UNIVERSITAT POLITÈCNICA DE CATALUNYA
PROGRAMA DE DOCTORAT EN ENGINYERIA CIVIL

DEPARTAMENT DE MATEMÀTICA APLICADA III

Doctoral Thesis

LOCAL SCALE AIR QUALITY MODEL SYSTEM FOR DIAGNOSTIC AND FORECASTING SIMULATIONS USING THE FINITE ELEMENT METHOD

by ALBERT OLIVER SERRA

Advisor: Agustí Pérez Foguet

September 2015

Acknowledgements

First of all, I would like to thank my advisor, Agustí Pérez Foguet. Without his guidance and encouragement throughout my time as a graduate student, this work would not have been possible. His insightful ideas contributed immeasurably to the research presented here, and his mentoring has been crucial to my development as a researcher.

I want to express my deepest gratitude to Rafael Montenegro for having trusted me, and given me the opportunity to work in his research group in the Universidad de Las Palmas de Gran Canaria. His lessons, commitment, and support have made a big impact on my research. Thanks to Gustavo Montero, Eduardo Rodríguez, and José María Escobar for such a warm welcome and for making me feel one more of the group.

I would also like to thank all the people in LaCàN that have shared some part of my PhD. Thanks to Antonio Rodríguez for the discussions on the modelling of the problem, to Sonia Fernández for the initial involvement, and to Josep Sarrate for his interest in my thesis and for the support in the research projects that we have shared.

Also thanks to the people in “la sala de gràfics”; thanks to Xevi and Eloi for their support and their help with the generation of meshes, to David Modesto for the discussions and chats, and to all of you that have made these years an enriching experience: Imma, Esther, Eva, Jordi, Nati, Rubén, Adrián, Raúl, Cristina, Alba, Elena, Raúl, Raquel, and all of you that I forget to mention. I also would like to thank my colleagues from ULPGC: Marina, Iván, Guillermo, and Jabel for the great atmosphere.

Gràcies a la meva família pel suport que he rebut i pels ànims que m’heu donat. Gràcies a la meva àvia i la meva mare per haver-me aguantat durant aquest final de tesis, i al meu germà per haver cregut sempre en mi. Finalmente, muchas gracias a Irene por estar allí, animándome y haciendo que todo parezca fácil.

Albert Oliver Serra
Barcelona, September 2015

List of Contributions

Publications

A. Oliver, E. Rodríguez, J.M. Escobar, G. Montero, M. Hortal, J. Calvo, J.M. Cascón, and R. Montenegro. Wind forecasting based on the HARMONIE model and adaptive finite elements. *Pure and Applied Geophysics*, 172(1):109–120, 2015. doi:10.1007/s00024-014-0913-9.

A. Oliver, G. Montero, R. Montenegro, E. Rodríguez, J.M. Escobar, and A. Pérez-Foguet. Adaptive finite element simulation of stack pollutant emissions over complex terrains. *Energy*, 49(0):47–60, 2013. doi:10.1016/j.energy.2012.10.051.

A. Oliver, G. Montero, R. Montenegro, E. Rodríguez, J.M. Escobar, and A. Pérez-Foguet. Finite element simulation of a local scale air quality model over complex terrain. *Advances in Science and Research*, 8:105–113, 2012. doi:10.5194/asr-8-105-2012.

Registered Software

E. Rodríguez, G. Montero, J.M. Escobar, R. Montenegro, and A. Oliver. Wind3D. `http://www.dca.iusiani.ulpgc.es/Wind3D`, 2012.

International Conferences

A. Oliver, A. Pérez-Foguet, E. Rodríguez, G. Montero, and R. Montenegro. Local Scale Air Quality Model with Several Pollutant Sources. In *Fourth African Conference on Computational Mechanics (AfriCOMP15)*, Marrakech, January 2015.

A. Oliver, E. Rodríguez, J.M. Escobar, G. Montero, M. Hortal, J. Calvo, J.M. Cascón, and R. Montenegro. Ensemble Wind Forecasting Based on the HARMONIE Model and Adaptive Finite Elements in Complex Orography. In *14th EMS Annual Meeting & 10th European Conference on Applied Climatology (ECAC)*, Prague, October 2014.

J. Ramírez, A. Oliver, and R. Montenegro. Adaptive finite element method with a local element parametrization of nested meshes. In *11th World Congress on Computational Mechanics, 5th European Conference on Computational Mechanics, 6th European Conference on Computational Fluid Dynamics*, Barcelona, July 2014.

A. Oliver, E. Rodríguez, G. Montero, and R. Montenegro. Wind ensemble forecasting using an adaptive mass-consistent model. In *11th World Congress on Computational Mechanics, 5th European Conference on Computational Mechanics, 6th European Conference on Computational Fluid Dynamics*, Barcelona, July 2014.

G.V. Socorro, E. Ruiz-Gironés, A. Oliver, J.M. Cascón, J.M. Escobar, J. Sarrate, and R. Montenegro. Tetrahedral mesh optimization combining boundary and inner node relocation and local adaptive refinement. In *11th World Congress on Computational Mechanics, 5th European Conference on Computational Mechanics, 6th European Conference on Computational Fluid Dynamics*, Barcelona, July 2014.

J. Ramírez, A. Oliver, J.E. González, and R. Montenegro. Calibration and validation of an air quality finite element model around an electric power plant in Gran Canaria island. In *13th EMS Annual Meeting & 11th European Conference on Applications of Meteorology*, Reading, September 2013.

R. Montenegro, A. Oliver, E. Rodríguez, J.M. Escobar, G. Montero, and A. Pérez-Foguet. Local scale finite element modelling of stack pollutant emissions. In *Conference on Computational Methods for Coupled Problems in Science and Engineering*, Ibiza, June 2013.

A. Oliver, and A. Pérez-Foguet. Local scale air quality modelling based on CMAQ forecast data. In *12th EMS Annual Meeting & 9th European Conference on Applied Climatology*, Łódź, September 2012.

R. Montenegro, A. Oliver, E. Rodríguez, J.M. Escobar, G. Montero, and A. Pérez-Foguet. Three-Dimensional Finite Element Modelling of Stack Pollutant Emissions. In *Proceedings of the Eighth International Conference on Engineering Computational Technology*, Dubrovnik, September 2012. doi:10.4203/ccp.100.35.

A. Oliver, A. Pérez-Foguet, E. Rodríguez, J.M. Escobar, R. Montenegro, and G. Montero. A local scale finite element moldel for stack pollutant dispersion over complex terrains. In *11th EMS Annual Meeting & 10th European Conference on Applications of Meteorology*, Berlin, September 2011.

A. Pérez-Foguet, and A. Oliver. Coupling mesoscale air quality models with finite element computation of dispersion at local level. In *European Geosciences Union General Assembly 2009 (Poster)*, Viena, April 2009.

A. Pérez-Foguet, and A. Oliver. Coupling regional and local air quality models for short-time prediction around punctual pollutant sources using finite elements. In *8th World Congress on Computational Mechanics and 5th European Congress on Computational Methods in Applied Sciences and Engineering*, Venezia, June-July 2008.

A. Oliver, J.M. Escobar, E. Rodríguez, A. Pérez-Foguet. Finite element local air quality modeling of punctual emissions. In *8th World Congress on Computational Mechanics and 5th European Congress on Computational Methods in Applied Sciences and Engineering*, Venezia, June-July 2008.

A. Pérez-Foguet, A. Oliver, J.M. Escobar, and E. Rodríguez. Finite Element Simulation of Chimney Emissions: A Proposal for Near Field Impact Assessment in Highly Complex Terrains. In *Proceedings of the Fifth International Conference on Engineering Computational Technology*, Las Palmas de Gran Canaria, September 2006. doi:10.4203/ccp.84.101.

