\alpha_k I_{diff}, \tag{4.4}$$

where β_k is the k-surface tilt angle and I_{diff} is the intensity of the diffuse solar radiation.

The reflected short-wave radiation on an external surface k is evaluated as:

$$q_{refl,k} = \alpha_k \sum_{j=1}^{N_{ext}}\left[\delta_j(\cos(\theta_j)I_{dir} + I_{diff})V_{F,k\to j}\right], \tag{4.5}$$

where θ_j is the angle between the solar beam and the surface j normal vector (unitary).

Across non-opaque surface, the direct solar radiation is multiplied by the optical transmittance (τ_g) that is a function of the type of the glazing system and the solar incidence angle (φ_g) on the translucent surface. The other radiative components - reflected and transmitted - are multiplied by the optical transmittance for an incidence angle of 60^o. In this way, we have:

$$q_{refl,k,\tau} = \tau_{g,\theta=60^o}\delta\frac{q_{refl,k}}{\alpha_k}, \tag{4.6}$$

$$q_{diff,k,\tau} = \tau_{g,\theta=60^o}\gamma\frac{q_{diff,k}}{\alpha_k}, \tag{4.7}$$

$$q_{dir,k,\tau} = \tau_g(\theta)\frac{q_{dir,k}}{\alpha_k}, \tag{4.8}$$

where δ and γ are the radiation filtering coefficients for diffuse and reflected transmitted radiation, respectively.

For the internal surfaces, the radiation that hits the surface k is given by:

$$q_k = q_{dir,k} + q_{diff,k} + q_{refl,k} + q_{dr,k}, \tag{4.9}$$

$$q_{dir,k} = \frac{A_{sunlit,k}}{A_k}\cos(\theta_k)\tau_g\alpha_k I_{dir}, \tag{4.10}$$

$$q_{diff,k} = \sum_{j=1}^{N_{gs}} \frac{V_{F,j\to k}A_j\alpha_k(q_{diff,j,\tau} + q_{refl,j,\tau})}{A_k}, \tag{4.11}$$

$$q_{refl,k} = \sum_{j=1}^{N_{int}} V_{F,j\to k}(1-\alpha_k)\frac{A_{sunlit,j}}{A_j}\cos(\theta_j)\tau_g\alpha_k I_{dir}, \tag{4.12}$$

$$q_{dr,k} = \sum_{j=1}^{N_{int}} V_{F,j\to k}q_{diff,j}\alpha_k\frac{\rho_j}{1-A_jA_k}V_{F,j\to k}\rho_j, \tag{4.13}$$

where N_{int} indicates the number of internal surfaces in the zone and N_{gs} represents the number of internal glazing surfaces in the zone. The surface reflectivity ρ depends on the material properties and can be defined as:

$$\rho = (1-\alpha), \tag{4.14}$$

for opaque surfaces, while and for translucent surface is:

$$\rho = (1-\tau-\alpha). \tag{4.15}$$

The long-wave radiative heat flux (W/m^2) for internal or external surfaces is calculated as [109]:

$$q_{LW} = \sum_{j=1}^{N_s} h_{rad,k\to j}T^0_{s,j}, \tag{4.16}$$

$$q_{LW,A} = \sum_{j=1}^{N_s} h_{rad,k\to j}, \tag{4.17}$$

$$h_{rad,k\to j} = 4\sigma\frac{\left(\frac{T^0_{s,k}+T^0_{s,j}}{2}\right)^3}{\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right)+\frac{1}{FF_{k\to j}}+\left(\frac{A_k}{A_j}\right)\left(\frac{1-\varepsilon_j}{\varepsilon_j}\right)}. \tag{4.18}$$

where ε is the emittance of surface, σ is the Stefan-Boltzmann constant (5.67 x 10^8 kg$s^{-3}K^{-4}$), T^0_s is the surface temperature [K] on the previous time-step. In order to improve the solution convergence in `Domus` wall TDMA solver, the long-wave radiation is included into two parts, q_{LW} included in the source term part of the equation and $q_{LW,A}$ added to the main diagonal coefficient part. The same Equation (4.16) is used to calculate the external long-wave radiative heat exchanges. However, in this work, the focus is stricted to internal radiative heat exchanges.

4.3.2 Detailed Building Model

The detailed building model presented as case study in this work comprises a two storey building as illustrated in Figure 4.2. With approximately 723 m^3 total volume, the building is composed by four 3-mm glazing windows, 1 wood door, a 20-mm tiled roof, 100-mm concrete walls, a 2 m x 4 m opening between the two storeys with a 13-step ladder (2 m x 0.1 m x 0.3 m), as depicted in Figures 4.3 and 4.4.

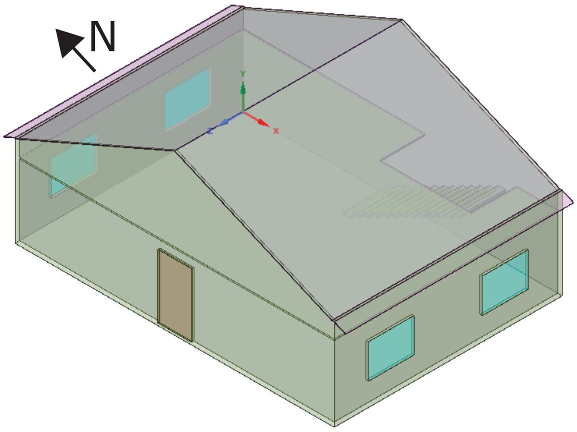


Figure 4.2: Isometric view of the case study in the SpaceClaim graphical interface.

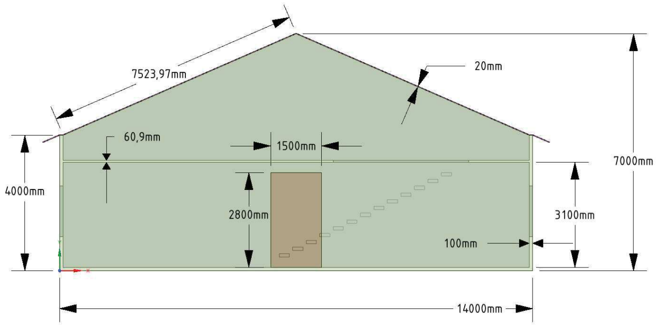


Figure 4.3: Details of case study building - west view.

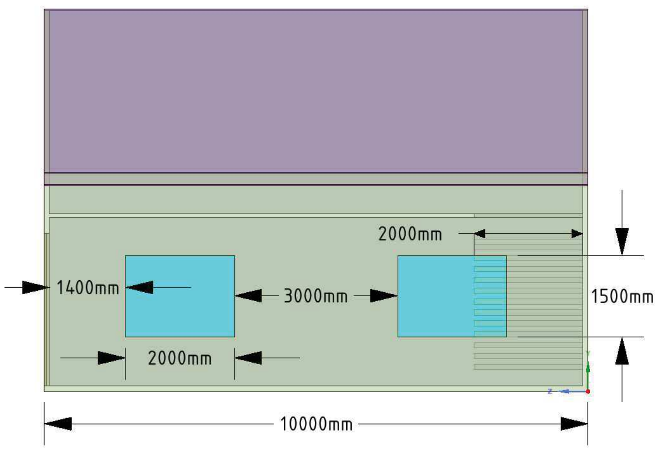


Figure 4.4: Details of case study building - south view.

Table 4.1: Range of parameters used in simulation.

Parameter	Material			
	Air	Roof Tile	Concrete Slab	Wood
Density (kgm^{-3})	1.200	2000	2200	720
Specific heat ($Jkg^{-1}K^{-1}$)	1007	920	1000	1255
Thermal conductivity (Wm^{-1}K^{-1})	0.024	1.05	1.75	0.16

In the external environment of the building, complex shading elements are inserted, composed of 2 trees near the northeast and northwest of the building (Figure 4.5), located in the southern hemisphere. The aim is to evaluate the effect of natural convection, three-dimensional heat transfer, sun patch on surfaces and also the effect of complex shadings, by means of a highly detailed simulation with the state-of-the-art in terms of solver for building energy and thermal performance and radiation calculation as a boundary condition on internal and external surfaces.

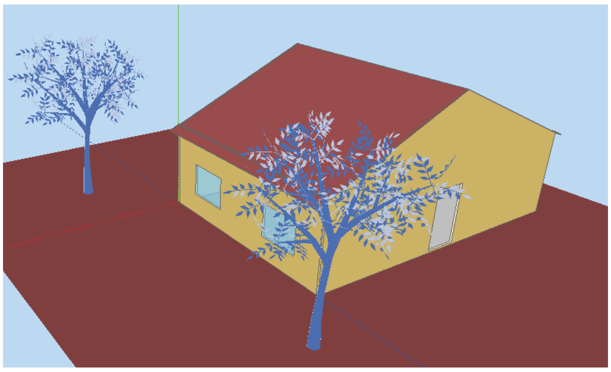


Figure 4.5: Case study with complex shadings in the Domus graphical interface.

For the CFD, an adaptive mesh is generated for the building geometry and, based on a grid convergence index (GCI) calculation [28], with limitations imposed by the available computational capacity, a mesh of 420 thousand nodes presents good mesh independent results, since a greater refinement does not bring considerable difference in the results of the model. Special attention was given to the refinement of the walls most affected by the sunlit, using a more refined mesh for them. This refinement is important to improve the quality of the sunlit contour geometry for CFX. Further information on the relationship of the mesh refinement with the contour condition is presented in the Section 4.4. The thermal parameters of different materials used in the simulation are listed in Table 4.1 and Table 4.2 for glass material. Simulations were performed using a computer equipped with Windows 8 64 bit and an Intel Core i7 4790 Processor with a four-core CPU at 3.6 GHz. The CFX used three cores for the simulation. The co-simulation time step was set to 6 min, considerably low in order to better represent the sunlit movement along the surfaces, since the sunlit will be updated at each step of communication between the tools. The total horizon time of co-simulation was 2 weeks (14 days).

CFD Model Setup

The numerical solution method is defined as a high-resolution transient scheme that alternates between the use of second-order and the first-order backward Euler scheme. The convergence criterion is defined as a RMS of 10^{-3}, considered as a loose convergence criterion but sufficient to fulfill the objective of this case study, as the building model represents mainly the natural convection phe-

Table 4.2: Glass properties.

Glass properties	
Absorption Coefficient (m^{-1})	7.6
Electrical Conductivity (Sm^{-1})	$10e^{-3}$
Density (kgm^{-3})	2200
Refractive Index (-)	1.526
Scattering Coefficient (m^{-1})	0.1
Specific Heat Capacity ($Jkg^{-1}K^{-1}$)	750
Thermal Conductivity ($Wm^{-1}K^{-1}$)	1

nomenon and there is no external air exchange or infiltration into the building. Focusing mainly on the solution stability, it is adopted the k-Epsilon turbulence model, that is represented by the k-equation (Equation (4.19)) and the ε-equation (Equation (4.20)):

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j k) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + P_k - \rho\varepsilon, \tag{4.19}$$

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j \varepsilon) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + \frac{\varepsilon}{k}(C_{\varepsilon 1} P_k - C_{\varepsilon 2}\rho\varepsilon). \tag{4.20}$$

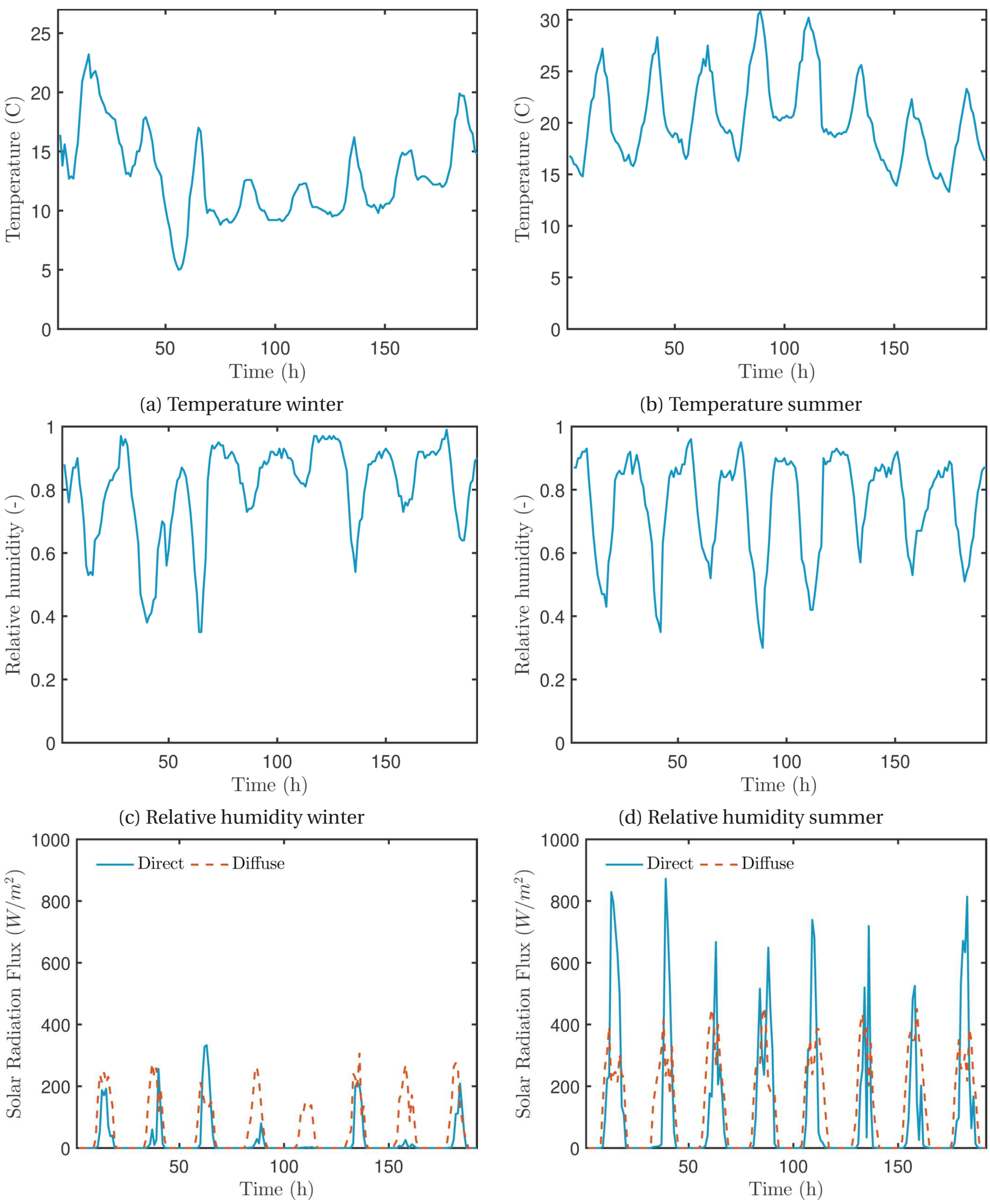
As described in [60], in addition to the independent variables, the density ρ and the velocity vector U are treated as known quantities from the Navier-Stokes method. P_k is the production rate of turbulence and the constants have the following values: $C_{\varepsilon 1} = 1.44$, $C_{\varepsilon 2} = 1.92$. The Boussinesq hypothesis is considered, applying the effect of gravity in the buoyancy term. For the boundary conditions, a non-slip condition is imposed at solid surfaces with a third-type boundary condition. The time-step in the `CFX` solver is set to 4 seconds, value empirically defined as the minimum required to ensure complete convergence of CFD simulation.

Boundary Conditions

By means of the co-simulation, `Domus` provides all boundary conditions to `CFX` . Providing the weather temperature, external convective heat transfer coefficient, direct, diffuse and reflected radiation fluxes, and the sunlit contour on each internal and external surface of the building.

The initial temperature is $T_i = 20\ °\mathrm{C}$. The convective heat transfer coefficients are set to $h_e = 23\ \mathrm{W/(m^2.K)}$. The outside conditions represent a temperate climate of Curitiba, Brazil. The inside temperature and relative humidity vary freely. Due to the extremely high computational cost, a two-week simulation period was chosen. One week including the winter solstice in the southern hemisphere, starting at June 21, and the other one on the summer solstice, starting at December 21. The objective is to closely analyze the results on days 21 and 22 to verify the thermal-physical behavior. The complete result of day 21 to 27 are used to evaluate the prediction model accuracy. Climatic data of the two weeks, taken from the `Domus` database, are shown in Figures 4.6a - 4.6f.

Exploring the precision of the Pixel Counting technique implemented in `Domus`, it is possible to obtain the shaded area of any complexity [87], which appears in Equations (4.2), (4.10) and (4.12). The most challenging task of this work is to obtain the contour of the sunny area and transport it to `CFX`. To perform this task, a 3-step process is devised and applied to each surface: once the PxC algorithm finished the computation of the sunlit area, each pixel (stored in the framebuffer) of the surface under analysis that lies in the sunlit area, is deprojected to the world coordinates and saved in a matrix of



(a) Temperature winter

(b) Temperature summer

(c) Relative humidity winter

(d) Relative humidity summer

(e) Radiation winter

(f) Radiation summer

Figure 4.6: Outside conditions for temperature, relative humidity and solar radiation flux.

coordinated points. Having the sunlit matrix of points complete, the second step consists of checking which face elements of the CFD building mesh are coincident with the obtained points. In order to carry out this second stage, software SALOME [110] is used by means of python scripts to locate the mesh elements reached by each point and create a group with these face elements for each surface. The third and last step occurs in CFX, where a modified version of the mesh is received, containing the original mesh plus the new sunlit groups. By means of CEL (CFX Expression Language) the solar beam radiation is applied only in the regions indicated by the sunlit groups, that logically induces temperature gradients that evidence the multidimensional nature of heat transfer in buildings.

Figure 4.7 illustrates an example of complex shadowing over the case study building. In Figure 4.8a, it is shown the result of Domus PxC for the north facade projected with respect to the Sun point of view. Note that it is necessary to perform a deprojection of each pixel in the sunlit area to the world coordinates (where the original geometry is defined) and then extract the coordinate points (Figure 4.8b). The screen resolution setup for the Domus pixel counting method determines the number of points to be exported. For this model, a value of 512 X 512 pixels is set following the lines exposed in [87] where resolutions greater than 768 X 768 produce almost the same results. However, the size used in the building mesh (defined in CFX model) is the main limiter of the precision in the final result.