National Conferences

G.V. Socorro, E. Ruiz-Gironés, A. Oliver, J.M. Cascón, J.M. Escobar, J. Sarrate, and R. Montenegro. Surface Insertion in a Tetrahedral Mesh Using The Meccano Method. In *Congreso de Métodos Numéricos en Ingeniería*, Lisboa, July 2015.

J. Ramírez, A. Oliver, and E. Rodríguez. Application of Genetic Algorithms for the Calibration of an Air Quality Model and its Validation using Pollutant Measures from the Surroundings of an Electric Power Plant. In *Congreso de Métodos Numéricos en Ingeniería*, Lisboa, July 2015.

E. Rodríguez, A. Oliver, R. Montenegro, and G. Montero. Desarrollo de un Método Ensemble para la Predicción del Viento a Escala Local usando Elementos Finitos. In *Congreso de Métodos Numéricos en Ingeniería*, Lisboa, July 2015.

A. Oliver, E. Rodríguez, J. Ramírez, J.I. López, M. Brovka, J.M. Escobar, J.M. Cascón, F. Díaz, G.V. Socorro, R. Montenegro, and G. Montero. 3-D Wind Field Simulation over Complex Terrain. In *Congreso anual de la Real Sociedad Matemática Española (RSME)*, Granada, February 2015.

A. Oliver, E. Rodríguez, J.M. Escobar, G. Montero, M. Hortal, J. Calvo, and R. Montenegro. Wind forecasting over complex terrain. In *Mathematics and Geosciences: Global and Local Perspectives*, Madrid, November 2013.

A. Oliver, R. Montenegro, E. Rodríguez, J.M. Escobar, G. Montero, and A. Pérez-Foguet. Simulación de la calidad del aire en la isla de Gran Canaria mediante el método de elementos finitos y su validación con datos experimentales. In *Congress on Numerical Methods in Engineering*, Bilbao, June 2013.

A. Oliver, L. Monforte, and A. Pérez-Foguet. Simulación 3D de la calidad del aire en un valle andino con condiciones atmosféricas estables y viento en calma. In *Congress on Numerical Methods in Engineering*, Coimbra, June 2011.

A. Pérez-Foguet, and A. Oliver. Acoplamiento del modelo CMAQ con un modelo de Elementos Finitos para el transporte de contaminantes de plumas de chimeneas. In *Congreso de Métodos Numéricos en Ingeniería*, Barcelona, June-July 2009.

Workshops and Invited Talks

A. Pérez-Foguet, A. Oliver, E. Rodríguez, J.M. Escobar, R. Montenegro, and G. Montero. High-resolution simulation of stack pollutant emissions. In *The 11^{th} Workshop on Numerical Methods in Applied Science and Engineering*, Barcelona, January 2012.

A. Oliver. Simulació de la qualitat de l'aire a la zona de l'Oroya (Perú). In *IX Jornades Enginyeria Geològica*, Barcelona, March 2010.

A. Pérez-Foguet, and A. Oliver. Simulación de la dispersión de contaminantes de La Oroya (2007-2008) y alternativas de actuación. In *The 8^{th} Workshop on numerical methods in applied science and engineering*, Vall de Núria, January 2009.

A. Pérez-Foguet, A. Oliver. Acoplamiento del modelo regional de calidad de aire (CMAQ) con el modelo de emisiones locales mediante elementos finitos. In *7^{th} Workshop on numerical methods in applied science and engineering (NMASE 08)*, Vall de Núria, January 2008.

A. Pérez-Foguet, A. Oliver. Modelización mediante elementos finitos de emisores puntuales de contaminantes atmosféricos. In *6^{th} Workshop on numerical methods in applied science and engineering (NMASE 07)*, Vall de Núria, January 2007.

A. Pérez-Foguet, A. Oliver. Acoplamiento de un modelo local de transporte y reacción de contaminantes atmosféricos con modelos regionales de calidad de aire. In *6^{th} Workshop on numerical methods in applied science and engineering (NMASE 07)*, Vall de Núria, January 2007.

Abstract

Air pollution is an important topic with a great social impact; it is related with public health, environment and ecology, and climate change. Scientists have developed several models in the last thirty years, and regional air quality operational systems are used routinely by governments and agencies. Efforts have also been done to simulate the air quality in the local scale; main models are Gaussian and Puff models, that are based on a Lagrangian approach. In contrast with these models, in this thesis we have developed a system using an Eulerian approach. This model is specifically designed for regions with complex orography where the Lagrangian models have problems computing the trajectory of the particles. This model can be used for diagnostic or prediction simulations.

Air quality operational systems depend on the orography, meteorological data, and emission data. Air quality models use processors to incorporate these data into the model. The data can come from numerical weather prediction systems, experimental data, or databases. In this thesis we have developed processors, specifically designed for the local scale, to incorporate these data into our system.

To incorporate the orography, we have developed a mesh generation algorithm suitable for complex terrain discretization; it also allows to insert layers that can match the regional models. A wind field model has also been used; it can interpolate a three-dimensional wind field from some station measurements using a log-linear vertical wind profile, or can interpolate it from a numerical weather prediction system. Once an interpolated wind field is computed, a mass-consistent model is applied to ensure null divergence and impermeability in the terrain.

The wind field is modified to take into account the injection of the pollutants into the atmosphere. Briggs studied the trajectory of the plume rise giving some empirical equations that will be used in our model. Briggs' equations describe the trajectory in a plane; our model will modify this trajectory adapting it to the ambient wind field. This modification allows the plume rise to surround the mountains or channel into the valleys.

The transport and reaction of pollutants in the atmosphere is then computed using an splitting method, so the transport and the chemical reactions are computed independently. To solve the transport of pollutants we have used a finite element method stabilized using least squares. The chemical reaction is simulated using simplified models such as RIVAD, or more complex ones such as CB05.

To obtain more accurate results we have used adaptation. An error indicator has been used to adapt the mesh to the solution. To adapt the mesh to the concentration distribution of all the species is very demanding, for this reason we have used a multimesh method where every chemical specie has its own mesh where we solve the transport and the chemical reactions are simulated in a common mesh.

The system developed in this thesis has diagnostic and forecasting capabilities. For this reason we present two different applications. The first one is a diagnostic application in La Palma island (Spain), where wind measurements are given, and SO_2 and NO_2 emissions from a stack are considered. The topography of the island is real, from a digital elevation model, but the wind field measurements, and the stack location and emissions, are simulated. The second application is a forecasting application data from the CMAQ benchmark test. It is located in the surrounding of Pineville Kentucky. In this application we have used all the data from CMAQ and the chemical reaction model CB05.

Resum

La contaminació atmosfèrica té gran impacte social; està relacionada amb la salut pública, l'ecologia, el medi ambient i el canvi climàtic. En els últims trenta anys, els científics han desenvolupat diversos models que els governs i les agències mediambientals utilitzen diàriament. També s'han fet esforços per simular la qualitat de l'aire en l'escala local. Els models principals són els models Gaussians i Puff, que es basen en una descripció Lagrangiana. En contrast amb aquests models, en aquesta tesi s'ha desenvolupat un sistema que utilitza una descripció Euleriana. Aquest model està dissenyat específicament per a les regions amb topografia complexa on els models Lagrangians tenen problemes calculant la trajectòria de les partícules. El model presentat en aquesta tesi pot ser utilitzat tant pel diagnòstic com per la predicció.

Els sistemes operatius utilitzats actualment depenen de l'orografia, de dades meteorològiques i de dades d'emissió. Per tal d'incorporar aquestes dades, els models de qualitat de l'aire han desenvolupat diferents preprocessadors. Les dades poden venir dels sistemes numèrics de predicció meteorològics, de dades experimentals o de bases de dades. En aquesta tesi hem desenvolupat preprocessadors dissenyats específicament per a l'àmbit local, per tal d'incorporar aquestes dades al nostre sistema.