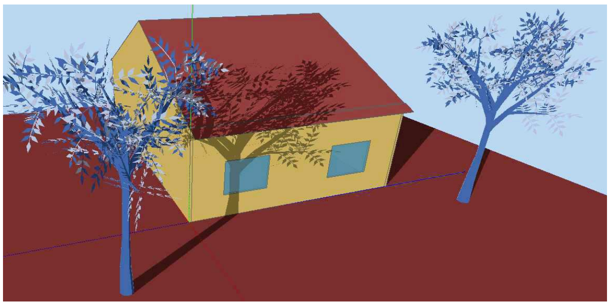
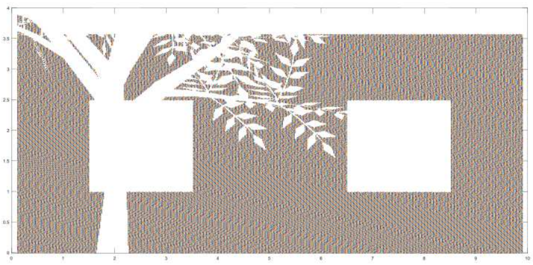


Figure 4.7: Shadowing pattern example in the Domus interface.

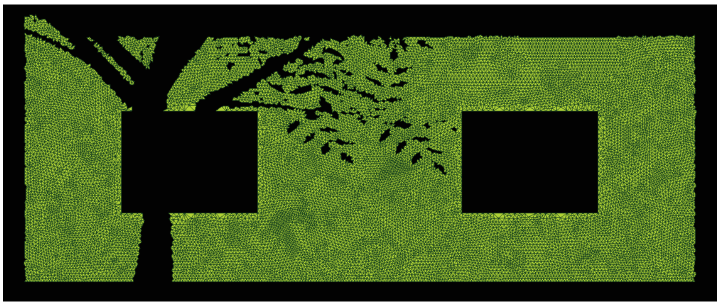
(a)



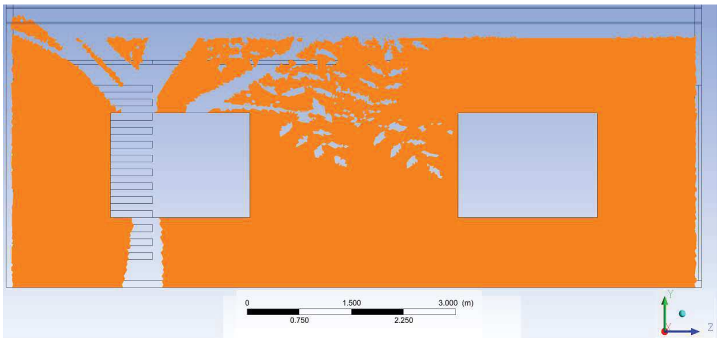
(b)

Figure 4.8: Domus projected shadowing over the north facade (a) and plot of the matrix with deprojected points (world coordinates) (b).

In second step performed in SALOME, some complications arise from the effect of pixellation and deprojection. These phenomena generate small differences between the coordinates of the matrix of sunlit points and the original geometry, not allowing a perfect match to locate the affected mesh elements. To solve this issue, some rotation, projection and translation operations are conditionally used over the matrix of points to fit their position exactly over the original geometry of the building. Figure 4.9a illustrates the final result of the second step, with the mesh elements selected in the original mesh of the building. These elements are then assembled and inserted as a group of elements in the building mesh.



(a)



(b)

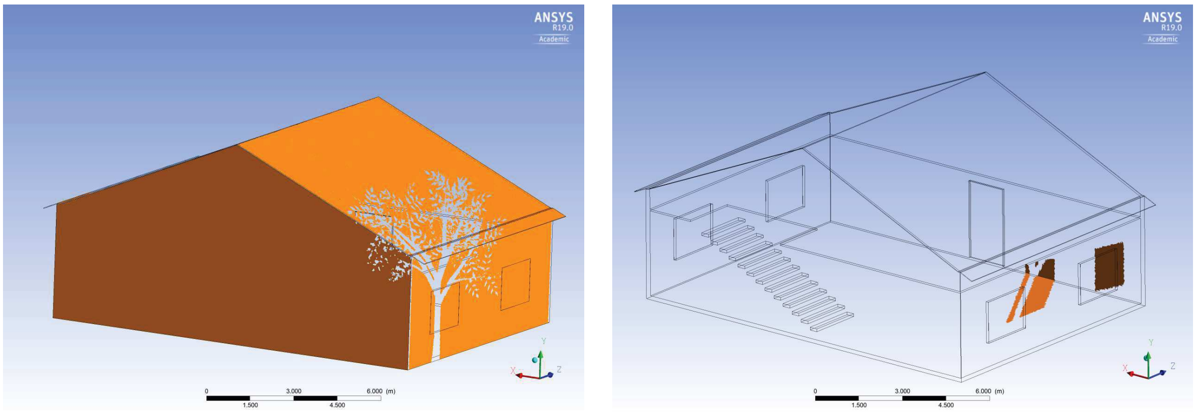
Figure 4.9: SALOME locating mesh faces based on coordinate points provided by Domus (a) and final mesh version with sunlit (orange) group identified over the building geometry in CFX (b).

In the third step, performed in CFX, also by means of scripts, the combination of the group of faces identified beforehand with the process of definition of boundary conditions is performed. In this step, using CEL commands, the direct radiation supplied by Domus for each face is applied only in the areas coincident with the elements of the mesh group that represents the sunlit area. Figure 4.9b illustrates the region of the mesh that receives solar beam radiation in this example. At this stage, it is common to have aliasing and some deformations in the form of the sunlit, because the results are dependent on the mesh refinement on the surface. Figures 4.10a and 4.10b illustrate a complete example of solar beam incidence patch on the external and internal surfaces. One of the most appealing feature of this approach is that the whole process is concentrated in Domus, with this tool performing the entire process, invisible to the user.

The direct solar radiation calculated by Equations (4.2) and (4.10) when coupled to a 3-D heat transfer code does not uniformly distribute the radiation over the whole surface. The direct solar radiation is ratter concentrated only on the sunlit surface, which enables to predict more accurately the 3-D heat transfer processes that occur in the building.

Model Ouputs

The outputs configured in the model consist of 12 fixed points inside the building (called sensors), and the precise position of these points are indicated in Figure 4.11. Moreover, the coordinates of the positions in relation to the origin of the building are presented in Table 4.3. The first option used for



(a) (b)

Figure 4.10: Example of final result of sunlit projection (in orange color) on external surfaces (a) and on internal surfaces (b) of the building.

the sensors was to use a small copper sphere of 1 mm radius. However, this option produced a very fine air mesh, consequently making the simulation much slower. The final option was to get the data directly from the mesh elements where the sensors are arranged. The data collected on these sensors are air temperature and velocity. In addition, the airflow between the two zones is also collected with the objective of evaluating the convective heat exchanged between the two zones.

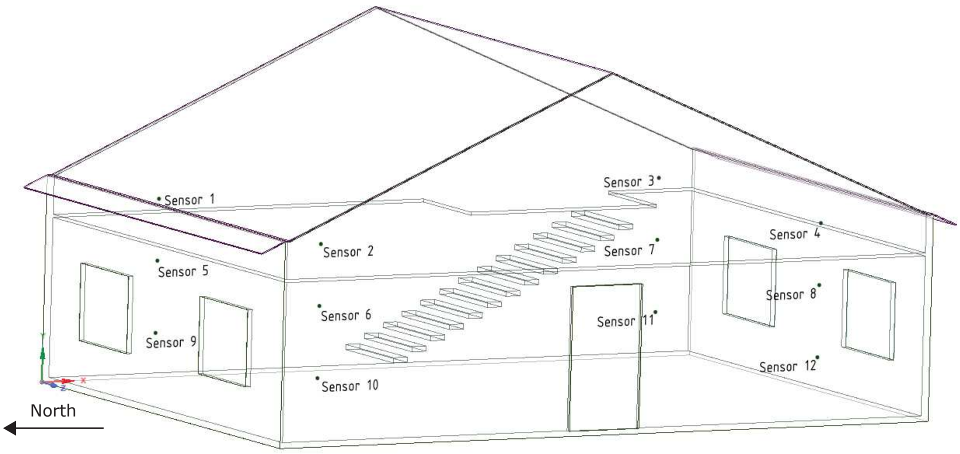


Figure 4.11: Sensors position within the building model.

4.3.3 Co-simulation `Domus-CFX`

There are several co-simulation approaches such as the one-to-one, middleware or a standard interface such the Functional Mock-up Interface (FMI) ([50]). The one-to-one and FMI approaches are implemented in the `Domus` simulation program to enable its coupling with other models and tools. For the co-simulation between `Domus` and `CFX`, the one-to-one coupling based on scripts is used. More information about co-simulation in Domus can be obtained in [49, 61]. `Domus` has been chosen as the BPS tool due to the full understanding and access to the source code by the authors. Moreover, currently, only `Domus` has the pixel counting technique available for insulation calculation.

Previously described in [61], Figure 4.12 illustrates the communication between `Domus` and `CFX`, where `Domus` is the master of the simulation. At the initial stage, the building envelope thermophys-

Table 4.3: Positions of the sensors with respect to the coordinate system of the building.

Sensor	Position		
	X [mm]	Y [mm]	Z [mm]
1	1600	3700	1600
2	1600	3700	8400
3	12400	3700	1600
4	12400	3700	8400
5	1600	2500	1600
6	1600	2500	8400
7	12400	2500	1600
8	12400	2500	8400
9	12400	1100	8400
10	12400	1100	1600
11	1600	1100	8400
12	1600	1100	1600

ical properties (ρ, c, λ) and the boundary conditions are informed to `CFX`. The latter computes a steady-state simulation step as initial conditions.

The boundary conditions for the 3-D building model consist of 38 input variables, provided by `Domus`. The common information used in the entire building are the external weather temperature (T_e), the outside convective heat transfer coefficients (h_e) and the external and internal heat fluxes (q_e,q_i). An additional information presented in this work for boundary conditions is the sunlit for all the internal and external surfaces. At each time step, `Domus` provides the outside and inside boundary conditions (T_e,q_e,q_i,h_e,A_{sunlit}) for each surface to `CFX`.

The `CFX` simulates the whole 3-D building model, considering the natural convection in the air domain and the 3-D diffusive heat transfer in all solid parts. After a successful run, `CFX` returns to `Domus` the spatially averaged surface temperature ($\bar{T}_s$) of all building internal surfaces that have been considered in the Domus-CFX coupling.

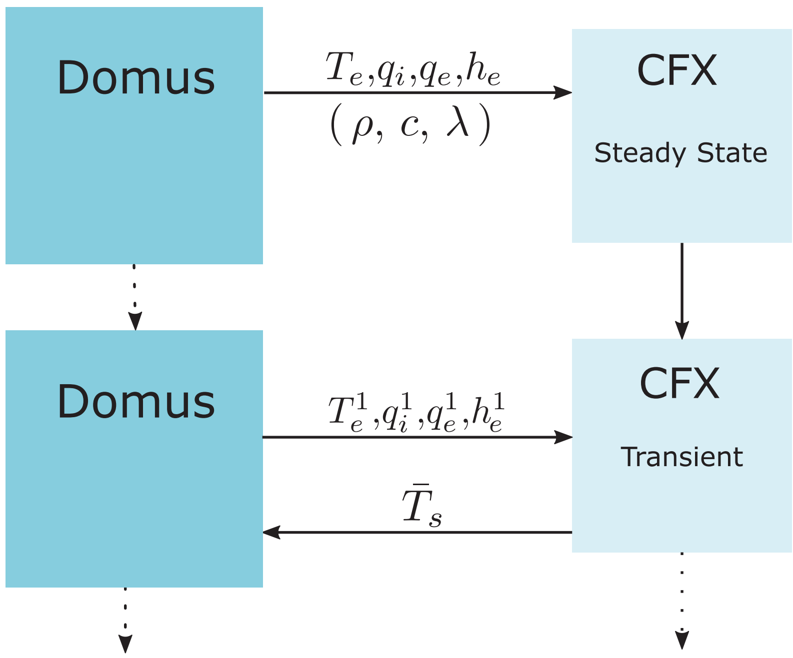


Figure 4.12: Co-simulation scheme between `Domus` and `CFX`.

The CFD coupled model receives non-uniform boundary conditions during the run time period of simulation. This coupling requires the pre-definition of a template of the target geometry, as well as the setup of the mesh by the user. The information transmitted to the CFD tool, such as air temperature, incident radiation flux and convective heat transfer coefficients, are from the previous time-step simulation.

Intelligent co-simulation approach

The intelligent co-simulation approach was first proposed in [61], and later improved versions were detailed in [91], considering `Domus`, as a building performance simulation tool, and `CFX` as a CFD specialized software. As described in [61, 91] there is a significant computational effort involved in the co-simulation approach. Therefore, the idea of intelligent co-simulation is to have a *machine learning model* that can stop the communication between the two programs and replace one of them. The intelligent co-simulation is composed of two phases: i) the training phase and ii) the prediction phase. Thus, as soon the prediction model is trained, `CFX` is disconnected from the co-simulation and `Domus` performs the rest of the simulation using the prediction model, being much faster to simulate with a low impact on the accuracy of results. An important improvement in the *intelligent co-simulation* technique was presented in [91], based on the definition of the training climate in order to build a robust and accurate prediction model. In its original version, the *intelligent co-simulation* used the first days of the simulation period to perform the training. In the second *intelligent co-simulation* publication, a new strategy was used to specify periods of training, using a set with the hottest and coldest days of the year. A second version of this training period was presented based on the temperatures and radiation fluxes for the definition of the hottest days, creating a training climate considering the equivalent outside temperature:

$$T_e^{\text{eq}} = T_e + \frac{q_e}{h_e}, \tag{4.21}$$

which is calculated for the whole climate of the case study. The hottest and coldest days are selected, based on a daily-averaged computation. Then, the training climate sequence is built adding the other hottest days according to the order defined in Figure 4.13. The same procedure is adopted for the coldest days. This sequence enables to simulate the problem for the hottest days increase and coldest decrease of temperature, respectively. The equivalent outside temperature is not used as boundary conditions for the training phase, only to select the days to be used in the training phase. The *intelligent co-simulation* with the improved training climate is used in the case study presented in this section and the results are presented in the next section. Based on previous experiences with this technique, a training period of 6 days has been chosen and presented in Figure 4.14.

The training of the prediction model uses two different approaches. The first one with all input variables included in `Domus-CFX` co-simulation to feed the prediction model. This approach tends to bring larger neural networks, with more neurons and consequently slower training and execution. On the other hand, have access to all contour conditions data that influenced the behavior of the building thermal zone. The second approach limits the number of training inputs to 8, a concept used in machine learning that is better to select the input data in smaller but more influential groups. Therefore, when selecting a smaller number of inputs, it is expected to achieve better prediction results than in the first approach. The criterion to select the inputs is the proximity to the sensors, then selecting the boundary conditions that affect the surfaces closest to the target sensor. To enable the previously defined approaches, it was chosen to perform the training of a neural network one-per-target, with the sensors and mass airflow in the region between zones being the targets to be analyzed.

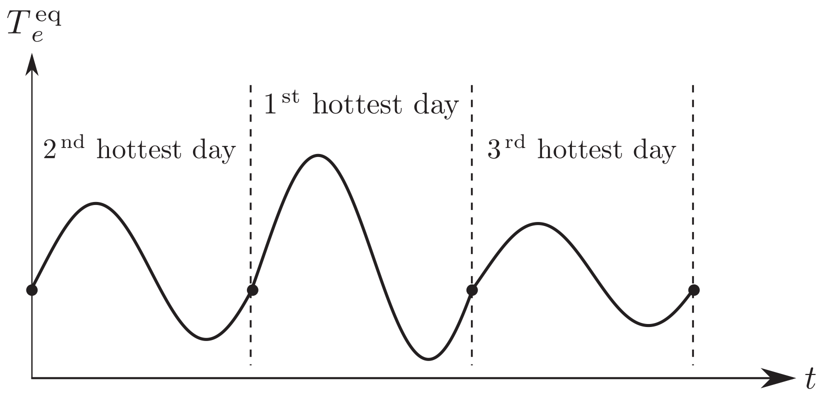


Figure 4.13: Definition of the improved training climate for the hottest days.

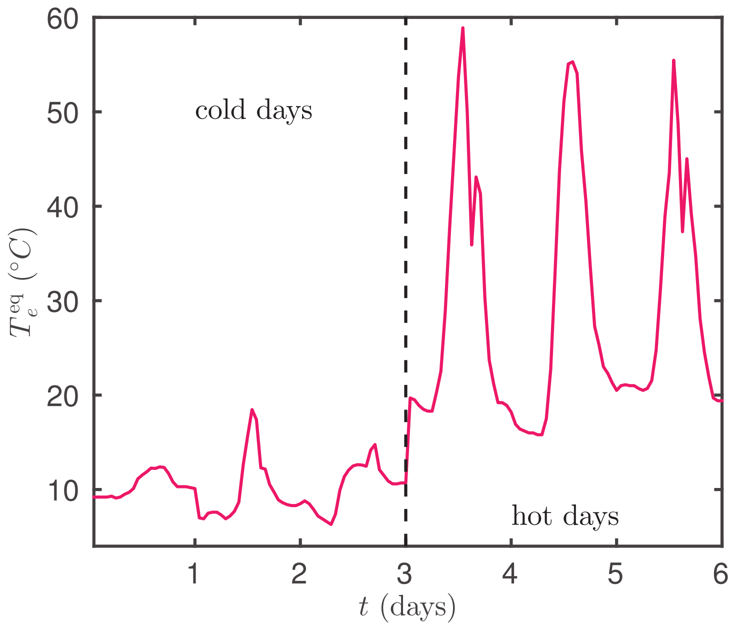


Figure 4.14: Variation of the outside equivalent temperature for a 6-day training period.

4.4 Results

This section is divided into three parts. The first one provides results according to the radiation distribution model presented in Section 4.3.1. Then, in Section 4.4.2 detailed results are presented in terms of three-dimensional profiles of temperature and air velocity, inter-zone airflow and virtual sensors. As the co-simulation run time is prohibitive, Section 4.4.3 shows how the *intelligent co-simulation* approach enables to get accurately most of the details of a 3-D advanced simulation in a short period of time.

4.4.1 Radiation Distribution

The results presented in what follows show the data obtained from the improvements obtained in the internal radiation calculations in `Domus`. Instead of accumulating the radiation only on the floor, as commonly done by BPS tools, the direct radiation is distributed on each wall according to the sunlit area. Figures 4.15 and 4.16 present the results of direct, diffuse, diffuse reflected and direct reflected radiation values on each internal surface of the present case study if they would be used in 1-D simulation.

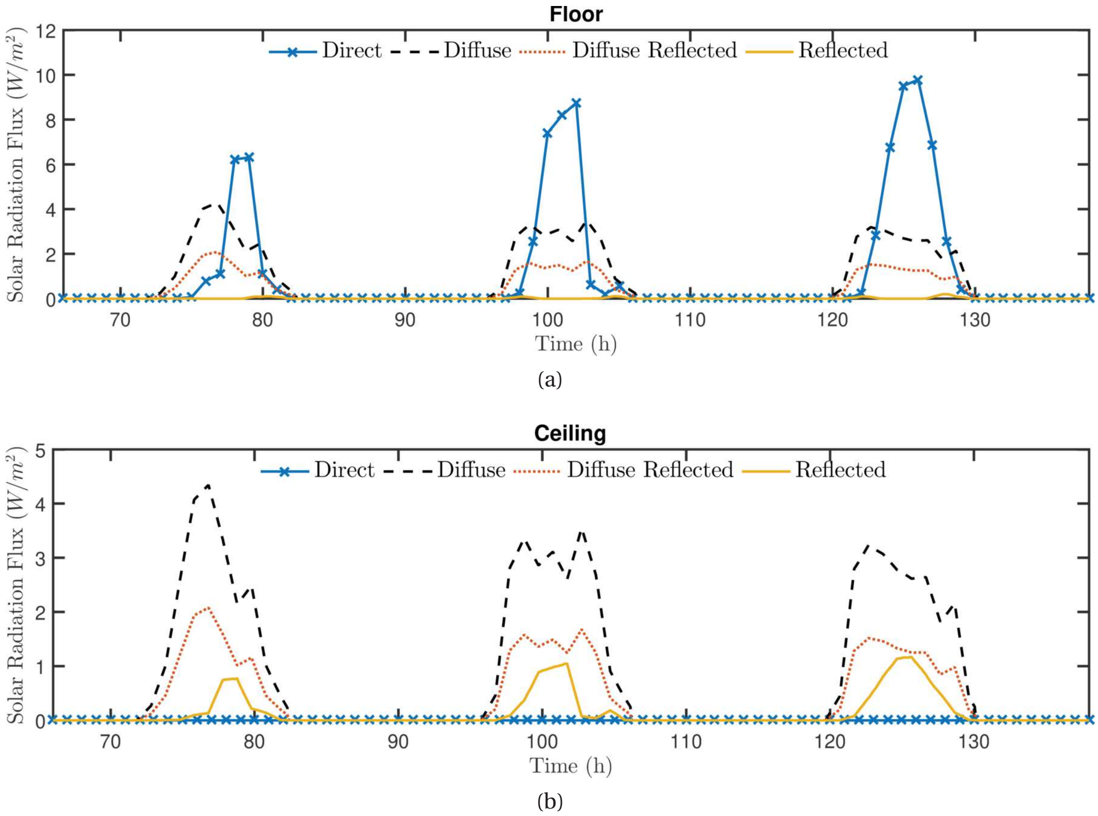
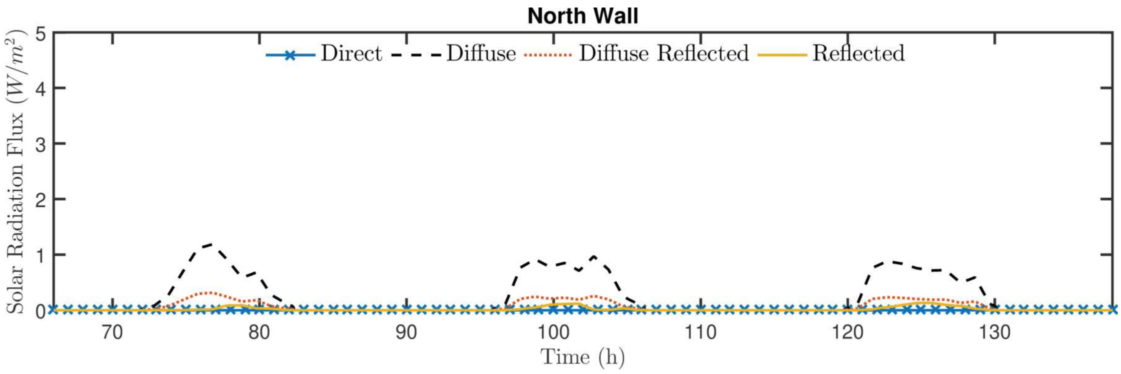


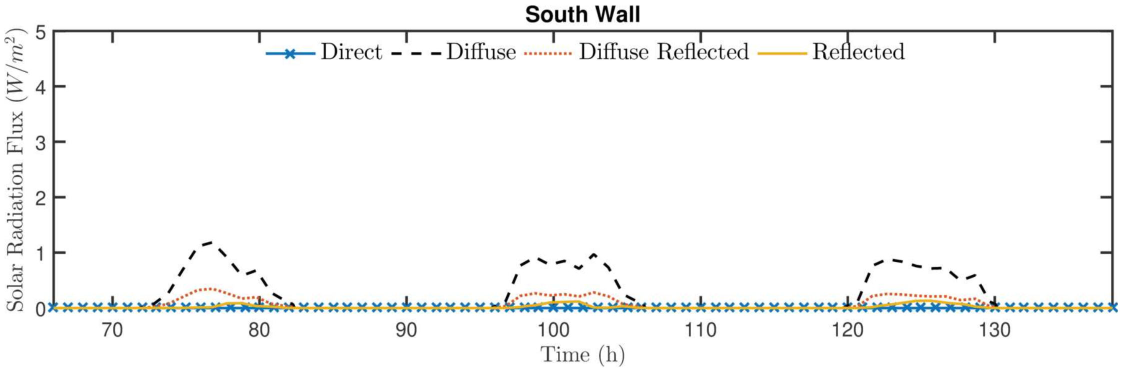
Figure 4.15: Internal solar radiation flux on the floor (a) and on the ceiling (b).

As detailed in Section 4.3.1, the radiation fractions are calculated by `Domus`. The direct solar radiation is obtained from the PxC calculation, that provides the sunlit area on each surface. `Domus` calculates the internal distribution of diffuse and reflected radiations as a function of solar beam vector (unitary) and the view factor among all internal surfaces. Figure 4.17 shows the total of radiation values that occurred within the zone for each type of radiation flux and, as a comparison, Figure 4.18 shows the sum of all the radiation fluxes, which is the value commonly used as radiative boundary condition in classical models by placing them only on the floor.

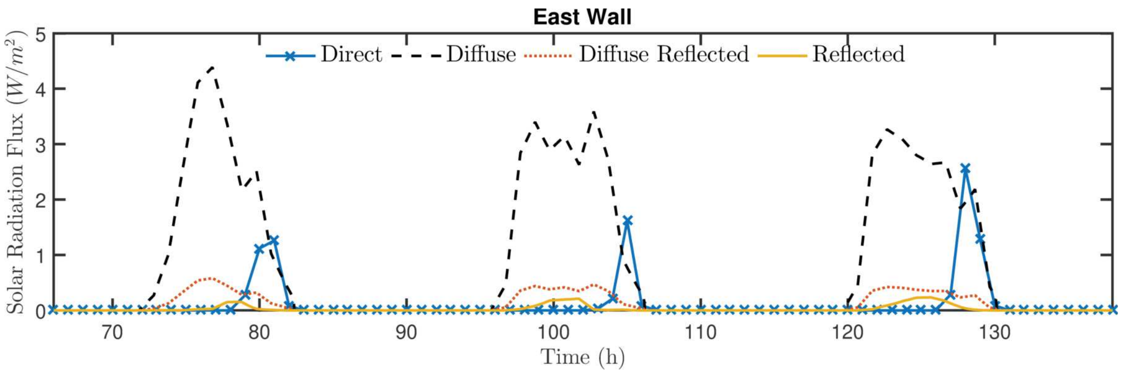
Figures 4.19-4.21 present the radiation values applied as boundary conditions of the 3-D detailed model, whose thermal results are presented in Section 4.4.2. As one can expect, by comparing Figures 4.19-4.21 with Figures 4.15-4.17, high differences in temperature fields can be noticed. This difference is attributed to the fact that all direct incident radiation - for the 3-D model - is distributed only on the area affected by the sunlit, and not uniformly over the whole surface area as done by 1-D models that can account for the solar beam radiation data distributed in different surfaces of the building enclosure. Since in the present case study that has relatively small windows, the sunlit areas are small. As a consequence, the effect of the simplification for 1-D models, distributing the radiation over the whole surfaces, can be considerably remarkable by creating important instantaneous temperature gradients, promoting surface heat fluxes and inducing higher air velocities near the concentrated sunny region. For the co-simulation between `Domus` and `CFX`, the direct radiation values used are shown in Figures 4.19-4.21, which correspond to values applied only on the sunlit areas, thus generating more important 3-D effects. The other radiation fluxes (diffuse, reflected and direct reflected) are distributed over the entire surface, in the same way as occurs in the classic `Domus` model.



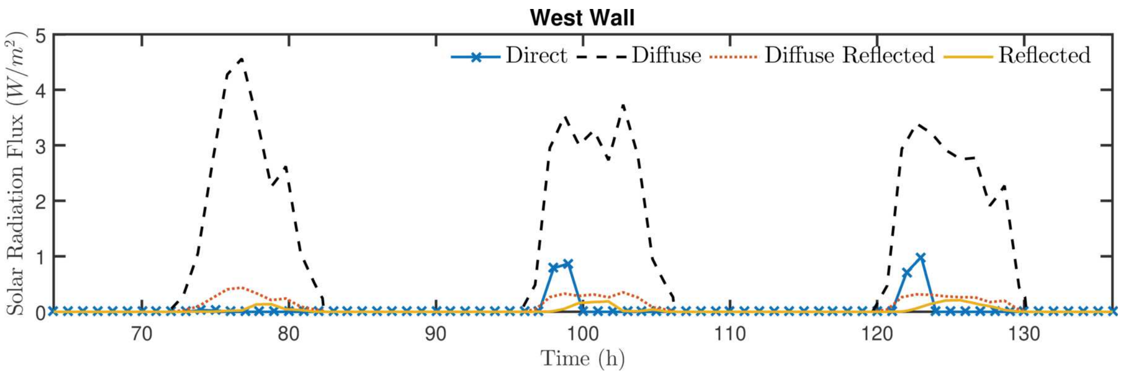
(a)



(b)



(c)



(d)

Figure 4.16: Internal solar radiation flux on different walls.

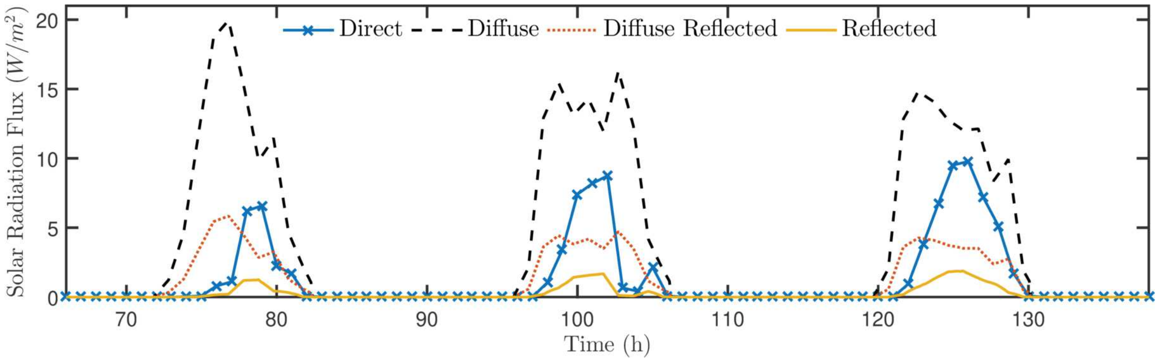


Figure 4.17: Total internal solar radiation flux.

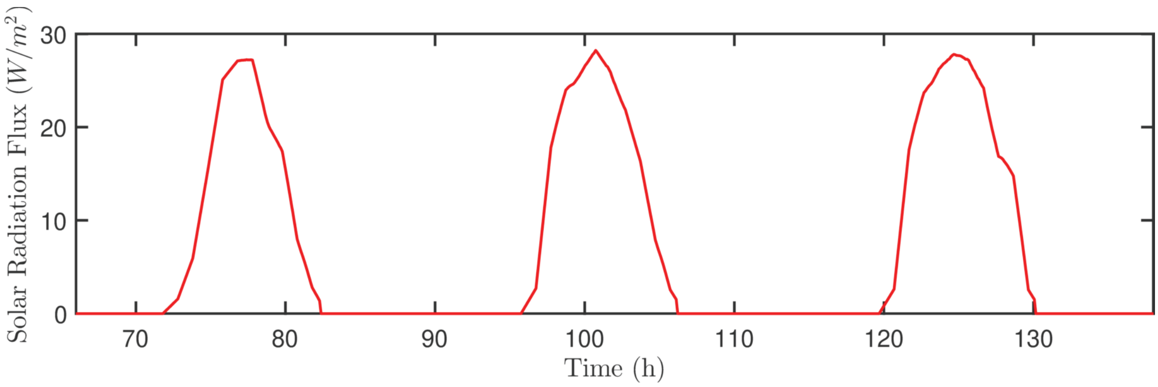


Figure 4.18: Accumulated solar radiation fluxes on all surfaces.

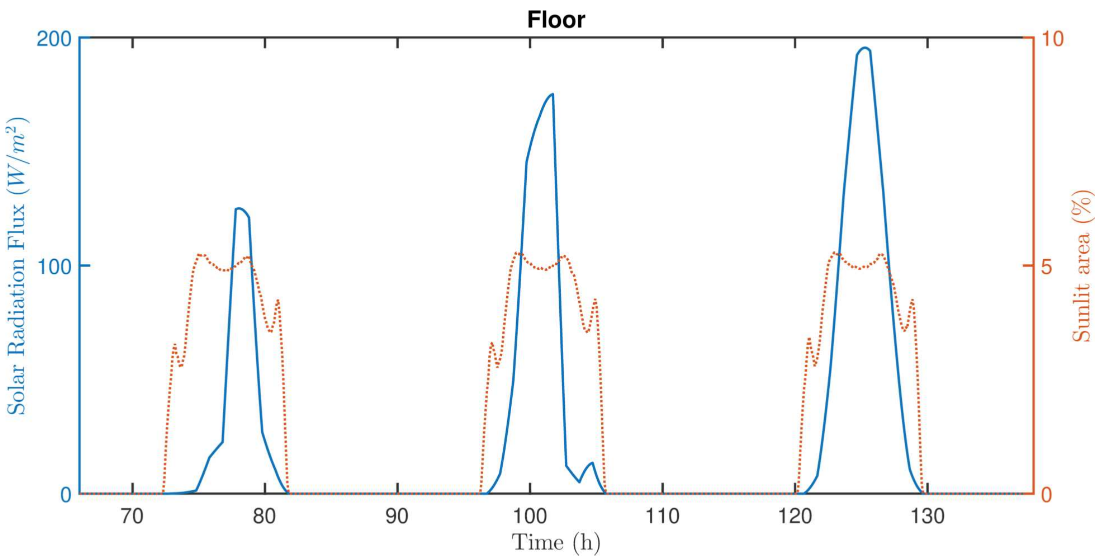


Figure 4.19: Solar radiation flux on the floor sunlit area.

4.4.2 Detailed Building Simulation

In order to explore the air temperature and velocity behaviors within the building space, major advantage of the physical simulation in three dimensions, the virtual no-mass sensors are the most detailed items in this section. Aiming to provide better visualization, the results of the sensors are grouped with respect to both Z and X coordinates, with each figure having sensors that only vary according to the vertical Y coordinate. Figure 4.22a shows the average air temperature regarding

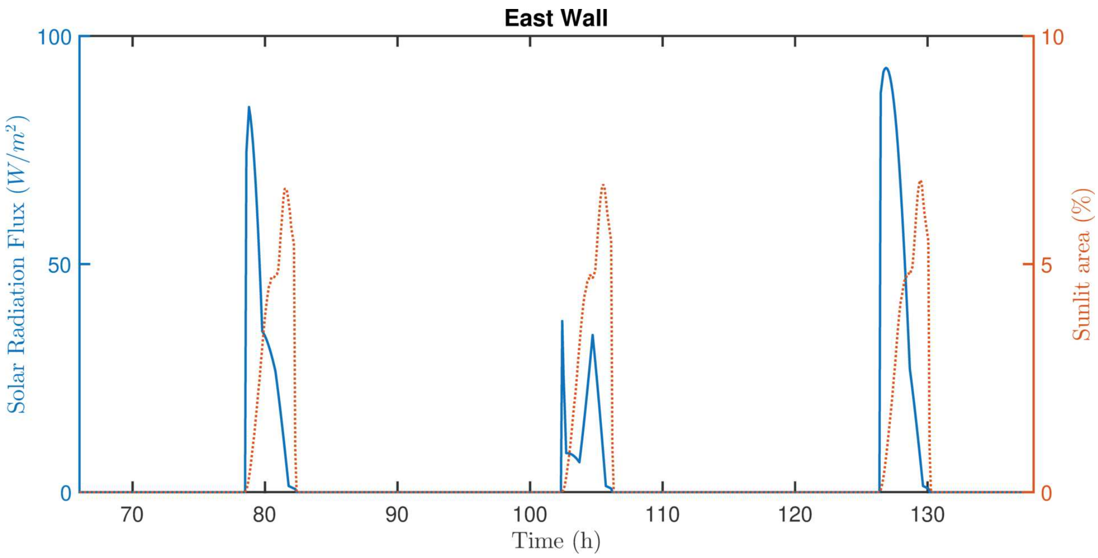


Figure 4.20: Solar radiation flux on east wall sunlit area.