Per incorporar l'orografia, hem desenvolupat un algoritme de generació de malles adequat per terreny complex; l'algoritme també ens permet inserir capes que poden coincidir amb la dels models regionals. S'ha desenvolupat un model de càlcul de camp de vent; a partir de les dades s'interpola un camp de vent tridimensional fent servir un perfil log-lineal vertical del vent, o s'interpola a partir dels resultats d'un sistema de predicció meteorològica. Quan s'ha calculat el vent interpolat, es fa servir un model de massa consistent per obtenir un camp de vent definitiu on s'ha imposat divergència zero i impermeabilitat del terreny.

El camp de vent s'ha de modificar per tenir en compte la injecció dels contaminants a l'atmosfera. Briggs va estudiar la trajectòria de l'elevació de la ploma i va trobar equacions empíriques que utilitzarem en el nostre model. Les equacions de Briggs descriuen la trajectòria de la ploma en un pla vertical; el nostre model modificarà aquesta trajectòria adaptant-la al camp de vent ambiental. Aquesta modificació permet que l'elevació de la ploma rodegi les muntanyes o s'acanali a les valls.

El transport i la reacció de contaminants a l'atmosfera es calcula utilitzant un mètode de "splitting", de manera que el transport i la reacció química es calculen de forma independent. Per resoldre el transport de contaminants, hem utilitzat el mètode d'elements finits estabilitzat amb mínims quadrats. La reacció química es simula mitjançant models simplificats com el model RIVAD, o més complexes com el CB05.

Per obtenir resultats més precisos hem adaptat la malla a la solució utilitzant un indicador d'error. Haver d'adaptar la malla a la distribució de la concentració de totes les espècies contaminants és molt exigent i per aquest motiu hem utilitzat un mètode "multimesh" on cada espècie química té la seva pròpia malla on resolem el transport i les reaccions químiques es simulen en una malla comú.

El sistema desenvolupat en aquesta tesi té capacitats de diagnòstic i pronòstic. Per aquesta raó es presenten dues aplicacions diferents. La primera és una aplicació de diagnòstic a l'illa de la Palma (Espanya), on es disposa de mesures de vent, i de les emissions de SO_2 i NO_2 d'una xemeneia. La topografia de l'illa és real, a partir d'un model digital del terreny, però les dades del camp de vent, la ubicació de la xemeneia i el valor de les emissions són simulades. La segona aplicació és una aplicació de predicció fent servir els resultats del benchmark del CMAQ. Es simula una zona a Pineville Kentucky. En aquesta aplicació hem utilitzat totes les dades del CMAQ i el model químic CB05.

Contents

List of Figures

List of Tables

1 Introduction

Air pollution is an important topic with a great social impact; it is related with public health [Pascal et al., 2013, Walton et al., 2015], environment and ecology [Abelson, 1985, Greaver et al., 2012], and climate change [Ramanathan and Feng, 2009, Patz et al., 2005]. For this reason, in the last thirty years, scientists have developed numerical methods to simulate the transport and reaction of pollutants on the atmosphere. However, today it remains as a scientific challenge. The transport and reaction of pollutants in the atmosphere occurs in multiphase systems that involve non-linear processes and are coupled between different time and space scales; this difficulties are of paramount importance when analysing climate modelling, acid rain, ozone, particle matter, and toxic emissions [Seigneur, 2005, Boyd, 2008].

Air quality modelling systems mainly involve three components: emissions, meteorology and transport–chemistry. The first component characterizes different emission sources of chemical compounds, the second one determines the atmospheric phenomena as wind and temperature fields, and the third one simulates the transport of pollutants (convection and diffusion) and their chemical reactions.

Several air quality modelling systems exist; the United States Environmental Protection Agency (`www.epa.gov`) classifies them as Dispersion, Photochemical, and Receptor models. Dispersion models estimate pollutant concentrations at ground level near a punctual source. Photochemical models consider all the sources in a large area. Receptor models identify and characterize the emission sources using receptor measures.

Main dispersion models include Gaussian plume models [Olcese and Toselli, 2005], particle tracking models [Hernandez et al., 1995], and puff models [Bourque and Arp, 1996, Souto et al., 2001, Walcek, 2002]. They are based on a Lagrangian approach and usually include first order chemistry reactions or linearized photochemical-reaction models. Versions with complex non-linear photochemical-reaction models have been also developed [Hurley et al., 2005].

In contrast with dispersion models, photochemical models follow an Eulerian description of the coupled transport and hydrodynamic problem. Most of the models approximate the solution to this problem by using finite differences schemes. This models cover domains in the low range of thousands of kilometres, using a grid spatial discretization of 16 or 32 km. Specific areas of interest (about tens of kilometres) are simulated with a higher resolution by using nested sub-grids. The finest horizontal resolution is usually of two or four kilometres, even though one kilometre resolutions can be used [Morris et al., 2002, Taghavi et al., 2005, San José et al., 2007, Pay et al., 2014]. The height of the computational domain usually goes from 4 km to 10 km with a non-uniform vertical discretization between 10 and 20 levels. Vertical spatial resolution varies from few tens of metres close to the ground level to one thousand metres exceeding two kilometres above ground level. The number of grid points in this kind of problems can vary from tens to hundreds of thousand points. The most advanced photochemical models include puff models to treat punctual emission. These models are known as hybrid ones, e.g. CAMx [ENVIRON International Corporation, 2015], and CMAQ [Byun and Ching, 1999].

In environmental management, dispersion and photochemical models have very different applications. The first ones are usually applied to punctual emission impact assessments, and the second ones to regional planning and monitoring. Both models have a clearly different application scale and photochemical reaction complexity. However, because of the great coupling among key components as ozone, nitrogen compounds, and Volatile Organic Components [Ryerson et al., 2001, Lee et al., 2003], and due to the awareness about the socio-economic impacts of their immission [Mauzerall et al., 2005, He et al., 2010], some references about the need of coupling punctual emissions using regional planning with hybrid photochemical models can be found [Hidy, 2002, Morris et al., 2002, Taghavi et al., 2005]. Although hybrid models can couple local scales (up to a resolution of one kilometre) with regional scales, several limitations has been reported [Morris et al., 2002]. For example, it is clear that this local high resolution can be insufficient for complex terrains. Thus, the search of alternatives is justified.

This thesis presents a new methodology for local scale air quality simulations by using a nonsteady finite element method with unstructured and adaptive tetrahedral meshes. The aim of this thesis is to introduce an alternative to the standard implementation of current models, improving the limitations of local scale air quality simulations [Morris et al., 2002, Hanjalić and Kenjereš, 2005]. Major contributions to the overall goal of this thesis have been presented in Oliver et al. [2013, 2015a]. The proposed system can be used with different chemical models and be applied in diagnostic or forecasting applications.

Three remarkable uses of unstructured meshes in atmospheric pollution problems are the two-dimensional regional–global examples presented in Lagzi et al. [2004], Ahmad et al. [2006], the three-dimensional regional examples, including local refinement with element sizes of 2 km, presented in Tomlin et al. [2000], and the three-dimensional tetrahedral meshes for local wind field analysis with element sizes ranging from two metres up to two kilometres, see Montenegro et al. [2004], Montero et al. [2004]. Also there is a novel method using an adaptive multimesh scheme based on unstructured meshes with element sizes ranging from hundreds of metres to centimetres [Monforte and Pérez-Foguet, 2014a,b].