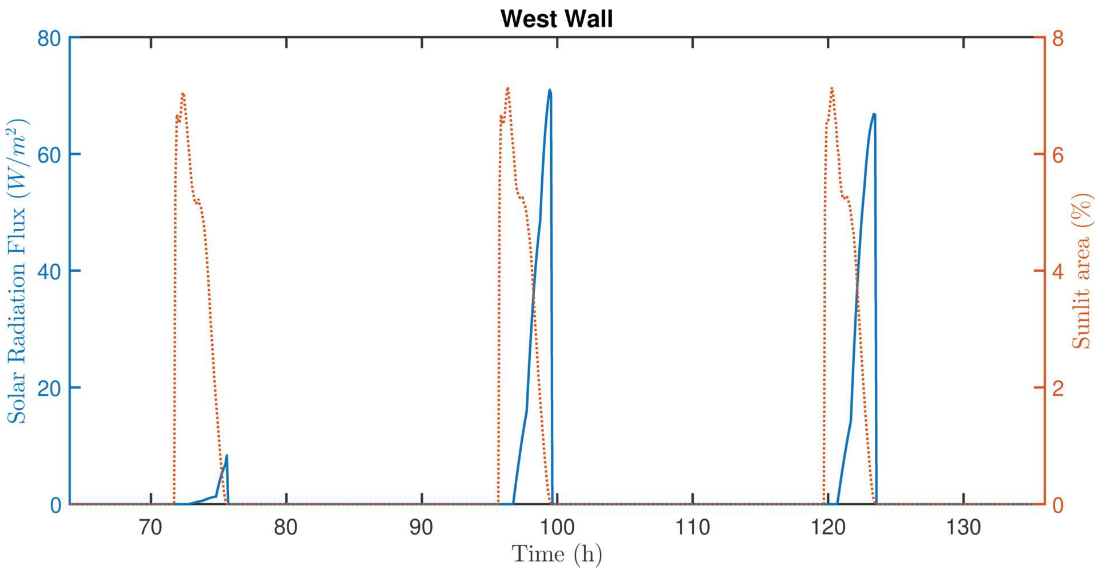


Figure 4.21: Solar radiation flux on west wall sunlit area.

sensors 1, 5 and 9, while Figure 4.22b shows the average air temperature for sensors 2, 6 and 10 on 21st and 22nd of June. The same can be observed in Figures 4.37 and 4.23b on 21st and 22nd of December.

The temperature results on sensors 3, 7 and 11 and sensors 4, 8 and 12 are shown in Figures 4.24a and 4.24b for the period of June 21-22, while in Figures 4.39 and 4.25b for the period of December 21-22. The solutions presented a maximum difference of 9.8°C between sensors 2 and 12 at around 13 pm of the day 20 of June. The mean value and the standard deviation among all sensors are presented in Figure 4.26, where it is possible to observe considerable differences between the results. Based on the overall results, it is possible to verify greater differences in temperatures occurring near the roof, especially in the hottest periods of the day, under the effect of a higher incidence of solar radiation on the roof surface.

Following the same logic of sensors grouping, Figures 4.27a and 4.27b present the averaged velocity obtained for sensors 1, 5 and 9 and 2, 6 and 10 for the winter period and Figures 4.28a and

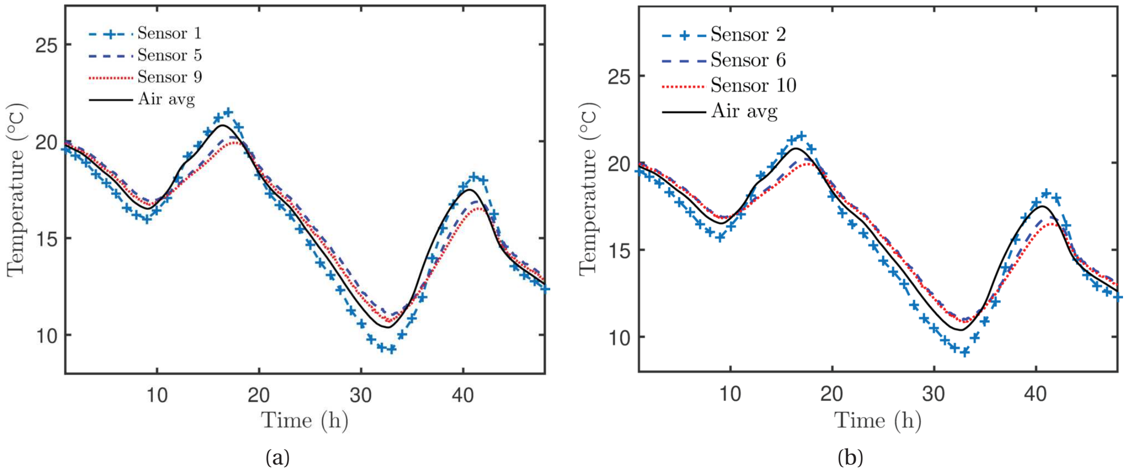


Figure 4.22: Averaged air temperature with respect to the values obtained for sensors 1, 5 and 9 (a) and sensors 2, 6 and 10 (b), on 21st and 22nd of June.

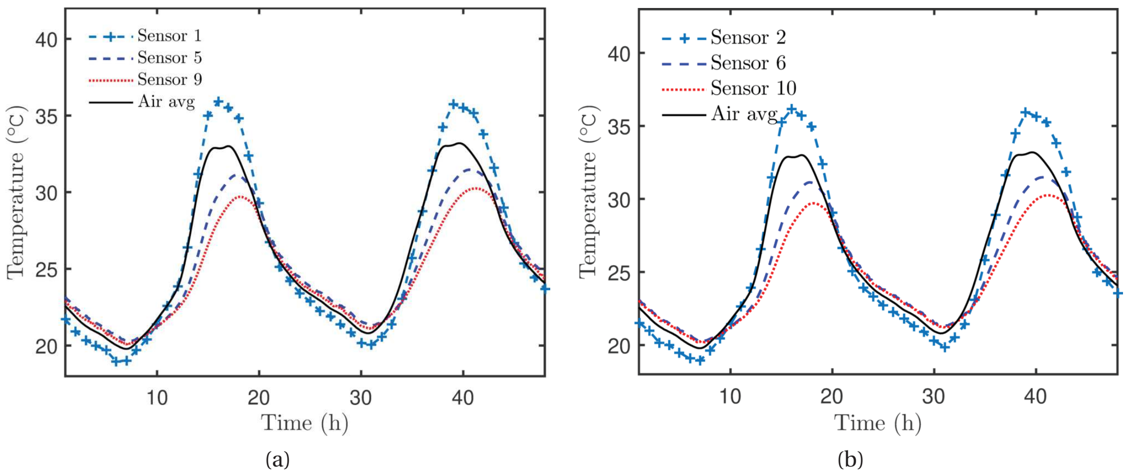


Figure 4.23: Averaged air temperature with respect to the values obtained by virtual sensors 1, 5 and 9 (a) and virtual sensors 2, 6 and 10 (b), 21st and 22nd December.

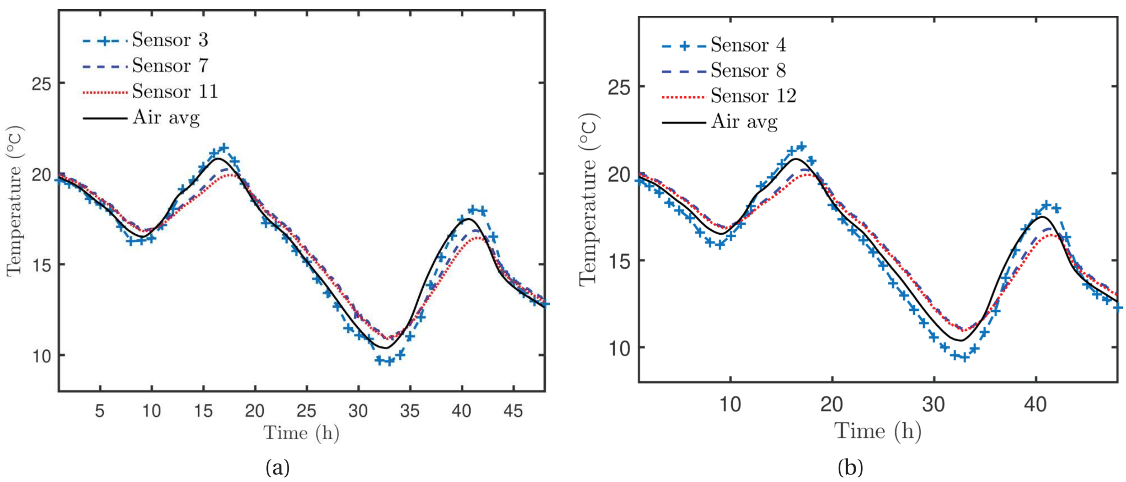


Figure 4.24: Averaged air temperature with respect to the values for sensors 3, 7 and 11 (a) and sensors 4, 8 and 12 (b), on 21st and 22nd of June.

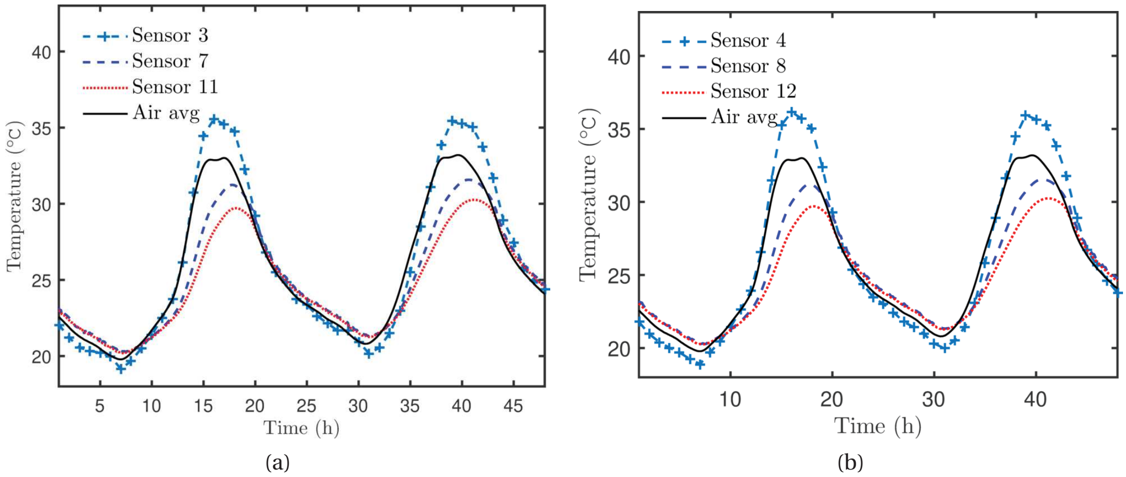


Figure 4.25: Averaged air temperature with respect to the values for sensors 3, 7 and 11 (a) and sensors 4, 8 and 12 (b), on 21st and 22nd of December.

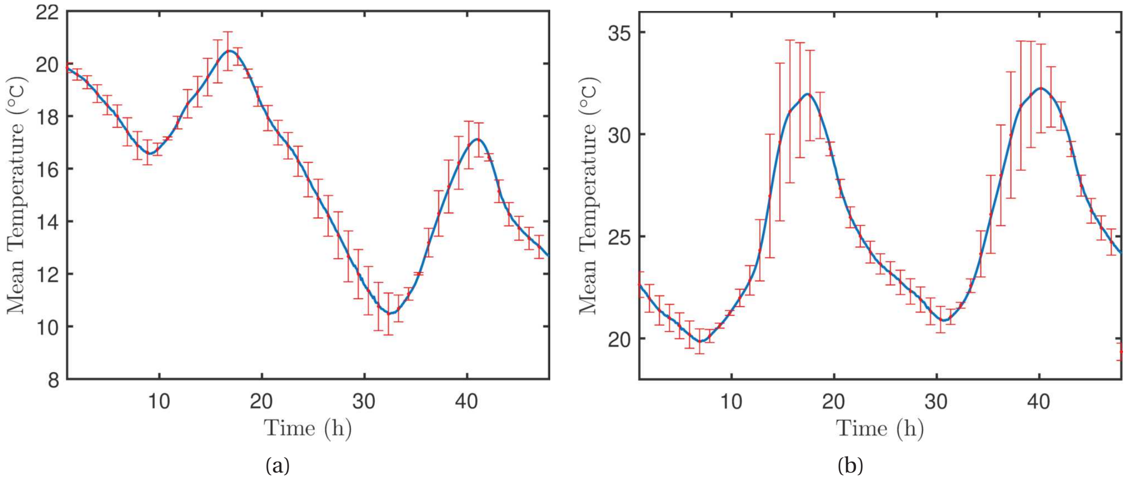


Figure 4.26: Mean sensor temperature and standard deviation values, on 21 and 22 of June(a) and on 21st and 22nd of December(b).

4.28b for the summer days.

The velocity results obtained on sensors 3, 7 and 11 as well as on sensors 4, 8 and 12 are shown in Figures 4.29a and 4.29b for June 21-22 and, in Figures 4.30a and 4.30b for December 21-22. It is possible to observe that higher velocity values occur also closer to the roof due to the high temperature in the region. Furthermore, a considerable difference can be observed on the Z-axis between sensors 1 and 2 in comparison to sensors 3 and 4, which is mainly attributed to the effect of air circulation induced by the ladder opening.

Normally, a simulation user would consider no interactions between the two zones and would simulate as one-zone building, which can be a good hypothesis, especially due to the difficulties on modeling the interactions between thermal zones connected by an opening as the one shown for the ladder way. However, in some cases, it is essential to compute the heat and moisture transfer process through the opening for assessing energy performance. In this way, to better illustrate the potential of the present approach, this chapter presents, in Figure 4.31, the airflow computed by the CFD tool, as well as the conduction and convection loads between the two zones (Figure 4.32). The only way to more generically predict the airflow is by means of CFD codes, which are normally not done due

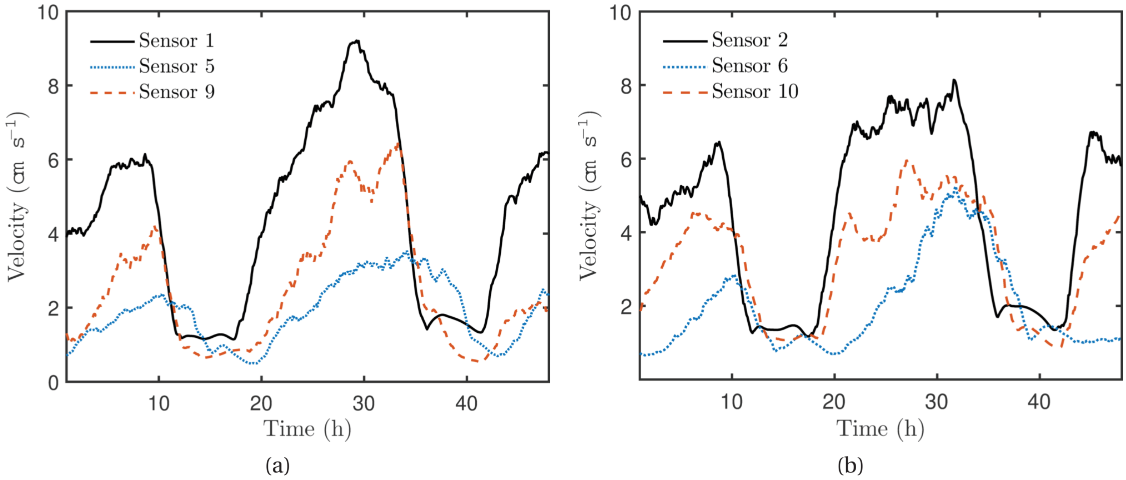


Figure 4.27: Mean air velocity at the position of sensors 1, 5 and 9 (a) and sensors 2, 6 and 10 (b), on 21st and 22nd of June.

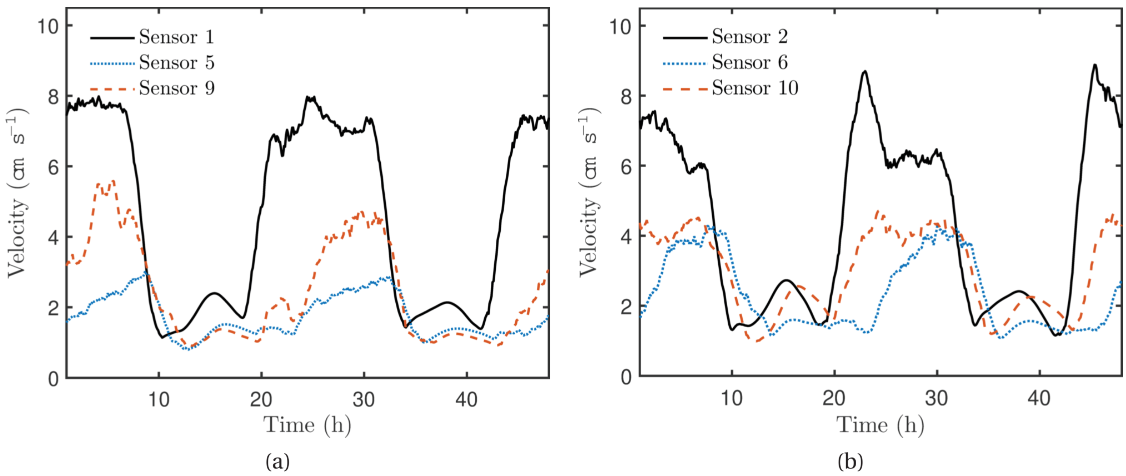
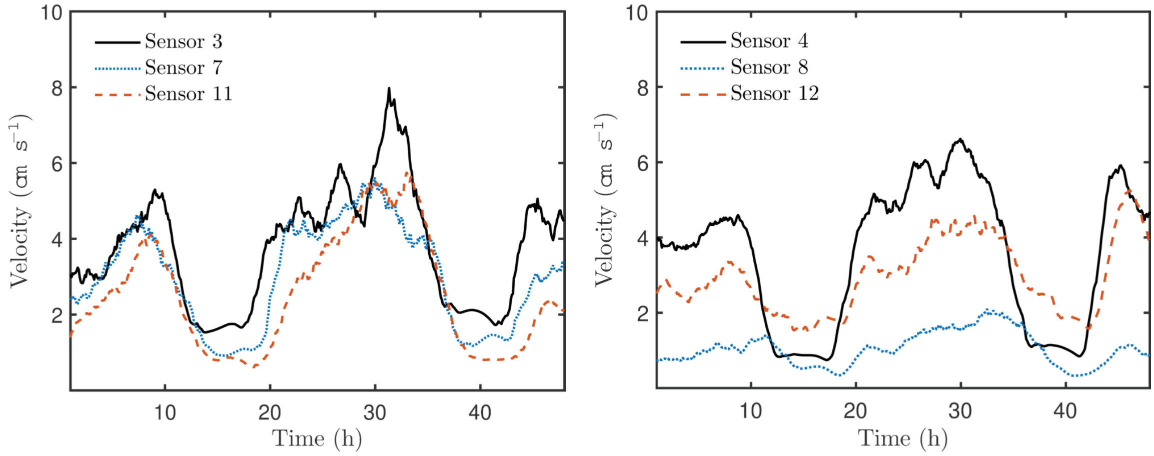


Figure 4.28: Mean air velocity at the position of sensors 1, 5 and 9 (a) and sensors 2, 6 and 10 (b), on 21st and 22nd of December.

to the prohibitive simulation cost. As one can observe in Figure 4.31, the airflow varies significantly along the day and a constant value, which is normally used by BPS tools, makes absolutely no sense. The magnitude of this airflow could be greatly magnified if a heater or a fan had been considered. It is also possible to observe considerable temperature differences within the building space at specific hours, as observed in Figure 4.34c, where a difference of approximately 23°C can be noticed in the indoor space.

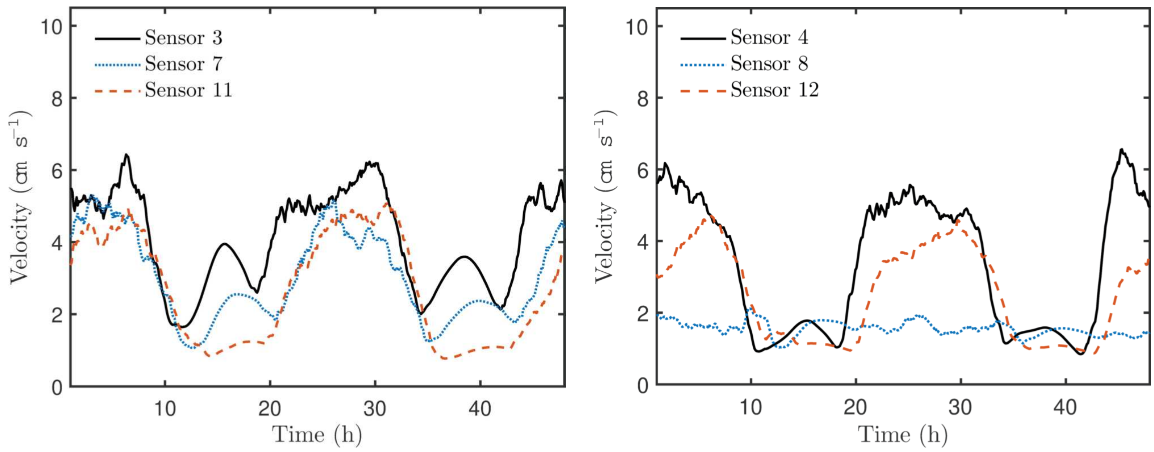
In Figure 4.33, it is possible to observe the effect of sunlit over the velocity field, showing higher velocities close to the regions where the beam hits the surface. The zoom illustrated by Figure 4.33b provides additional evidence about the importance of a detailed radiation model showing the spot where the solar beam hits the wall and causes a temperature rise. In Figure 4.33b, we can also see details on the mesh generated during the process of locating the sunny face elements.

Figures 4.34 and 4.35 additionally illustrate the important multidimensional effect caused by the incidence of solar beam on both floor and east wall surfaces, inducing a considerable influence on the buoyancy-driven flows within the room, modeled according to the Boussinesq approximation. These figures present a view of the floor, north and east wall surfaces between 13:30h and 17h on June 21st.



(a) (b)

Figure 4.29: Mean air velocity at the position of sensors 3, 7 and 11 (a) and by virtual sensors 4, 8 and 12 (b), on 21st and 22nd of June.

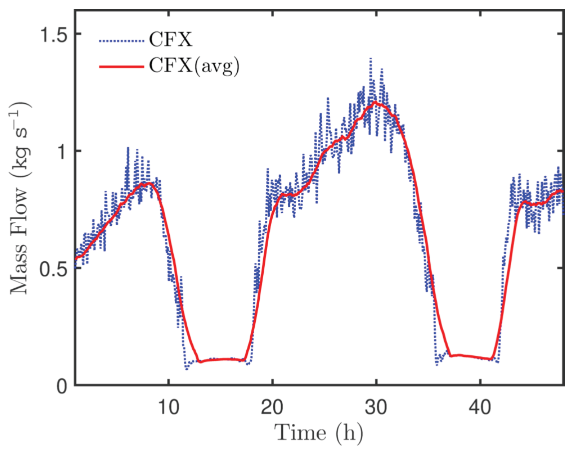


(a) (b)

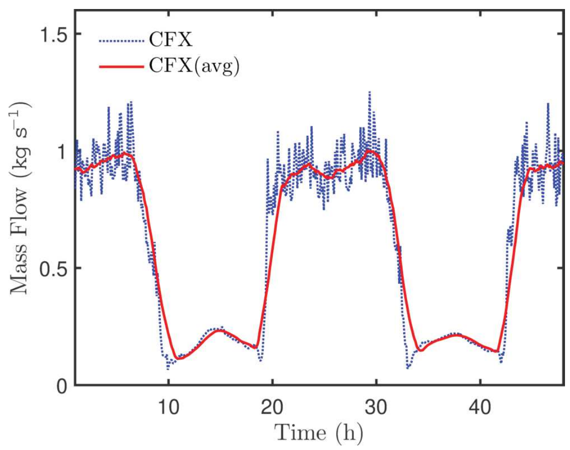
Figure 4.30: Mean air velocity at the position of sensors 3, 7 and 11 (a) and sensors 4, 8 and 12 (b), on 21st and 22nd of December.

In those figures, on the left side, the sunlit incident area on the surfaces is shown, in orange color, as well as the description of the amount of solar radiation that hits those areas (upper left corner of each figure). In Figures 4.34e and 4.34g, for example, it is noted the presence of small spaces within the sunlit area, which are caused by the leaves of the tree used as a shading element. It is important to remark that the accuracy of the sunlit area definition is directly associated to the mesh refinement. In the right column of Figures 4.34 and 4.35, the temperature field on the surfaces is shown. It is possible to observe clearly the increase of the temperature in the regions directly affected by the solar beam, showing, in some cases, temperature differences greater than 5°C. The unsteady solar-spot effect, moving over the surfaces can also be visualized by the transient temperature distribution over the surface.

The entire 14-day simulation period took approximately 1200 hours of computation (50 days). An average time of 18 minutes per time step is consumed by `CFX`, while 30 seconds by the other elements involved in the coupling process such as the generation of the sunlit data that consumes itself about 20 seconds per time step. Obviously, the sunlit data is only generated when direct radiation is not null. Another important detail is that the procedure of locating the sunlit on surfaces with solar ra-

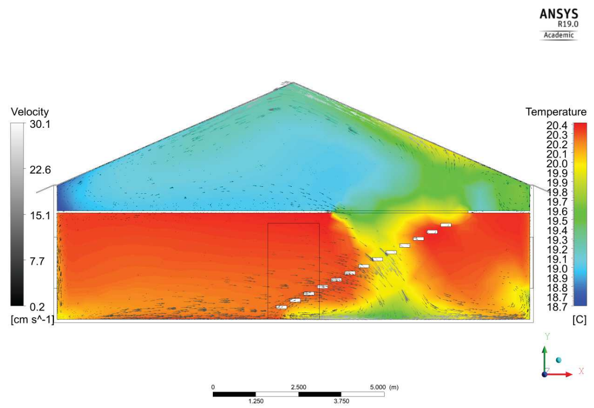


(a)

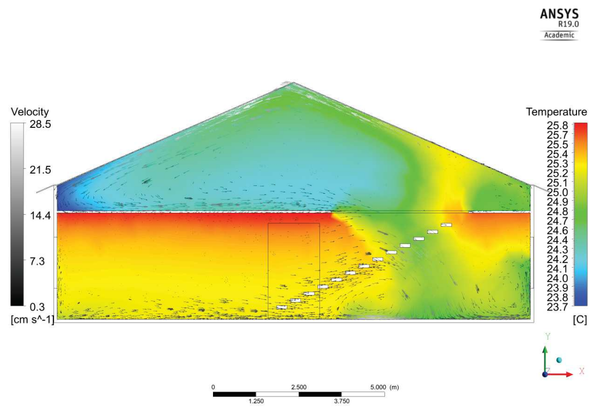


(b)

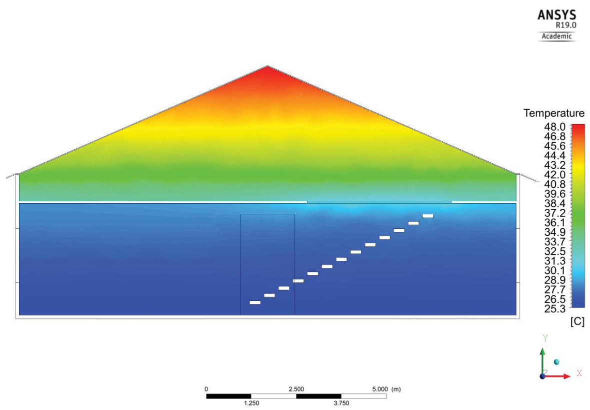
Figure 4.31: Mass airflow at the opening, on 21st and 22nd of June(a) and on 21st and 22nd of December(b)



(a) t = 6 h



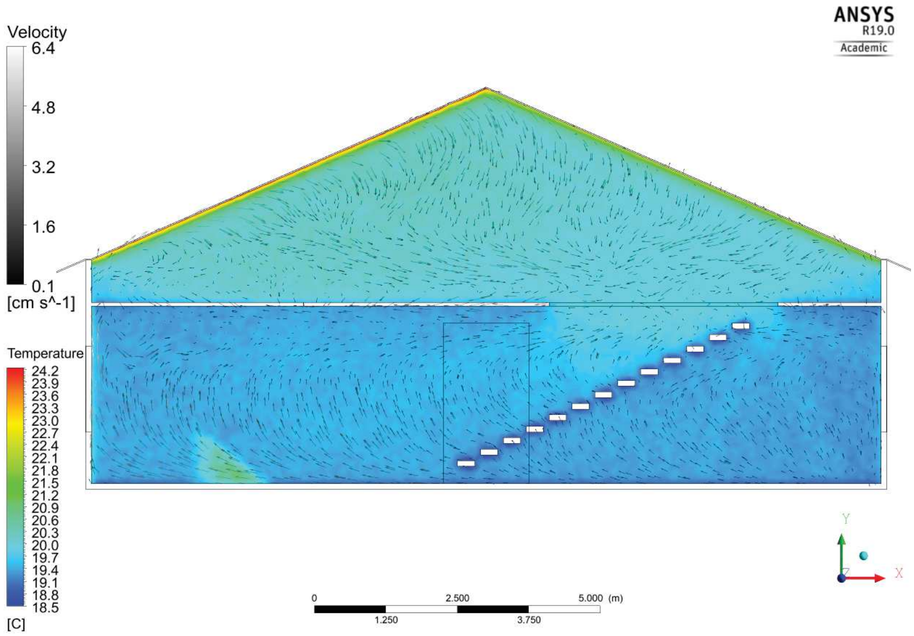
(b) t = 22 h



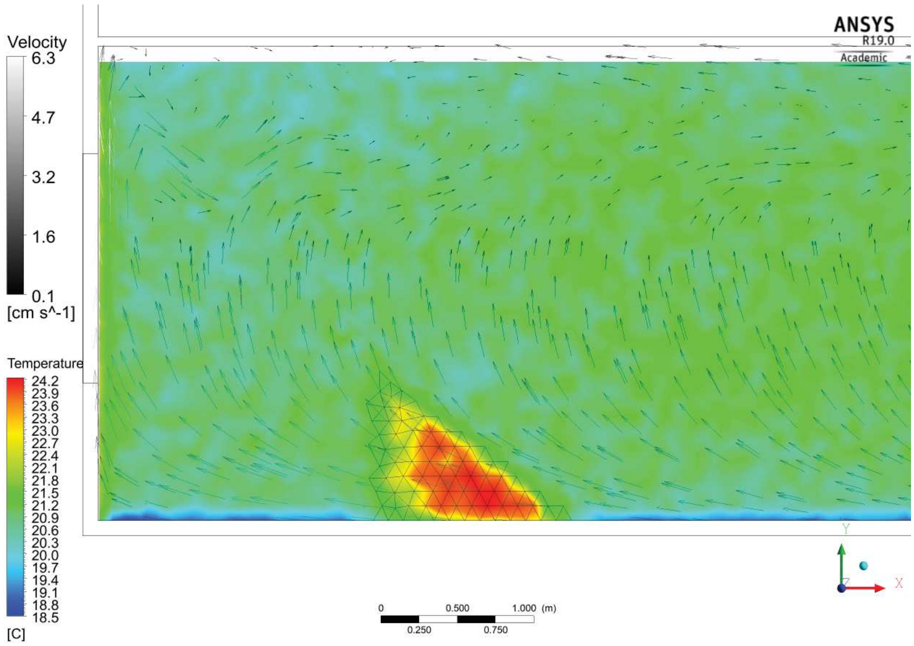
(c) t = 14 h

Figure 4.32: Contour plots of the air local range temperature, overlaid by the velocity vector field, calculated at a vertical plane 1 m from the west wall, on December 21st.

diation incidence close to 100% of the surface area is not done; in this case, all the direct radiation is distributed over the entire surface. There are still possibilities to optimize the sunlit generation procedure, using faster tools for mesh editing and sunny face elements definition or even using `SALOME` tool as a service associated with the BPS tool, without having to initialize and execute the tool every time



(a)



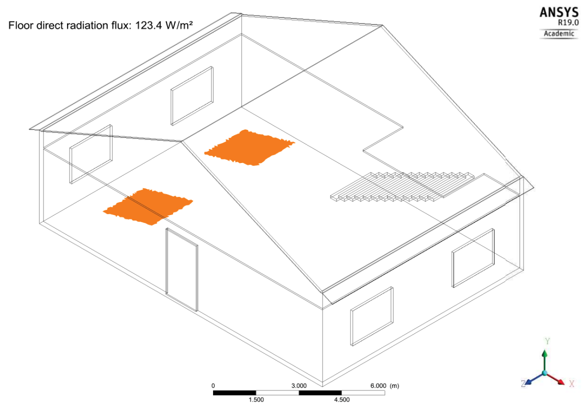
(b)

Figure 4.33: Contour plots of the east wall local range temperature, at 14h30 of the day 21 of June, overlaid by the velocity vector field calculated at a vertical plane 1 cm from the east wall (a), zoom in the left corner showing the sunlit mesh (b).

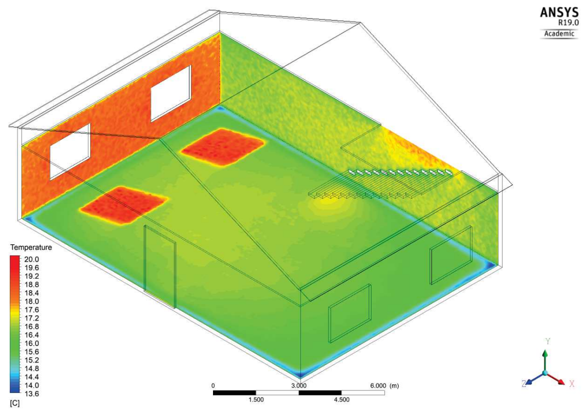
step, which can save considerable time in long-period simulations.

4.4.3 Intelligent Co-Simulation

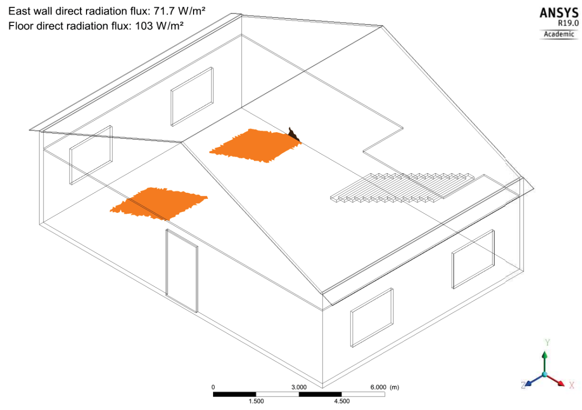
The generation of prediction models has been performed for sensors 1 and 7. The objective was to capture information from different positions within the building and thus to cover the different situations provoked by the multidimensional effects of heat transfer and airflow on the values of tem-