The ideas of the approach presented in Montenegro et al. [2004], Montero et al. [2004] are considered to determine the wind field, that includes the effect of the plume rise, used in the air quality problem. The wind field is crucial for the pollutant transport, specially in complex terrain areas. In order to simulate it, we have used a mass-consistent model. Several two-dimensional [Winter et al., 1995] and three-dimensional [Montero et al., 1998, 2005, Ferragut et al., 2010] adaptive finite element solutions have been developed. The mass-consistent model has also been used as a local scale forecasting model by coupling it with the HARMONIE mesoscale numerical weather prediction model [Oliver et al., 2015b].

The ideas of Pérez-Foguet et al. [2006], Oliver et al. [2012, 2013] and Monforte and Pérez-Foguet [2014a,b] are considered to solve the convection–diffusion–reaction equation using the finite element method. Usually, the convection, diffusion and reaction problem is solved using splitting schemes [Martín et al., 2003, Chock et al., 2005] and specific numerical solvers for time integration of photochemical reaction terms [Saylor and Ford, 1995, Sandu et al., 1997b, Ahmad et al., 2006]. A stabilized finite element formulation, specifically Least-Squares, with a Crank-Nicolson temporal integration is proposed to solve the non-steady and non-linear transport-reaction problem [Donea and Huerta, 2003, Rodríguez-Ferran and Sandoval, 2007]. The chemistry is simulated by using the RIVAD chemical model [Scire et al., 2000], or the CB05 chemical mechanism [Yarwood et al., 2005]. The transport and chemical terms are treated separately with Strang splitting operators [Ropp et al., 2004]. The non-linear chemical part is solved node by node with a second order Rosenbrock method [Verwer et al., 1999]. The mesh used to solve this equation will be adapted to the terrain and the plume rise [Pérez-Foguet et al., 2006, Oliver et al., 2012, 2013] and then it will be adapted to the solution using and independent mesh for every pollutant specie [Monforte and Pérez-Foguet, 2014a,b].

2 Air quality model overview

Air quality models are systems that share the same structure. They are composed by several models. The most important ones are the wind field, the plume rise, and the transport and reaction of pollutants. This applies to local and global models, as well as diagnostic and predictive ones. In this Chapter we will describe these components for the local scale model proposed in this thesis. The predictive capability of the local model is achieved using the predictions from the CMAQ community system as input data. So, we will present the CMAQ model and its discretization, a key aspect for the implementation, and finally the interpolation between the CMAQ model and the local scale model.

2.1 Local scale model overview

The presented system includes three processors to incorporate the orography, and the meteorological and emission data into the model. Initial and boundary conditions are also incorporated. A general view of the system is described in Fig. 2.1, the arrows show the flow of data through the modelling system. This flow is the same for both the diagnostic and the forecast simulations, being the main difference the origin of the data, as well as some smalls variations in the methods of the processors. These processors and the main model are described shortly below; for each topic we have referenced the Chapter where it is discussed.

- The *Mesh Generation Processor* reads the terrain elevations from a Digital Terrain Model (DTM) and generates a tetrahedral mesh of the domain of interest. The resulting mesh is adapted to the terrain. The mesh is optimized to ensure a minimum quality. (Chapter 3)

- The *Wind Processor* uses meteorological data to compute a wind field in the domain of interest. The processor imposes incompressibility and impermeability in the terrain by using a mass-consistent model. The meteorological data is different depending on the application; for a diagnostic application it is from meteorological stations measurements, while for a forecast application we use the output from a numerical weather predicition system. (Chapter 4)

- The *Plume Rise Processor* is responsible to compute the emission plume rise. It will use emissions data and the geometric characteristics of the stack. Briggs formulation is used to compute the trajectory and final elevation and distance of the plume. (Chapter 5)

- The *Air Quality Model* is the main module of the System. It computes the transport and reaction of the pollutant in the atmosphere. To deal with the complexity, we will use an splitting technique to treat the transport and the reaction separately. The transport is driven by the

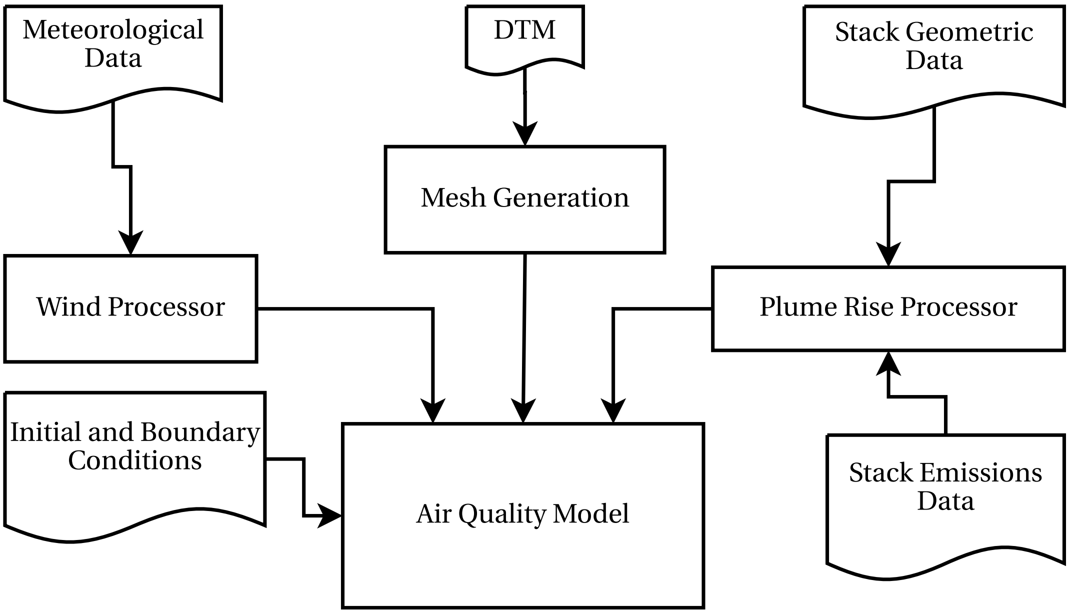


Figure 2.1: Air Quality Model System General diagram

wind field computed by the processor, and a diffusion term. This diffusion term will be computed in this processor by using a model for the horizontal diffusion and another one for the vertical diffusion. The whole transport will be computed using the Finite Element Method with a Least-Squares stabilization. The reaction can be solved using some simplified models or a more complete photolytic model as CB05. To have a reliable solution adaptivity is mandatory, and an adaptive multimesh scheme is used to reduce the system size and to parallelize the computations. (Chapter 6)

Algorithm 2.1 shows the main steps in the local scale model in a more detailed view.

In the diagnostic model, usually the data is from measurements in meteorological stations and immissions stations. The predictive capacity of the model is given by using forecast data to be able to simulate future events. An air quality mesoscale forecast system has been used to get the future data. This data will be downscaled into the local scale region of the model. This mesoscale model is the Community Multiscale Air Quality Model (CMAQ). To show both capabilities, two different applications are shown; in Chapter 7 a diagnostic application in La Palma island (Canary islands, Spain) is presented, while in Chapter 8 it is a forecast application using the CMAQ benchmark test case.

The time scale of the problems solved with the local scale model is the short-range forecasting; i.e. from one to three days. Usually an output every hour or every few hours (e.g. 3 hours) is saved. This temporal range means that the relevant physical aspects of the wind field are the ones given at the local scale such as hourly changes as well as the day-night regimes; regarding the chemical reactions we are focusing on those that happens in the hourly range. We want to remark that given this time scale, the number of downscaling needed to solve the predictive problem is up to 100 — one for each input change—, so the interpolation is an important part of the algorithm.

In the following Section we present the CMAQ model, describing briefly the main components, and taking special attention to the spatial discretization of the model, how data is saved in the CMAQ

Algorithm 2.1 Local Scale Air Quality Model

```
 1: Construct a adaptive tetrahedral mesh of the domain
     1.1: Adaptive discretization of the terrain surface
     1.2: Vertical spacing function and 3D distribution of points
     1.3: Three-dimensional mesh generation
     1.4: Mesh optimization
 For each time step where data is available
   2: Wind field simulation from measurement stations of forecast model data
       2.1: Interpolate wind field data
       2.2: Approximation of the wind field with a mass-consistent model
   3: Wind field modification including the plume rise effect
       3.1: Compute a plume rise trajectory
       3.2: Mesh refinement along the plume rise trajectory
       3.3: Simulate a wind field in the new mesh (applying step 2)
       3.4: Modify vertical components of the wind field along the plume rise
   4: Available data from measurement stations or forecast model (diffusion, temperature, wet
 deposition, boundary condition)
       4.1: Horizontal interpolation
       4.2: Vertical interpolation
   5: if (first time step) interpolate initial condition
   6: Multimesh air pollution simulation from stack emission data
       6.1: Compute each specie mesh depending on the first local time steps using error indicator
       6.2: Compute transport and reaction of pollutants re-adapting the mesh every some time
 steps
```

grid and how MCIP works. To know how data is organized in CMAQ will let us to interpolate data between CMAQ and our local model.

2.2 CMAQ system overview

CMAQ is a regional Air Quality Model used in the US Environment Protection Agency (EPA). The CMAQ system is suited both for regulatory issues and scientific research. The Chemical Transport Model (CTM) is the core of the system; it simulates the transport (advection and diffusion) and the chemical reactions between pollutants. To solve the transport of pollutants, it uses an Eulerian description and proposes different schemes based on Finite Volume method. Specifically, the Piecewise Parabolic Method [Colella and Woodward, 1984], the Bott scheme [Bott, 1989], and the Yamartino-Blackman cubic scheme [Yamartino, 1993].

As shown in Fig. 2.2, CMAQ is also composed by various processors in order to incorporate the data from meteorological and emission systems. The Meteorology-Chemistry Interface Processor (MCIP) can process data from the fifth-generation Penn State/NCAR Mesoscale Model (MM5) [Grell et al., 1994], as well as, from the Weather Research and Forecasting (WRF) Model's Advanced Research WRF (ARW) core [Skamarock et al., 2008].

On top of the processors, there is the CMAQ Chemical Transport Model [Byun and Ching, 1999, Byun and Schere, 2006], and [CMAQ Operational Guide, 2012]. Here, the most relevant modules are described briefly

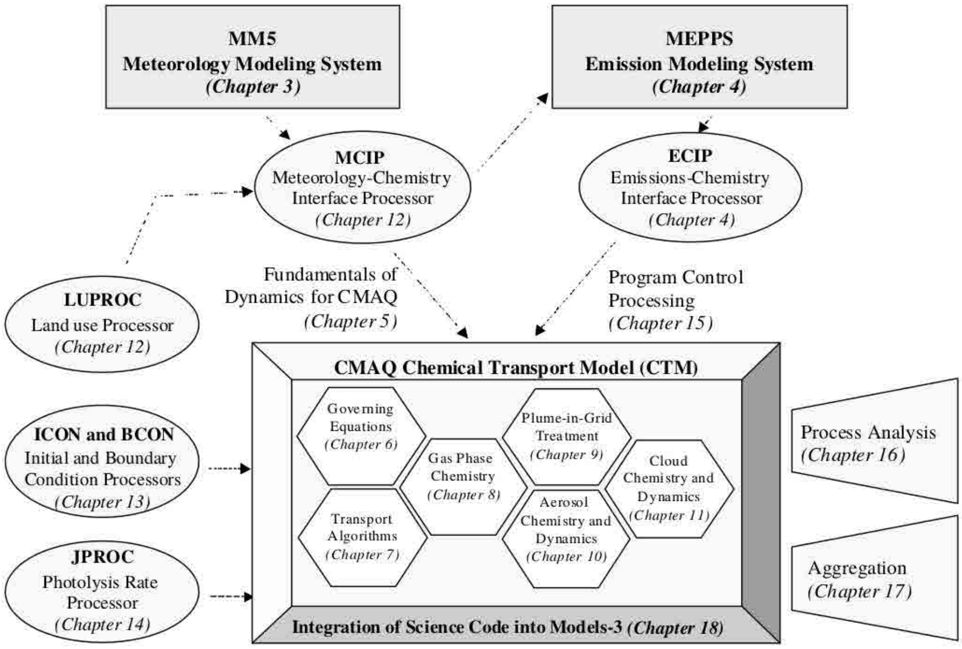


Figure 2.2: CMAQ diagram [Byun and Ching, 1999]

- *Advection and Diffusion.* As stated previously, there are several methods to solve the advection and the diffusion of the pollutants. Horizontal and vertical diffusion is treated separately with different models.

- *Gas Phase Chemistry.* This module is the responsible for the chemistry of the model. It includes the CB05 chemical mechanism. The module also includes different solvers, the Sparse Matrix Vectorized Gear (SMVGEAR) algorithm, the Rosenbrock (ROS3) solver, and the Euler Backward Iterative (EBI) method.

- *Plume-in-Grid (PinG) Modelling* CMAQ includes algorithms to treat emissions from major emitters. The PinG simulates the plume rise, transport and chemical reactions of the emission.

In this thesis we provide a new strategy to simulate emissions from majors emitters. The main objective of the thesis is to provide a system comparable to PinG that can be useful in zones with complex orography. Comparing to PinG, this system does not return the simulation back to the CMAQ system, and its purpose is to provide a model capable to predict the air quality in the local scale.

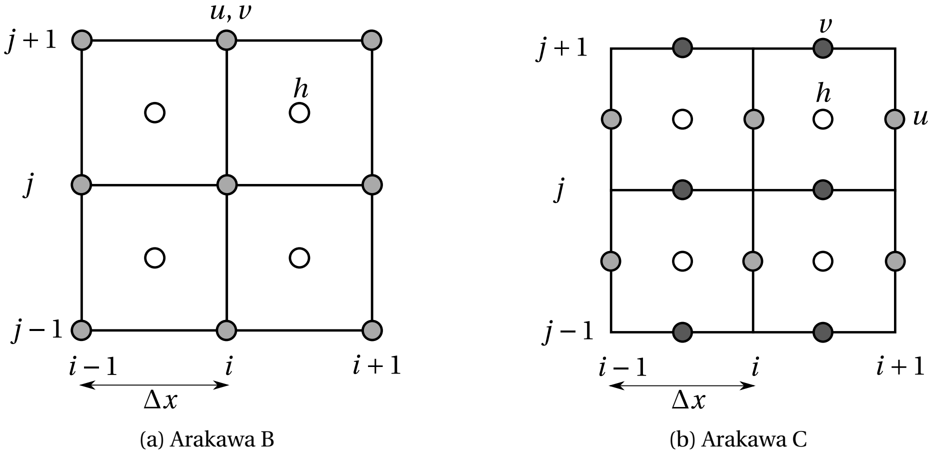
The diffusion modules as well as the CB05 chemical mechanism [Yarwood et al., 2005] used in CMAQ will be also used in the local scale model presented in this thesis.

2.2.1 Spatial discretization and data organization in CMAQ

To use the data from CMAQ into the local scale model, we need to interpolate the results from the mesoscale model into the local model.

The Meteorology-Chemistry Interface Processor (MCIP) [Otte and Pleim, 2010] is the processor that interpolates the values from the numerical weather prediction model to the CTM. This module defines how data is stored and used in CMAQ system. CMAQ differentiates between horizontal coordinates and vertical coordinates.

The Models-3 CMAQ system uses square cells for the horizontal grid representation, this restriction arises from the fractional time splitting approach that requires the use of square grid to maintain the accuracy of the algorithms consistently. The horizontal position of the data depends on the need of describing it; Arakawa described five different ways to locate the data in orthogonal grids. MM5 uses Arakawa B horizontal staggering (Fig. 2.3a), and both the CMAQ CTM and WRF-ARW uses Arakawa C horizontal staggering (Fig. 2.3b).