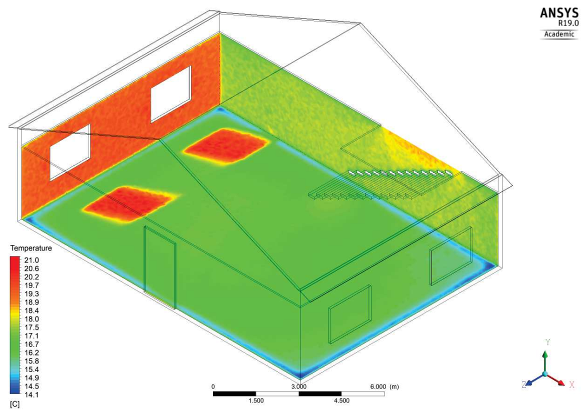
(a) t = 13 : 30 h



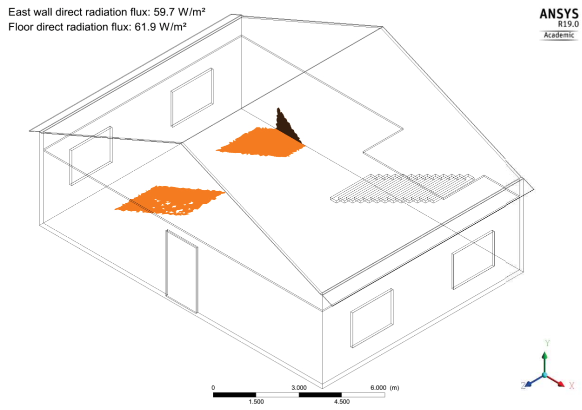
(b) t = 13 : 30 h



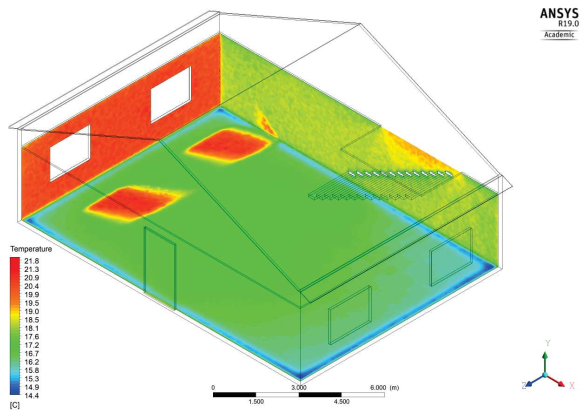
(c) t = 14 h



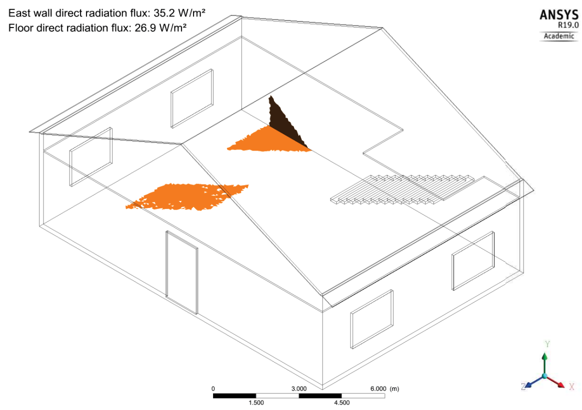
(d) t = 14 h



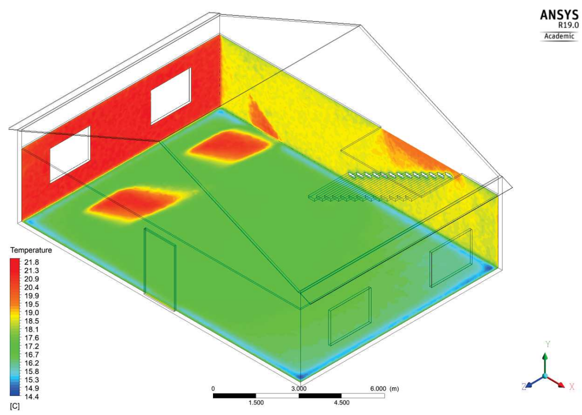
(e) t = 14 : 30 h



(f) t = 14 : 30 h

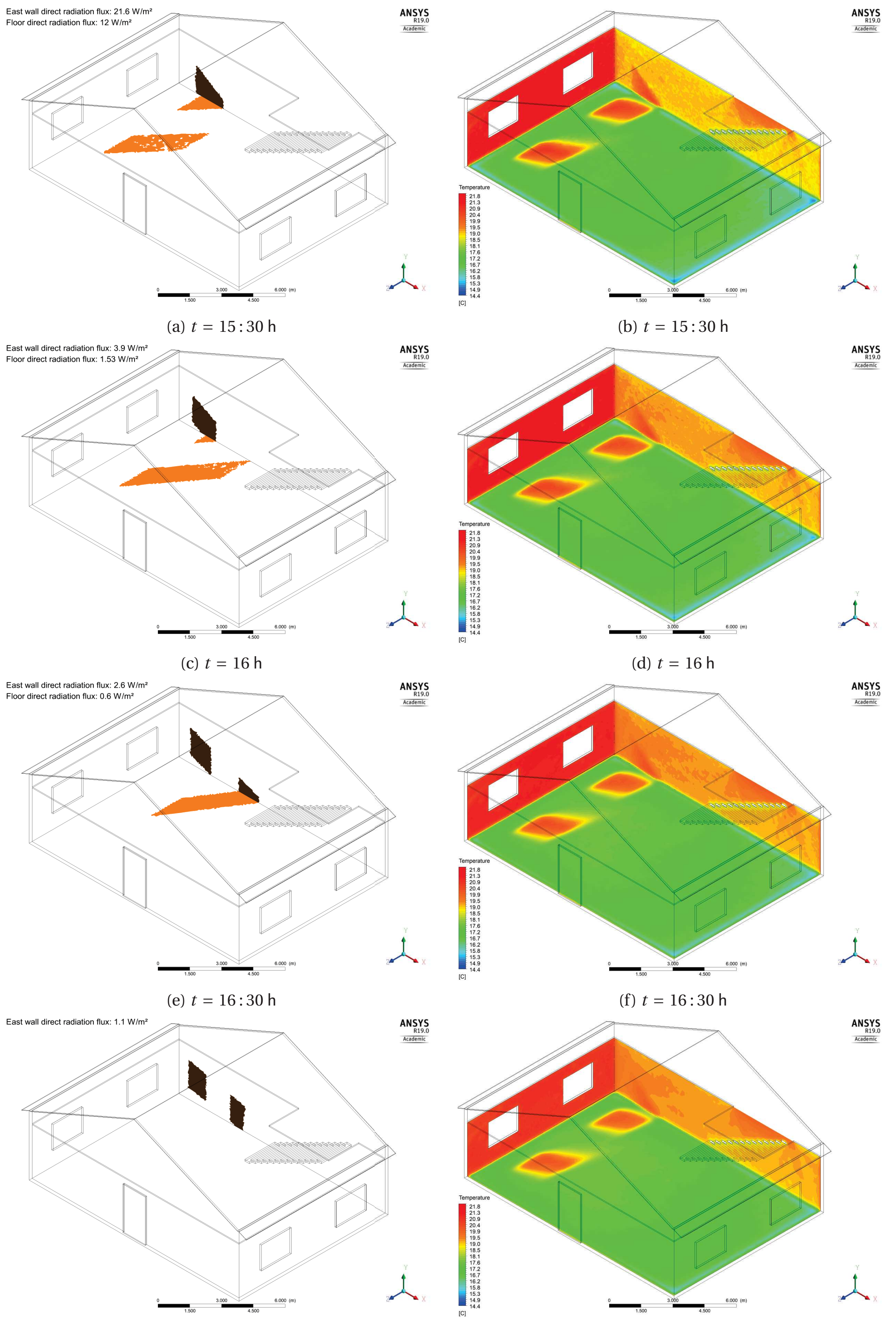


(g) t = 15 h



(h) t = 15 h

Figure 4.34: Contour plots of the surface local range temperature, 13:30h to 15h on June 21st.



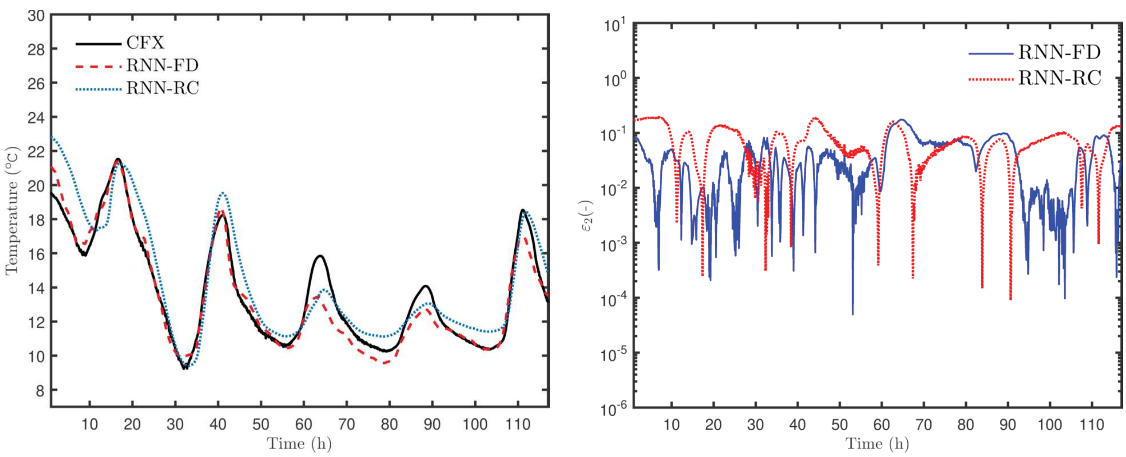
(a) t = 15 : 30 h
(b) t = 15 : 30 h
(c) t = 16 h
(d) t = 16 h
(e) t = 16 : 30 h
(f) t = 16 : 30 h
(g) t = 17 h
(h) t = 17 h

Figure 4.35: Contour plots of the surface local range temperature, 15:30h to 17h on June 21st.

perature and air velocity.

The results presented in this section comes out from three different approaches: i) `Domus-CFX` co-simulation as the reference result; ii) RNN prediction model trained by using the first 2-days (RNN-FD) obtained from the co-simulation data, as adopted in Chapter 2; iii) RNN prediction model trained by using a representative climate (RNN-RC) as described in Section 4.3.3. This section actually aims at comparing the predictive capability of the two RNN-based models for periods not used in the training process and at estimating the possibility to use those prediction models in one-year-long simulation of a building as complex as the one presented in Section 4.3.2.

Figure 4.36a shows the temperature evolution captured by Sensor 1 in the winter period (June, 21-25). It is possible to observe the RNN-FD model shows smaller errors in the first two-day period, exactly the same used for model training. Regarding the following days (3-5), Figure 4.36b presents the RMS errors, showing the RNN-FD presents higher values than those obtained by the RNN-RC model. This behavior is expected since the RNN-RC model was trained with values that cover a larger spectrum among the situations that may occur in the climate chosen. Consequently, the RNN-RC model tends to be able to predict more accurately a full-year simulation. The prediction results and RMS errors for the summer period (December, 21-25) are presented in Figures 4.37a and 4.37b, showing a similar behaviour than the one presented for the winter period, i.e., better results appear for the RNN-FD only for the first two days.

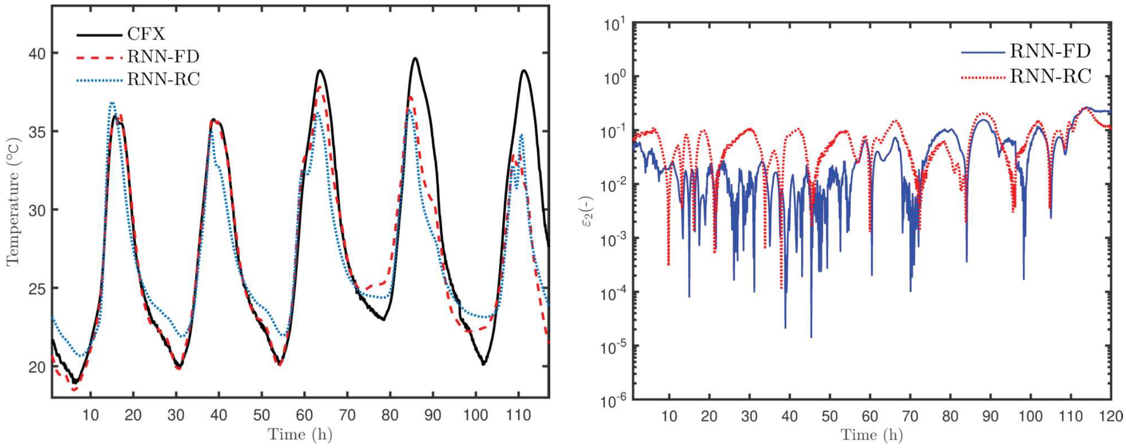


(a) (b)

Figure 4.36: Air temperature and RNN prediction for sensor 1 (a) and temperature prediction RMS error (b), in the winter period.

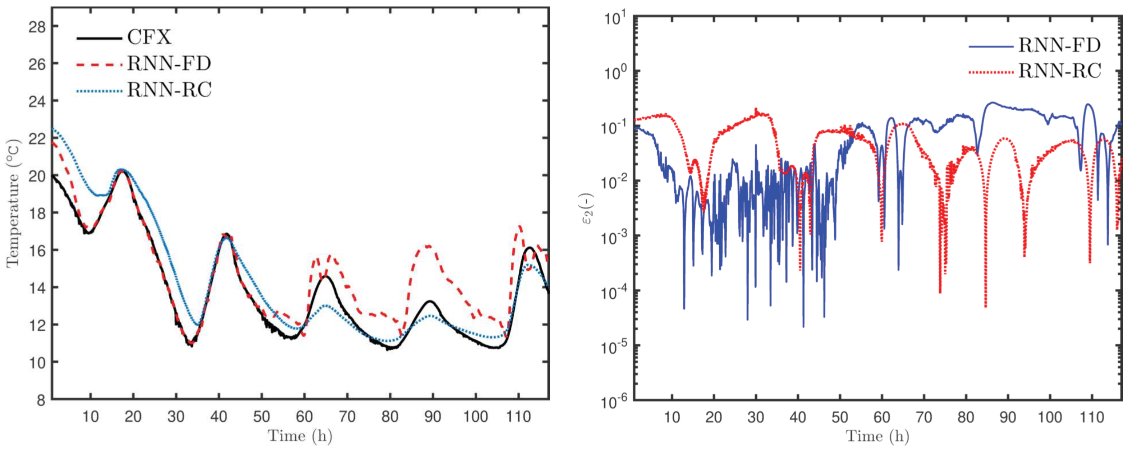
Figures 4.38 and 4.39 present the same as Figure 4.36, but for sensor 7. Comparing with the prediction results of sensor 1, one can observe that the RNN-FD results show a larger error out the period used for training, and a higher temperature drop than that occurred in sensor 1 can be noticed. For RNN-RC, the error remains in the order $\mathcal{O}(10^{-1})$ because it has included lower temperatures for the training process.

Following the same conditions used in temperature prediction, Figures 4.42 and 4.43 present the prediction of air velocity relatively to the position of sensors 1 and 7, using the same three approaches. For this physical variable, the prediction is more complex, showing higher errors for both prediction approaches, when compared to temperature prediction errors. The training process used the instantaneous velocity variation as a target, which may present abrupt variations during the simulation. Even so, the results show the prediction model can estimate trends of air velocity variation along the day, with very satisfactory approximations at certain moments.



(a) (b)

Figure 4.37: Air temperature and RNN prediction for sensor 1 (a) and temperature prediction RMS error (b), in the summer period.

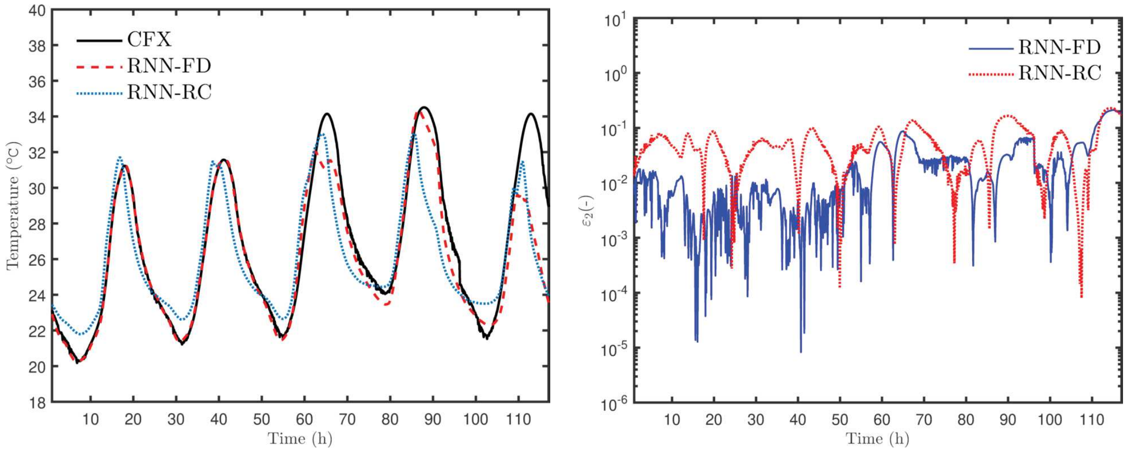


(a) (b)

Figure 4.38: Air temperature and RNN prediction for sensor 7 (a) and temperature prediction RMS error (b), in the winter period.

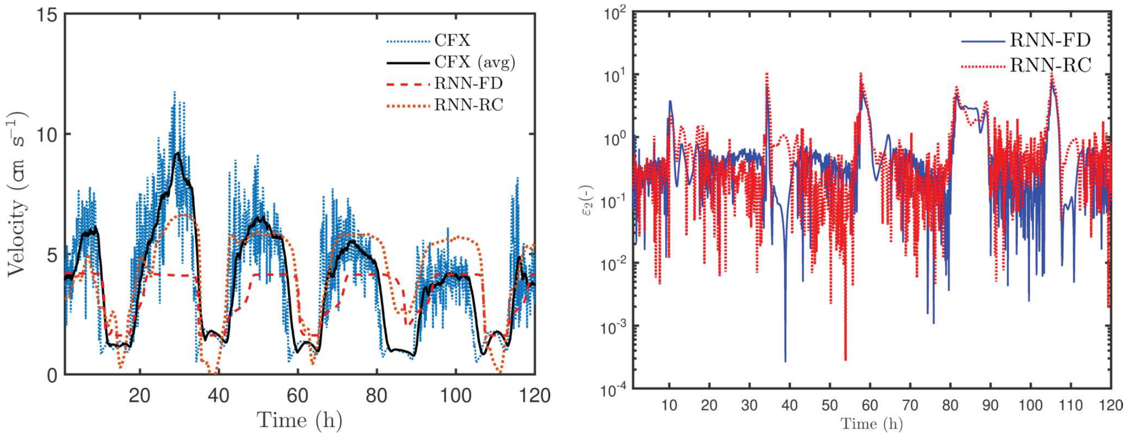
The RNN used in the intelligent co-simulation approach managed to predict satisfactorily the airflow (See Figures 4.44 and 4.45), to be used very rapidly in annual simulations, even varying the external boundary conditions, i.e., simulating the same building for other climates. The models built to predict the mass airflow were trained by using the representative climate.

The computational time for the 14-day simulation using the RNNs model is of the order of seconds, i.e., a way faster than the detailed `Domus-CFX` model. Once trained, the prediction model can be stored to further usage. The training phase generated 8 RNNs, one for each sensor and type of training. All with only one hidden layer and number of nodes ranging between 1 and 9, the networks to predict the air velocity tended to form networks with fewer nodes in the hidden layer. With the purpose of aiding reproducibility, some weights calculated for the generated networks are detailed in Tables 4.4 to 4.6. These weights are the information received from `Matlab` by `Domus`, enabling `Domus` to use the generated prediction model. For the cases of neural networks presented in Tables 4.4 to 4.6 the number of nodes in the hidden layer is 3, 1 and 2, respectively. The inputs defined for the training phase are the climate temperature and 2 radiation data that hit the wall closer to the evaluated sensor.