(a) Arakawa B

(b) Arakawa C

Figure 2.3: Arakawa grids

Figure 2.3 shows the two grids used in this problem. The staggered Arakawa B horizontal grid uses corners for the horizontal velocity (u, v), with scalar variables (e.g. h) evaluated at grid centres, while the Arakawa C horizontal staggered grid separates the evaluation of horizontal velocities, instead of evaluating both u and v at the grid corners, u is evaluated at the centre of the x-axis faces, and v is evaluated in the centre of y-axis faces.

CMAQ uses contravariant wind components instead of the regular wind components. Each contravariant velocity is located at the centre faces of the horizontal grid following Arakawa C scheme. When MM5 is used as the meteorological system, MCIP also writes the wind components at the grid corners of the horizontal grid following Arakawa B scheme.

MCIP writes the output variables in two different files. The variables that are located in the grid centre of the Arakawa grid are written in “CRO” files, and the rest of the variables are located in “DOT” files. “DOT” files require array sizes larger than “CRO” files.

Once the horizontal distribution is explained, we need to know the vertical one. Vertical layers use sigma-pressure coordinates, but CMAQ uses a monotonically increasing vertical coordinate, so the vertical coordinate $(\hat{x}^3)$ is defined as $\hat{x}^3 = 1 - \sigma_p$.

Sigma-pressure coordinates can be time dependent or time independent depending on the meteorological model used. If the meteorological model is hydrostatic, the sigma-pressure coordinate is time dependent. When the meteorological model is non-hydrostatic the sigma-pressure coordinates are constant.

The definition of the sigma-pressure is given by the following equation

$$\sigma_{\tilde{p}} = \frac{\tilde{p} - p_T}{\tilde{p}_s - p_T} = \frac{\tilde{p} - p_T}{\tilde{p}^*} \tag{2.1}$$

where p_t is the pressure at the model top, which is constant, $\tilde{p}_s(\hat{x}^1, \hat{x}^2, t)$ is the surface pressure, and $\tilde{p}(\hat{x}^1, \hat{x}^2, \hat{x}^3, t)$ is the hydrostatic pressure. We define $\tilde{p}^* = \tilde{p}_s - p_T$. The pressure $\tilde{p}$ is defined as

$$\frac{\partial \tilde{p}}{\partial z} = -\tilde{\rho} g \tag{2.2}$$

And the Jacobian is defined as

$$J_{\hat{x}^3} \equiv \left| \frac{\partial \hat{x}^3}{\partial z} \right|^{-1} = \left| -\frac{1}{\tilde{p}^*} \frac{\partial \tilde{p}}{\partial z} \right|^{-1} = \frac{\tilde{p}^*}{\tilde{\rho} g} \tag{2.3}$$

If the meteorological model is hydrostatic surface pressure (in $\tilde{p}^*$) and air density ($\tilde{\rho g}$) are not constant, so the vertical coordinates are time dependent, in the case of a non-hydrostatic model both variables are constant so the vertical coordinates are time independent.

To calculate the value of the height of a point above the surface level (z) from sigma pressures a simple description of reference temperature profile is needed. The temperature profile for the non-hydrostatic version of MM5 is defined as

$$\tilde{T}(z) = \tilde{T}_s + A \ln\left(\frac{\tilde{p}(z)}{\tilde{p}_s}\right) \tag{2.4}$$

where $\tilde{T}_s$ is the surface temperature in K, $\tilde{p}_s$ is the surface pressure, and A is the atmospheric lapse rate with a typical value of 50 K. With equations (2.4), (2.2) and the equation of state we can relate the reference pressure in terms of the height (z) as

$$\frac{d \ln \tilde{p}(z)}{dz} = -\frac{g}{R} \frac{1}{\left[\tilde{T}_s - A \ln(\tilde{p}_s)\right] + A \ln\left(\tilde{p}(z)\right)} \tag{2.5}$$

The solution of this equation is a quadratic one, we will use the positive square root term

$$\ln\left(\tilde{p}(z)\right) = \frac{-b + \sqrt{b^2 - 2Ac}}{A} \tag{2.6}$$

where $b = \tilde{T}_s - A\left(\ln \tilde{p}_s\right)$, and $c = \frac{g}{R} z - \left(\frac{1}{2} A (\ln \tilde{p}_s)^2 + b(\ln \tilde{p}_s)\right)$. With this result you can compute the height above the surface level.

The vertical grid is composed by layers. This layers have data in the top of the layer (full-layer) and at the middle of the layer (half-layer). The values at the terrain level are not in a layer but in a different file with just a horizontal grid when necessary (wind velocities at terrain level are zero). Figure 2.4 shows all the possible positions in a CMAQ grid.

So, MCIP writes the variables in different files depending on the horizontal and the vertical location of data. As noted before, data located at the centre of the Arakawa grid is in a file named “CRO”, while data located at corners and faces is written in “DOT” files. Regarding the vertical coordinates, variables only needed at terrain are written in “2D” files and variables described in all the layers (in half-layer or full-layer position) are written in “3D” files. Finally, those variables that are time independent are written in “GRID” files, and time dependent variables are written in “MET” files. Tables 2.1 and 2.2 describe the variables that are processed by MCIP and the file where they are written. (Table 2.1 define time independent variables, and Tab. 2.2 time dependent variables).

Besides the meteorological data, the local scale also needs concentration data. CMAQ CTM uses the same grid for the output of the simulation than the described for MCIP. Concentration data is located at the centre points of the grid, so the number of columns and rows will be the same as the MCIP “CRO” files.

CMAQ CTM writes five output files (other output files can be written optionally). Specifically

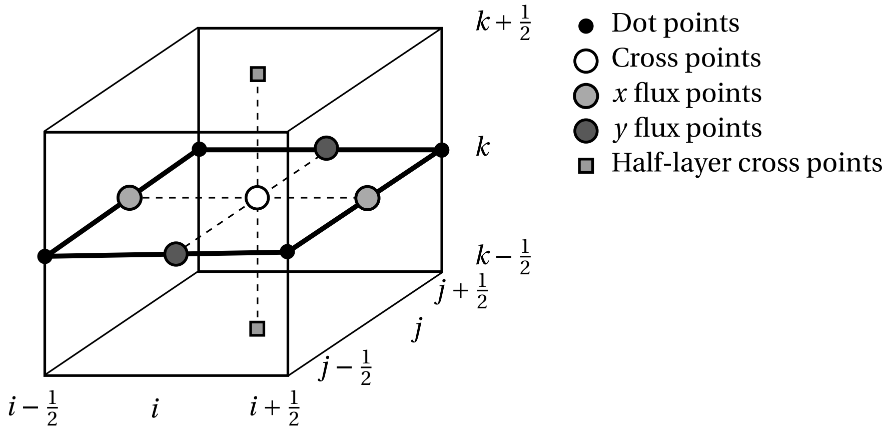


Figure 2.4: Position of variables in a portion of the grid

Field	Name	Units	File(s)
LAT	latitude (cell corners)	°	GRIDDOT2D
LON	longitude (cell corners)	°	GRIDDOT2D
MSFD2	map-scale factor squared (cell corners)	$m^2\,m^{-2}$	GRIDDOT2D
LAT	latitude (cell centers)	°	GRIDCRO2D, GRIDBDY2D
LON	longitude (cell centers)	°	GRIDCRO2D, GRIDBDY2D
MSFX2	map-scale factor squared (cell centers)	$m^2\,m^{-2}$	GRIDCRO2D, GRIDBDY2D
HT	terrain elevation	m	GRIDCRO2D, GRIDBDY2D
DLUSE	dominant land use	category	GRIDCRO2D, GRIDBDY2D
LWMASK	land-water mask	category	GRIDCRO2D, GRIDBDY2D
PURB[a]	percentage of urban area	%	GRIDCRO2D, GRIDBDY2D
LUFRAC_XX[a] ,[b]	land use fraction by category	fraction	GRIDCRO2D, GRIDBDY2D
X3HT0F[c]	height of layer face (top) above ground level	m	GRIDCRO3D
X3HT0M[c]	height of layer middle above ground level	m	GRIDCRO3D

[a] Only output if fractional land use fields are provided in the meteorological input file

[b] Output for XX land use categories from 1 to NN , where NN is the number of categories in the classification system used by the meteoro- logical model

[c] Only output for models with time-invariant reference layer heights (e.g., currently only for MM5 and not WRF)

Table 2.1: MCIP time independent output

- *CONC* file includes instantaneous model species concentrations at the end of each model hour. It contains gas-phase species mixing ratios (ppmV) and aerosol species concentrations ($\mu g\,m^{-3}$). The species that are written to this file are defined in CONC.EXT include file.
- *CGRID* file includes model species concentrations at the end of each simulation period. This file can be used to restart the simulation from a given moment. For this reason, this file contains all the species required by the chemical mechanism and aerosol model configurations that have been used.
- *ACONC* file includes average model species concentrations at the end of each model hour. The species written to the ACONC file are set by the user in the CCTM run script using the variable AVG_CONC_SPCS.
- *DRYDEP* includes cumulative hourly dry deposition fluxes ($kg\,ha^{-1}$) for selected model species.

Field	Name	Units	File(s)
PRSFC	surface pressure	Pa	METCRO2D
USTAR	friction velocity	$\mathrm{m\,s^{-1}}$	METCRO2D
WSTAR	convective velocity scale	$\mathrm{m\,s^{-1}}$	METCRO2D
PBL	planetary boundary layer height	m	METCRO2D
ZRUF	surface roughness length	m	METCRO2D
MOLI	inverse of Monin-Obukhov length	$\mathrm{m^{-1}}$	METCRO2D
HFX	sensible heat flux	$\mathrm{W\,m^{-2}}$	METCRO2D
QFX	latent heat flux	$\mathrm{W\,m^{-2}}$	METCRO2D
RADYNI	inverse of aerodynamic resistance	$\mathrm{m\,s^{-1}}$	METCRO2D
RSTOMI	inverse of stomatal resistance	$\mathrm{m\,s^{-1}}$	METCRO2D
TEMPG	skin temperature at ground	K	METCRO2D
TEMP2	temperature at 2 m a.g.l.	K	METCRO2D
Q2	water vapor mixing ratio at 2 m a.g.l.	$\mathrm{kg\,kg^{-1}}$	METCRO2D
WSPD10	wind speed at 10 m a.g.l.	$\mathrm{m\,s^{-1}}$	METCRO2D
WDIR10	wind direction at 10 m a.g.l.	°	METCRO2D
GLW	longwave radiation at ground	$\mathrm{W\,m^{-2}}$	METCRO2D
GSW	shortwave radiation absorbed at ground	$\mathrm{W\,m^{-2}}$	METCRO2D
RGRND	shortwave radiation reaching ground	$\mathrm{W\,m^{-2}}$	METCRO2D
RN	non-convective precipitation over interval	cm	METCRO2D
RC	convective precipitation over interval	cm	METCRO2D
CFRAC	total cloud fraction	fraction	METCRO2D
CLDT	cloud top layer height	m	METCRO2D
CLDB	cloud bottom layer height	m	METCRO2D
WBAR	average liquid water content of cloud	$\mathrm{g\,m^{-3}}$	METCRO2D
SNOCOV	snow cover	non-dimen	METCRO2D
VEG	vegetation coverage	fraction	METCRO2D
LAI	leaf-area index area	$\mathrm{area^{-1}}$	METCRO2D
WR[a]	canopy moisture content	m	METCRO2D
SOIM1[a]	soil moisture in near-surface soil	$\mathrm{m^3\,m^{-3}}$	METCRO2D
SOIM2[a]	soil moisture in deep soil	$\mathrm{m^3\,m^{-3}}$	METCRO2D
SOIT1[a]	soil temperature in near-surface soil	K	METCRO2D
SOIT2[a]	soil temperature in deep soil	K	METCRO2D
SLTYP[a]	soil type	category	METCRO2D
VD_species[b]	dry deposition velocity for species	$\mathrm{m\,s^{-1}}$	METCRO2D
JACOBF	total Jacobian at layer face	m	METCRO3D, METBDY3D
JACOBM	total Jacobian at layer middle	m	METCRO3D, METBDY3D
DENSA_J	Jacobian-weighted air density	$\mathrm{kg\,m^{-2}}$	METCRO3D, METBDY3D
WHAT_JD	Jacobian- and density-weighted contravariant vertical velocity	$\mathrm{kg\,m^{-1}\,s^{-1}}$	METCRO3D, METBDY3D
TA	air temperature	K	METCRO3D, METBDY3D
QV	water vapor mixing ratio	$\mathrm{kg\,kg^{-1}}$	METCRO3D, METBDY3D
PRES	pressure	Pa	METCRO3D, METBDY3D
DENS	density of air (MM5: total density; WRF: dry density)	$\mathrm{kg\,m^{-3}}$	METCRO3D, METBDY3D
WWIND	vertical velocity	$\mathrm{m\,s^{-1}}$	METCRO3D, METBDY3D
ZH	mid-layer height a.g.l.	m	METCRO3D, METBDY3D
ZF	full-layer height a.g.l.	m	METCRO3D, METBDY3D
QC	cloud water mixing ratio	$\mathrm{kg\,kg^{-1}}$	METCRO3D, METBDY3D
QR	rain water mixing ratio	$\mathrm{kg\,kg^{-1}}$	METCRO3D, METBDY3D
QI[c]	ice mixing ratio	$\mathrm{kg\,kg^{-1}}$	METCRO3D, METBDY3D
QS[c]	snow mixing ratio	$\mathrm{kg\,kg^{-1}}$	METCRO3D, METBDY3D
QG[c]	graupel mixing ratio	$\mathrm{kg\,kg^{-1}}$	METCRO3D, METBDY3D
TKE[c] or TKEF[c]	turbulent kinetic energy	$\mathrm{J\,kg^{-1}}$	METCRO3D, METBDY3D
UWIND	U-component wind (cell corners)	$\mathrm{m\,s^{-1}}$	METDOT3D
VWIND	V-component wind (cell corners)	$\mathrm{m\,s^{-1}}$	METDOT3D
UHAT_JD	contravariant U-component wind×density×Jacobian	$\mathrm{kg\,m^{-1}\,s^{-1}}$	METDOT3D
VHAT_JD	contravariant V-component wind× densit×Jacobian	$\mathrm{kg\,m^{-1}\,s^{-1}}$	METDOT3D

[a] Only output if fields are available from land-surface model in input meteorological model fields

[b] Optional output controlled by run-time user option that generates fields for 31 species

[c] Only available if output by meteorological model

Table 2.2: MCIP time dependent output

It is a “2D” file.

- *WETDEP* includes cumulative hourly wet deposition fluxes (kg ha^{-1}) for selected model species. It is a “2D” file.

2.3 Interpolation from CMAQ to the local scale model

With all the data available from the CMAQ system we can interpolate it to the local scale model. The local scale model is a nested domain in the CMAQ simulation. This interpolation would be straightforward if the structure of both meshes were the same, but the local scale model uses tetrahedral meshes where there is no grid structure. To solve this limitation the next interpolation is proposed.

The interpolation consists in two different steps, the first step is an horizontal interpolation and the second one is a vertical interpolation.