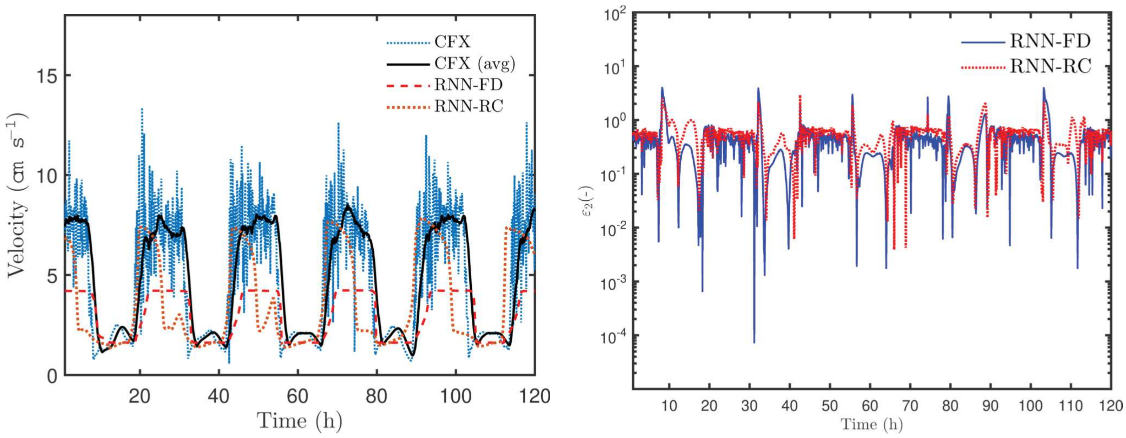
(a) (b)

Figure 4.39: Air temperature and RNN prediction for sensor 7 (a) and temperature prediction RMS error (b), in the summer period.



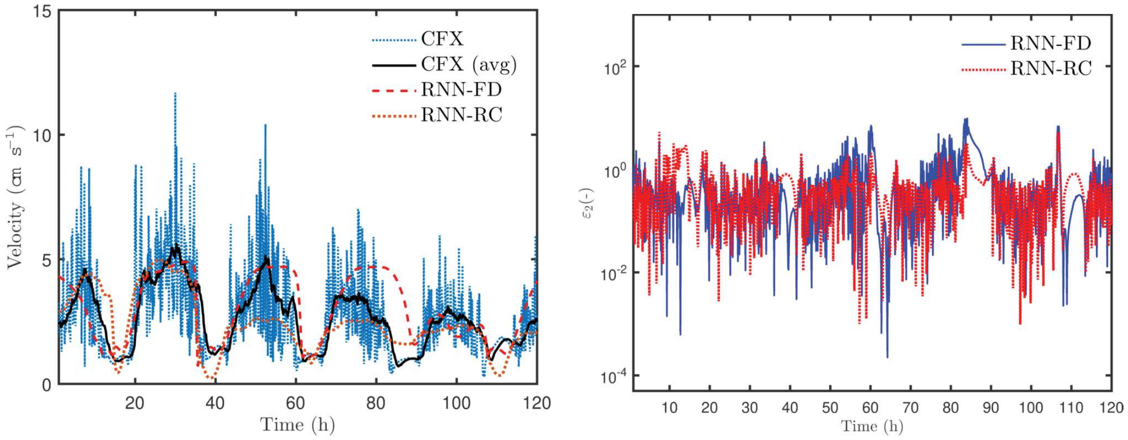
(a) (b)

Figure 4.40: Velocity and RNN prediction for sensor 1 (a) and prediction RMS error (b), in the winter period.



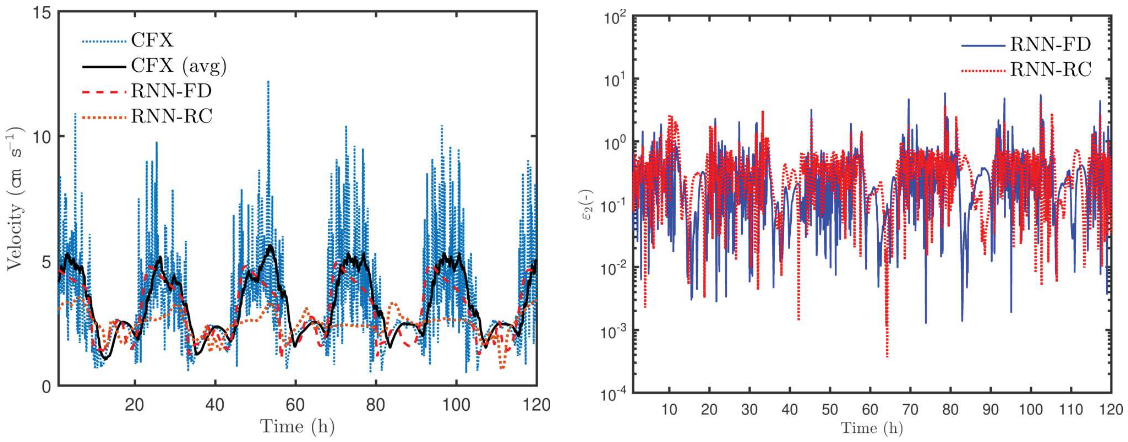
(a) (b)

Figure 4.41: Velocity and RNN prediction for sensor 1 (a) and prediction RMS error (b), in the summer period.



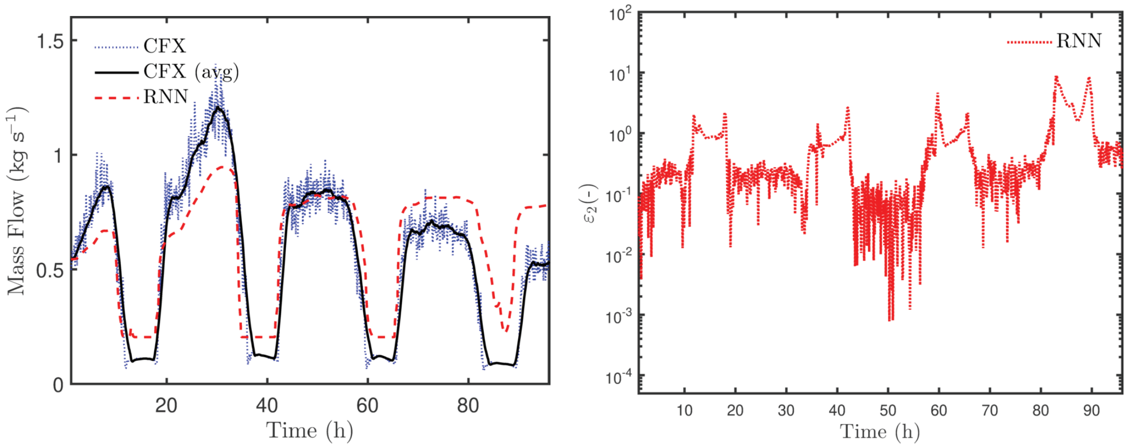
(a) (b)

Figure 4.42: Velocity and RNN prediction for sensor 7 (a) and prediction RMS error (b), in the winter period.



(a) (b)

Figure 4.43: Velocity and RNN prediction for sensor 7 (a) and prediction RMS error (b), in the summer period.



(a) (b)

Figure 4.44: Mass airflow at the opening between the two zones (a) and prediction RMS error (b), in the winter period.

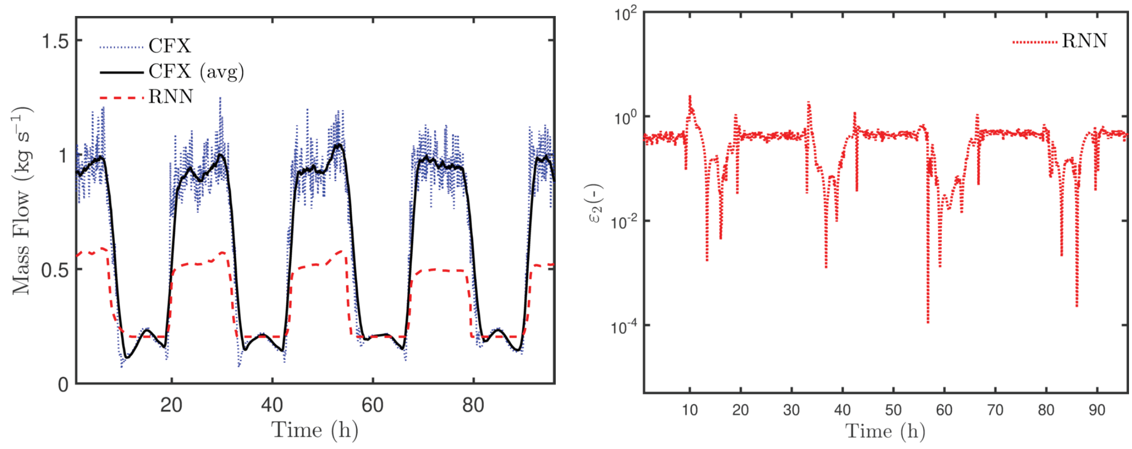


Figure 4.45: Mass airflow at the opening between the two zones (a) and prediction RMS error (b), in the summer period.

Table 4.4: Table of weights calculated for the RNN-FD model to predict temperature in sensor-1

	Weights between layers			
Context to hidden layer	9.8424804e-02	-9.1713454e-01	-2.7010306e+00	-1.7010766e-01
	-3.1092913e-01	6.8724874e-01	-8.9506219e-01	-4.9102203e-02
	-4.2154849e-02	-2.6310232e-01	5.6920730e-01	6.9145474e-03
Hidden to output layer	-7.7256000e-01	2.8537324e+00	8.1714411e-01	-2.9118199e-01
Input to hidden layer	1.6294683e+00	-7.2087459e-02	-1.4259023e-01	0.0000000e+00
	5.4253699e-01	-2.6040597e-02	-3.9985888e-02	0.0000000e+00
	3.3668291e-01	2.2085181e-02	-8.5153771e-03	0.0000000e+00

Table 4.5: Table of weights calculated for the RNN-RC model to predict temperature in sensor-1

	Weights between layers			
Context to hidden layer	9.8624231e-01	-2.1228470e-02		
Hidden to output layer	-2.8862003e+00	-3.5307469e-01		
Input to hidden layer	-7.5339910e-03	-3.4474396e-03	-1.5657778e-02	0.0000000e+00

4.5 Conclusions

An artificial intelligence based method to bring CFD to building simulation, called intelligent co-simulation, has been proposed in [61]. The method is based on the use of a machine learning model aiming at reducing the computational effort demanded by complex models. Latter, deeper exploration of this method was performed in [91], proposing representative climates in a pre-simulation training period, using artificially created climates to reduce training time. However, one of the issues that remained to be investigated was about the capability to represent the physical models that can be achieved using the *intelligent co-simulation* technique. This technique achieved predictions with low errors in 2-D and 3-D models, for restricted environments such as an attic.

This chapter proposed to raise to a new level the complexity of the case study, focusing mainly on the boundary conditions. Using a two-storey building with an approximate 723 m^3 total volume, innovative non-homogeneous radiative boundary conditions were applied in terms of realistic representation of the physics, locating the solar radiation in all internal and external surfaces, with great precision in the sunlit area thanks to the use of a pixel counting technique and a generic view factor algorithm. Such detailing in the sunlit boundary conditions is not performed in BPS tools of the current generation due to a series of reasons explained in this work. To improve the treatment of radia-

Table 4.6: Table of weights calculated for the RNN-RC model to predict temperature in sensor-7

Weights between layers				
Context to hidden layer	5.8690339e-01	-1.9596946e+00	-3.3550842e-01	
	-3.3454847e-01	-5.9639635e-01	-2.8481204e-01	
Hidden to output layer	2.0247697e+00	-2.4506445e+00	-2.0319824e-01	
Input to hidden layer	-2.0459216e-01	-7.7865198e-02	-2.8293448e-01	0.0000000e+00
	-1.7226308e-01	-6.4473632e-02	-2.3900859e-01	0.0000000e+00

tive boundary conditions, a new formulation to represent the incidence and reflections of radiation in building enclosures has been proposed along with calculation of sunlit to make the simulations more realistic, providing accurate boundary conditions to the co-simulation tool (`CFX`). The new radiation model has been implemented in the master co-simulation tool (`Domus`).

The results were presented in terms of 12 virtual sensors that captured temperature and air velocity at different points within a two-zone building. The mean air temperature and mass airflow between the two zones have also been calculated. These results showed considerable differences in temperature and air velocity within the same environment, reaching temperature differences close to 10°C between sensors.

When applying the intelligent co-simulation technique, the results have shown high predictive capacity of the physical behavior, despite the complexity of the case study, with average errors around 4.5% in the winter period and 5% in summer period for temperature captured in sensor 1, for example. It is expected that the prediction model will be able to overcome in accuracy the current BPS tools with simplified mathematical models. A representative climate was created artificially only to perform the training of prediction models more efficiently. It was also defined two training approaches of the prediction model, using the first 2-days of the co-simulation results and a second approach using the representative climate. The results showed that despite the limitations imposed, the representative climate achieved better results than the one that uses the first days of simulation, which reflect in a better capability to predict a whole year simulation.

As the results are considerably promising, showing an alternative to bring 3-D CFD computation to building performance assessment, further investigation should be stimulated by the use of the intelligent co-simulation approach considering other aspects such as: combined heat and moisture transfer; natural ventilation; HVAC systems and components and photovoltaic (PV) systems among others; Such evolution could be seen as a perspective of the next generation of building simulation tools. Regarding the learning and prediction parts, new challenges are expected in the calibration of prediction models because much more complex and slow physical phenomena will be involved.

Conclusion

Conclusion

The main contribution of this work is the development of a new co-simulation method to bring advanced physics to the building simulation tools. Building energy simulation programs usually present simplified mathematical models, such as lumped models for calculating room air temperatures, which might create considerable differences in the representation of physical phenomena. A first proposition in this work is to link specialized physical simulation tools and building simulation tools through co-simulation in order to extend their capabilities and to perform accurate whole-building energy simulations. It also shown that using co-simulation to obtain more accurate results is not a simple task, mainly due to the high computational cost. For the case studies presented in this work, the CPU time was greatly increased, in order of 45000 times, which makes the use of unsteady CFD simulation not affordable in the field of building performance analysis, at least for the computational capacity of the computers available nowadays. To provide a solution to the drawback associated to the high computational cost of detailed complex models such as unsteady CFD simulation, this work introduced a strategy - called *intelligent co-simulation* -, which consists of constructing a new model, called prediction model, capable to provide results, as close as possible to the ones provided by the complex model, with a much lower computer run time. Once the model is trained, the accurate model can be used for yearly simulation, running even faster than the simplified purely lumped model. In terms of accuracy, comparing the results of the standalone lumped model with the intelligent co-simulation model, the prediction model reached RMS errors up to one thousand percent smaller than those presented by the lumped model in relation to the CFD simulation result. Therefore, those presented results may highlight that this innovative strategy is very promising to accurately bring advanced physics to building simulation tools.

Also presented in this work, a deep investigation of two prediction techniques was performed - recurrent neural networks (RNN) and proper orthogonal decomposition (POD) reduction method -, whose main goal was reduce the training time period and to improve the accuracy. With the definition of a training climate, it was possible to require a shorter training phase (around 2 days for a total horizon simulation of 1 year), for example.

In the last part of this work, a case study created by adding complex physical events involved in the simulation of buildings such as: multidimensional heat transfer, complex shading elements, internal and between zones airflow, spatially variable convective heat transfer coefficients calculated according to realistic boundary conditions, in special, the accurate sunlit area over internal and external surfaces. At the end, the prediction technique, was applied to this complex case study to finally demonstrate the high potential of the intelligent co-simulation approach.

In addition as a possible further work, other neural network structures and new training models must be studied to reduce the training time period. Where the machine learning technique can be applied to the same geometry but for different climates, carrying out a procedure called adaptation of the neural network that will accumulate all the training performed. It is of great value for the building simulation field to achieve a prediction model that has generic operation for any climate. Another important subject is to develop and implement the co-simulation technique between the `Domus` program and CFD code with support for three-dimensional heat, air and humidity transfer model (HAM), and apply the intelligent co-simulation for the humidity problem, which can be considered a great challenge since the physical behavior is considerable different from that presented by the thermal analysis only.

Bibliography

[1] EPE Empresa de Pesquisa Energética. Brazilian energy balance 2017. *Rio de Janeiro: EPE*, 2017.

[2] L. E. Sulfstede. Applying power electronics to residential hvac-the issues. *IEEE transactions on industry applications*, 29(2):300–305, 1993.

[3] Berkeley LBNL. The home of DOE-2 based building energy use and cost analysis software. *http://doe2.com/DOE2/index.html*, 1970s.

[4] Washington National Institute of Building Sciences. Building loads analysis and system thermodynamics. *https://www.wbdg.org/tools/blast.php*, 1970s.

[5] Glasgow ESRU. Modelling tool for building performance simulation. *http://www.esru.strath.ac.uk/Programs/ESP-r.htm*, 1970s.

[6] DOE. Energyplus energy simulation software. *http://apps1.eere.energy.gov/buildings/energyplus/*, 2014.

[7] N. Mendes, R. C. L. F. Oliveira, and G. H. dos Santos. Domus 2.0: a whole-building hygrothermal simulation program. In *Proceedings of the 8th International Building Performance Simulation Association Conference*, pages 863–870, Eindhoven, Netherlands, August 2003.

[8] J. Hensen. A comparison of coupled and decoupled solutions for temperature and air flow in a building. Technical report, University of Strathclyde, Glasgow, UK, July 1999.

[9] M. Wetter. Co-simulation of building energy and control systems with the building controls virtual test bed. *Journal of Building Performance Simulation*, 4(3):185–203, 2011.

[10] J. Hensen, E. Djunaedy, M. Radošević, and A. Yahiaoui. Building performance simulation for better design: some issues and solutions. In *Proceedings of the 21st Conference on Passive and Low Energy Architecture (PLEA)*, pages 1185–1190, Eindhoven, Netherlands, September 2004.

[11] M. Trcka, J. L. M. Hensen, and M. Wetter. Co-simulation of innovative integrated HVAC systems in buildings. *Journal of Building Performance Simulation*, 2(3):209–230, 2009.

[12] M. Trcka. *Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings*. PhD thesis, Eindhoven University of Technology, Eindhoven, Netherlands, 2008.

[13] C .O. R. Negrão. *Conflation of Computational Fluid Dynamics and Building Thermal Simulation*. PhD thesis, University of Strathclyde, Glasgow, UK, 1995.

[14] I. Beausoleil-Morrison. *The adaptive coupling of heat and air flow modelling within dynamic whole-building simulation*. PhD thesis, University of Strathclyde, Glasgow, UK, May 2001.

[15] C. R. Maliska. Issues on the integration of CFD to building simulation tools. In *Proceedings of the 7th International Building Performance Simulation Association Conference*, Rio de Janeiro, RJ, Brazil, August 2001.

[16] J. Srebric, Q. Chen, and L. R. Glicksman. A coupled airflow and energy simulation program for indoor thermal environmental studies. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 2000.

[17] L. Wang and N. H. Wong. Coupled simulations for naturally ventilated rooms between building simulation (BS) and computational fluid dynamics (CFD) for better prediction of indoor thermal environment. *Building and Environment*, 44(1):95–112, 2009.

[18] E. Djunaedy, J. L. M. Hensen, and M. G. L. C. Loomans. Comparing internal and external run-time coupling of CFD and building energy simulation software. In *Proceedings of the 9th International Conference on Air Distribution in Rooms (ROOMVENT 2004)*, pages 393–396, Coimbra, Portugal, September 2004.

[19] C. Van Treeck, P. Wenisch, A. Borrmann, M. Pfaffinger, M. Egger, O. Wenisch, and E. Rank. Towards interactive indoor thermal comfort simulation. In *Proceedings of the European Conference on Computational Fluid Dynamics (ECCOMAS CFD 2006)*, Egmond aan Zee, Netherlands, September 2006.

[20] C. A. Rundle, M. F. Lightstone, P. Oosthuizen, P. Karava, and E. Mouriki. Validation of computational fluid dynamics simulations for atria geometries. *Building and Environment*, 46(7):1343–1353, 2011.

[21] E. Djunaedy. *External coupling between building energy simulation and computational fluid dynamics*. PhD thesis, Building and Planning, Eindhoven University of Technology, Eindhoven, Netherlands, 2005.

[22] E. Djunaedy, J. L. M. Hensen, and M. G. L. C. Loomans. Development of a guideline for selecting a simulation tool for airflow prediction. In *Proceedings of the 8th International Building Performance Simulation Association Conference*, pages 267–274, Eindhoven, Netherlands, August 2003.

[23] Hilding Elmqvist. Modelica — A unified object-oriented language for physical systems modeling. *Simulation Practice and Theory*, 5(6), August 1997.

[24] M. Wetter, M. Bonvini, T.S. Nouidui, W. Tian, and W. Zuo. Modelica buildings library 2.0. pages 387–394, 2015.

[25] N.-G. Christoph, M. Wetter, M. Fuchs, P. Grozman, L. Helsen, F. Jorissen, M. Lauster, D. Mulle, D. Picard, P. Sahlin, and M. Thorade. IEA EBC Annex 60 modelica library - An international collaboration to develop a free open-source model library for buildings and community energy systems. pages 395–402, 2015.

[26] N. L. Jones, D. P. Greenberg, and Pratt K. B. Fast computer graphics techniques for calculating direct solar radiation on complex building surfaces. *Journal of Building Performance Simulation*, 5(5):300–312, 2012.

[27] G. Cherem-Pereira. *Thermal Performance of Concrete Blocks, PhD qualification examination(in portugese)*. PhD thesis, Graduate Program in Mechanical Engineering, Pontifical Catholic University of Paraná, Brazil, 82p., 2016, 2016.

[28] P. J. Roache. Perspective: A method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, 116(3):405–413, September 1994.

[29] L. Rong, P. V. Nielsen, B. Bjerg, and G. Zhang. Summary of best guidelines and validation of CFD modeling in livestock buildings to ensure prediction quality. *Computers and Electronics in Agriculture*, 121:180–190, February 2016.

[30] N. Batista, E. L. La Rovere, and J. C. R. Aguiar. Energy efficiency labeling of buildings: An assessment of the brazilian case. *Energy and Buildings*, 43(6):1179 – 1188, 2011.

[31] A. Piot, C. Abele, M. Woloszyn, and J. Brau. Numerical simulation aided design of an experimental protocol. In *Proceedings of the 8th Symposium on Building Physics in the Nordic Countries*, volume 2, pages 731–738, Copenhagen, Denmark, June 2008.

[32] M. Woloszyn and C. Rode. Tools for performance simulation of heat, air and moisture conditions of whole buildings. *Building Simulation*, 1(1):5–24, 2008.

[33] F. Tariku, K. Kumaran, and P. Fazio. Transient model for coupled heat, air and moisture transfer through multilayered porous media. *International Journal of Heat and Mass Transfer*, 53(15-16):3035–3044, 2010.

[34] M. Steeman, A. Janssens, H. J. Steeman, M. Van Belleghem, and M. De Paepe. On coupling 1D non-isothermal heat and mass transfer in porous materials with a multizone building energy simulation model. *Building and Environment*, 45(4):865–877, 2010.

[35] SEL. Trnsys. *http://sel.me.wisc.edu/trnsys/features/features.html*, 2012.

[36] J. Clarke. Moisture flow modelling within the ESP-r integrated building performance simulation system. *Journal of Building Performance Simulation*, 6(5):385–399, 2013.

[37] N. Mendes, M. Chhay, J. Berger, and D. Dutykh. *Numerical methods for diffusion phenomena in building physics*. PUC Press, Curitiba, 2016.

[38] Z. Zhai, Q. Chen, P. Haves, and J. H. Klems. On approaches to couple energy simulation and computational fluid dynamics programs. *Building and Environment*, 37(8):857–864, 2002.

[39] W. Zuo, M. Wetter, W. Tian, D. Li, M. Jin, and Q. Chen. Coupling indoor airflow, HVAC, control and building envelope heat transfer in the Modelica Buildings library. *Journal of Building Performance Simulation*, 9(4):366–381, 2016.

[40] A. Kalogirou, C. C. Neocleous, and C. N. Schizas. Building heating load estimation using artificial neural networks. *Building and Environment*, 44(2):280–289, 2009.

[41] H. Jaeger. A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach. *GMD Report 159, German National Research Center for Information Technology*, page 48pp., 2013.

[42] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan. *Neural Network Design*. Martin Hagan, USA, 2nd edition, 2014.

[43] Y. Liu, J. A. Starzyk, and Z. Zhu. Optimized Approximation Algorithm in Neural Networks Without Overfitting. *IEEE Transactions On Neural Networks*, 19(6), March 2008.

[44] James A. Freeman and David M. Skapura. *Neural Networks Algorithms, Applications, and Programming Techniques*. Addison-Wesley Publishing Company, USA, 1991.

[45] A. H. Neto and F. A. S. Fiorelli. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. *Energy and buildings*, 40(12):2169–2176, 2008.

[46] R. Z. Freire, L. C. dos Santos, G. H. dos Santos, and V. C. Mariani. Predicting building's corners hygrothermal behavior by using a fuzzy inference system combined with clustering and Kalman filter. *International Communications in Heat and Mass Transfer*, 71:225–233, 2016.

[47] IEA Annex 60. Annex 60, new generation computational tools for building and community energy systems based on the modelica and functional mockup interface standards. *http://www.iea-annex60.org/pubs.html*, 2012.

[48] T. Nouidui, M. Wetter, and W. Zuo. Functional mock-up unit for co-simulation import in EnergyPlus. *Journal of Building Performance Simulation*, 7(3):192–202, 2014.

[49] W. Mazuroski, N. Mendes, and R. C. L. F. Oliveira. Co-simulation to bring advanced physics to building thermal performance analysis. In *Proceedings of the 15th International Building Performance Simulation Association Conference*, pages 2361–2369, San Francisco, CA, USA, August 2017.

[50] FMI-Standard. FMI-Standard Functional Mock-up Interface. `https://www.fmi-standard.org`, 2017. Accessed: 2017-07-03.

[51] D. J. C. MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

[52] Global Optimization Toolbox. User's guide (r2017b). *The MathWorks Inc*, 2017.

[53] K. Kuniar and M. Zajkac. Some methods of pre-processing input data for neural networks. *Computer Assisted Methods in Engineering and Science*, 22(2):141–151, 2015.

[54] A. P. A. Rocha, R. C. L. F. Oliveira, and N. Mendes. Experimental validation and comparison of direct solar shading calculations within building energy simulation tools: Polygon clipping and pixel counting techniques. *Solar Energy*, 158(Supplement C):462 – 473, 2017.

[55] N. Mendes, R. M. Barbosa, R. Z. Freire, and R. C. L. F. Oliveira. A simulation environment for performance analysis of HVAC systems. *Building Simulation*, 1(2):129–143, 2008.

[56] R. M. Barbosa and N. Mendes. Combined simulation of central HVAC systems with a whole-building hygrothermal model. *Energy and Buildings*, 40(3):276–288, 2008.

[57] R. Z. Freire, M. O. Abadie, and N. Mendes. Integration of natural ventilation models in the hygrothermal and energy simulation program PowerDomus. In *Proceedings of the 11th International Building Performance Simulation Association Conference*, pages 1037–1044, Glasgow, Scotland, July 2009.

[58] R. Z. Freire, M. Abadie, and N. Mendes. Numerical simulation of building-integrated photovoltaic systems. In *Proceedings of the 13th Brazilian Congress of Thermal Sciences and Engineering (ENCIT 2010)*, Uberlândia, MG, Brazil, December 2010.

[59] R. Z. Freire, W. Mazuroski, M. O. Abadie, and N. Mendes. Capacitive effect on the heat transfer through building glazing systems. *Applied Energy*, 88(12):4310–4319, 2011.

[60] Ansys Inc. Ansys CFX – solver theory guide. Technical report, Ansys Inc., 2015.

[61] W. Mazuroski, J. Berger, R. C. L. F. Oliveira, and N. Mendes. An artificial intelligence-based method to efficiently bring CFD to building simulation. *Journal of Building Performance Simulation*, 11(5):588–603, 2018.

[62] C. Miller, D. Thomas, J. Kampf, and A. Schlueter. Urban and building multiscale co-simulation: case study implementations on two university campuses. *Journal of Building Performance Simulation*, 0(0):1–13, 2017.

[63] G. H. dos Santos and N. Mendes. Simultaneous heat and moisture transfer in soils combined with building simulation. *Energy and Buildings*, 38(4):303–314, 2006.

[64] J. Berger, W. Mazuroski, S. Guernouti, N. Mendes, and M. Woloszyn. 2D whole-building hygrothermal simulation analysis based on a PGD reduced order model. *Energy and Buildings*, 112:49–61, January 2016.

[65] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating eulerian fluid simulation with convolutional networks. *CoRR*, abs/1607.03597, 2016.

[66] C. Allery, C. Béghein, and A. Hamdouni. Applying proper orthogonal decomposition to the computation of particle dispersion in a two-dimensional ventilated cavity. *Communications in Nonlinear Science and Numerical Simulation*, 10(8):907–920, December 2005.

[67] J. Berger, S. Tasca-Guernouti, M. Woloszyn, and C. Buhe. On the integration of hygrothermal bridges into whole building and HAM modeling. In *Proceedings of the 13th International Building Performance Simulation Association Conference*, Chambéry, France, August 2013.

[68] K. Karhunen. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung (About linear methods in probability theory)*, volume 37. Universitat Helsinki, 1947.

[69] M. Loeve. *Probability Theory I.* Springer Science & Business Media, March 1977.

[70] J. L. Lumley. The structure of inhomogeneous turbulent flows. In A. M. Yaglom and V. I. Tatarski, editors, *Atmospheric turbulence and radio propagation*, pages 166–178. Nauka, 1967.

[71] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993.

[72] P. J. Holmes, J. L. Lumley, G. Berkooz, J. C. Mattingly, and R. W. Wittenberg. Low-dimensional models of coherent structures in turbulence. *Physics Reports*, 287(4), 1997.

[73] C. Leblond, C. Allery, and C. Inard. An optimal projection method for the reduced-order modeling of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 200(33-36):2507–2527, 2011.

[74] D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, and A. Muggeridge. Non-intrusive reduced order modelling of the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 293:522–541, 2015.

[75] A. Sempey, C. Inard, C. Ghiaus, and C. Allery. Fast simulation of temperature distribution in air conditioned rooms by using proper orthogonal decomposition. *Building and Environment*, 44(2):280–289, 2009.

[76] O. Y. Chung H. M. Park. On the solution of inverse heat transfer problem using the Karhunen-Loéve Galerkin method. *International Journal of Heat and Mass Transfer*, 42(1):127–142, 1999.

[77] S.-J. Cao and J. Meyers. Fast prediction of indoor pollutant dispersion based on reduced-order ventilation models. *Building Simulation*, 8(4):415–420, 2015.

[78] W. S. Jung H. M. Park. The Karhunen-Loéve Galerkin method for the inverse natural convection problems. *International Journal of Heat and Mass Transfer*, 44(1):155–167, 2001.

[79] J. Berger, S. Guernouti, and M. Woloszyn. Evaluating model reduction methods for heat and mass transfer in porous materials : Proper orthogonal decomposition and proper generalised decomposition. *Journal of Porous Media*, 2016.

[80] S. Gasparin, J. Berger, D. Dutykh, and N. Mendes. Spectral methods – part 2: A comparative study of reduced order models for moisture transfer diffusive problems. submitted, 2017.

[81] Francisco Chinesta, Pierre Ladeveze, and Elías Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395, 2011.

[82] S. Gasparin, J. Berger, D. Dutykh, and N. Mendes. An improved explicit scheme for whole-building hygrothermal simulation. *arXiv preprint arXiv:1703.10119*, 2017.

[83] Ali H Nayfeh. *Perturbation methods.* John Wiley & Sons, 2000.

[84] W Kahan and J Palmer. On a proposed floating-point standard. *ACM SIGNum Newsletter*, 14(si-2):13–21, 1979.

[85] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations.* Cambridge University Press, USA, 2003.

[86] G. Box and G. Jenkins. *Time Series analysis : forecasting and control.* Wileys–Blackwell, USA, 1970.

[87] A.P.A. Rocha, R. C. L. F. Oliveira, and N. Mendes. Experimental validation and comparison of direct solar shading calculations within building energy simulation tools: Polygon clipping and pixel counting techniques. *Solar Energy*, 158:462 – 473, 2017.

[88] I. Maestre, L. Pérez-Lombard, J. Foncubierta, and P. Cubillas. Improving direct solar shading calculations within building energy simulation tools. *Journal of Building Performance Simulation*, 6(6):437–448, 2013.

[89] Y. Gao, J. J. Roux, C. Teodosiu, and L. H. Zhao. Reduced linear state model of hollow blocks walls, validation using hot box measurements. *Energy and buildings*, 36(11):1107–1115, 2004.

[90] A. Rodler, Joseph Virgone, and Jean-Jacques Roux. Impact of sun patch and three-dimensional heat transfer descriptions on the accuracy of a building's thermal behavior prediction. *Building Simulation*, 9(3):269–279, Jun 2016.

[91] J. Berger, W. Mazuroski, R. C. L. F. Oliveira, and N. Mendes. Intelligent co-simulation: neural network vs. proper orthogonal decomposition applied to a 2D diffusive problem. *Journal of Building Performance Simulation*, 11(5):568–587, 2018.

[92] M. Wall. Distribution of solar radiation in glazed spaces and adjacent buildings. a comparison of simulation programs. *Energy and Buildings*, 26(2):129 – 135, 1997.

[93] P. Tittelein. *Simulation environments adapted to the study of energy behaviour of low-consumption buildings*. PhD thesis, University of Savoie, France, 2008.

[94] Y. Boukhris, L. Gharbi, and N. Ghrab-Morcos. A sunspot model for energy efficiency in buildings. In Anne Hakansson, Mattias Höjer, Robert J. Howlett, and Lakhmi C Jain, editors, *Sustainability in Energy and Buildings*, pages 827–836, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[95] K. Weiler and P. Atherton. Hidden surface removal using polygon area sorting. In *ACM SIGGRAPH computer graphics*, volume 11, pages 214–222. ACM, 1977.

[96] J. F. Blinn and M. E. Newell. Clipping using homogeneous coordinates. In *ACM SIGGRAPH Computer Graphics*, volume 12, pages 245–251. ACM, 1978.

[97] B. R. Vatti. A generic solution to polygon clipping. *Communications of the ACM*, 35(7):56–63, 1992.

[98] S. Szokolay. *Introduction to architectural science 2nd Edition*. Routledge, 2008.

[99] Y. Cascone, V. Corrado, and V. Serra. Calculation procedure of the shading factor under complex boundary conditions. *Solar Energy*, 85(10):2524–2539, 2011.

[100] N. L. Jones and D. P. Greenberg. Fast computation of incident solar radiation from preliminary to final building design. In *Proceedings of the 12th International Conference of the International Building Performance Simulation Association*, pages 577–584, Sydney, Australia, 2011.

[101] Yezioro A. and Shaviv E. Shading: A design tool for analyzing mutual shading between buildings. *Solar Energy*, 52(1):27 – 37, 1994. Solar Buildings.

[102] A. Rodler. *Modélisation dynamique tridimensionnelle avec tache solaire pour la simulation du comportement thermique d'un bâtiment basse consommation*. PhD thesis, 2014. Thèse de doctorat dirigée par Virgone, Joseph Génie civil Lyon, INSA 2014.

[103] F. Déqué, F. Ollivier, and J.J. Roux. Effect of 2d modelling of thermal bridges on the energy performance of buildings: Numerical application on the matisse apartment. *Energy and Buildings*, 33(6):583 – 587, 2001.

[104] H. Ge and F. Baba. Dynamic effect of thermal bridges on the energy performance of a low-rise residential building. *Energy and Buildings*, 105:106–118, 2015.

[105] E. Wurtz, J.M. Nataf, and F. Winkelmann. Two- and three-dimensional natural and mixed convection simulation using modular zonal models in buildings. *International Journal of Heat and Mass Transfer*, 42(5):923 – 940, 1999.

[106] M. Wetter and C. v. Treeck. *IEA EBC Annex 60: New Generation Computing Tools for Building and Community Energy Systems*. September 2017.

[107] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[108] G. Bonaccorso. *Machine Learning Algorithms*. Packt Publishing, 2017.

[109] T.H. Kuehn, J.W. Ramsey, and J.L. Threlkeld. *Thermal Environmental Engineering*. Prentice Hall, 1998.

[110] A. Ribes and C. Caremoli. Salome platform component model for numerical simulation. In *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*, volume 2, pages 553–564, Beijing, China, July 2007.