To interpolate some variable in a point $P^* = \{x^*, y^*, z^*\}$ located inside the CMAQ grid mesh, we will follow the algorithm presented in Alg. 2.2.

Algorithm 2.2 CMAQ interpolation

```
1 Locate the CMAQ mesh pillar where $P^*$ lies
2 For each CMAQ layer, compute the height $(z)$ in the position $\{x^*, y^*\}$ using equation (2.6)
3 Locate the cell where $P^*$ lives
4 Interpolate the variable of interest in the two layers that define the cell.
5 Interpolate the variables in the vertical axis
```

The location of the point P^* in the CMAQ mesh is computed taking advantage of the grid structure of the mesh. This location is found using the parametric coordinates of the point in the quadrilateral; the method described by Hua [1990] will be used for this goal. So, the point P^* is located in the horizontal cell n defined between (i^*, i^*+1) and $\left(j^*, j^*+1\right)$ and has horizontal parametric coordinates (u^*, v^*).

To compute the height of all the layers we need to interpolate the skin temperature (TEMPG), the surface reference pressure (PRSFC), and a value of the pressure at the top layer. This interpolation is a bilinear interpolation using the parametric coordinates found in the location stage. Specifically, we use the following shape functions

$$h_1(r,s) = \frac{(1-r)(1-s)}{4} \tag{2.7}$$

$$h_2(r,s) = \frac{(1+r)(1-s)}{4} \tag{2.8}$$

$$h_3(r,s) = \frac{(1+r)(1+s)}{4} \tag{2.9}$$

$$h_4(r,s) = \frac{(1-r)(1+s)}{4} \tag{2.10}$$

Evaluating them at point (u^*, v^*) and defining V_i as the value of the variable to interpolate in the i^{th} vertex of the horizontal grid, The value of the interpolated variable (V^*) is defined as

$$V^*(u^*, v^*) = \sum_{i=1}^{4} h_i(u^*, v^*) V_i \tag{2.11}$$

Interpolating all the necessary variables and computing the height using equation (2.6), we know the height z_k^* for all the layers k of the grid.

With the height of all the layers, we can find out the two layers that enclose the point ($z_k^* <= z^* <= z_{k+1}^*$), and the height differences between the bottom and the top local layers. To compute the value of a variable whose values in the bottom and top layers are V_k^* and V_{k+1}^* respectively, the following linear interpolation will be used

$$V^*(x^*, y^*, z^*) = \frac{z^* - z_k^*}{z_{k+1}^* - z_k^*} V_k^* + \frac{z_{k+1}^* - z^*}{z_{k+1}^* - z_k^*} V_{k+1}^* \tag{2.12}$$

Now that we know the cell where P^* is located and we have computed the parametric coordinates and the vertical interpolation weights, we can interpolate any variable by first computing its value in the top and bottom layer, and finally interpolating linearly between both values.

3 Automatic tetrahedral mesh generation

In this Chapter we describe the main steps for the construction of an adaptive tetrahedral mesh of the three-dimensional domain of interest. The main goal is to obtain an adaptive mesh that captures the orographic information of a complex terrain within a given precision, and the possibility to insert anthropomorphic entities such as stacks or buildings.

Some mesh generators for this kind of problems exists. For example, the meccano method [Montenegro et al., 2009, 2010, Cascón et al., 2013] is suitable for generating a mesh from a triangulation of the terrain (e.g. a digital terrain model). To mesh a volume of the atmosphere, the meccano mesh generates a cuboid that resembles the dimensions of that volume. Then, the triangulation of the terrain is mapped to the bottom face of the cuboid using the mean value parameterization proposed by Floater [2003]. Vertical and top walls are also mapped. Then, the cuboid is split in an specific number of cubes, that are divided into six tetrahedra compatible with the Kossaczký refinement [Kossaczký, 1994]. Then the mesh is locally refined to obtain an approximation of the terrain within a given precision. The distance between the mesh and the triangulation of the terrain will be computed using the mapping. Once the precision is reached an structured mesh has been created in the cuboid. Using the mappings from the cuboid faces to the volume faces, all the boundary nodes are located in the physical space boundary. Finally, using mesh optimization [Escobar et al., 2003] the inner nodes are located in the physical space. This method generates a very high quality mesh for this kind of geometries and constructs a volumetric parameterization between the physical volume and the parametric cuboid, but the use of the Kossaczký refinement creates a very rigid structure that can be problematic for certain applications.

To avoid this rigid structure two different approaches are presented. These methods allow to construct a mesh with an structure that follows the layers of the CMAQ.

The first proposed method is based on the generation of a two-dimensional mesh adapted to the terrain using a refinement-coarsening algorithm, the top surface of the volume is meshed with a regular mesh, then a vertical distribution of nodes is generated, finally the whole domain is tetrahedralized using the Delaunay algorithm.

The second method is very similar to the first one, but two fundamental differences exists. The first one is the adaptation method, instead of a refinement-coarsening algorithm the size of the elements will be based on the curvature of the terrain. The second difference is the construction of the vertical distribution. In this method, a number of layers (that can match the CMAQ layers) are considered. This layers are located using an smoothing algorithm to have the terrain layer at the bottom and a flat layer at the top. This distribution coupled with the curvature based element size provides a mechanism to fix the structure of the mesh.

3.1 Refinement-coarsening strategy

This method is presented in algorithm 3.1. It consists in five different phases. In the first stage we construct a triangular mesh adapted to the terrain, in the second phase we create a vertical distribution of the nodes in the three-dimensional volume, in the third stage the nodes in the volume are generated, the fourth phase constructs a three-dimensional mesh and finally, in the fifth stage, this mesh is optimized to untangle the inverted elements and ensure a good mesh quality.

Algorithm 3.1 Refinement-coarsening mesh generator

```
1: Adaptive discretization of the terrain surface
   1.1: Regular refinement until capturing all the details
   1.2: Coarsening of the unnecessary elements of the mesh
2: Vertical spacing distribution of the nodes
3: Determination of the set of points
   3.1: Generate a flat mesh for the top of the mesh
   3.2: The flat mesh computed in 3.1 is refined and coarsened to be compatible with the terrain mesh
   3.3: A set of points is generated between both meshes taking into account the vertical spacing distribution computed in 2
4: Three-dimensional mesh generation
   4.1: Recreate the point distribution in a cuboid and generate a mesh using a Delaunay algorithm
   4.2: Remap the points and connectivities of the cuboid into the physical domain
5: Mesh optimization
```

3.1.1 Adaptive discretization of the terrain surface

The three-dimensional mesh generation process starts by fixing the nodes placed on the terrain surface. Their distribution must be adapted to the orography to minimize the number of required nodes. First, we construct a sequence of nested meshes $T = \{\tau_1 < \tau_2 < \cdots < \tau_m\}$ from a uniform triangulation τ_1 of the rectangular area under consideration. The τ_j level is obtained by previous level τ_{j-1} using the 4-T Rivara refinement algorithm [Rivara, 1987]. All triangles of the τ_{j-1} level are divided into four sub-triangles by introducing a new node in the centres of each edge and connecting the node introduced on the longest side with the opposite vertex and with the other two introduced nodes. Finally, a new sequence $T' = \{\tau_1 < \tau'_2 < \cdots < \tau'_{m'}\}$, $m' \leq m$, is constructed by applying the coarsening algorithm; details may be seen in Ferragut et al. [1994], Plaza et al. [1996]. In this step we consider the coarsening parameter ε that fixes the precision to approximate the terrain topography. The difference in absolute value between the resulting heights at any point of the triangulation $\tau'_{m'}$ and its corresponding real height (interpolated from a digital elevation map) will be less than the tolerance ε.

3.1.2 Vertical spacing function

We will only use the information of node locations and their corresponding proper levels of T' to generate the set of nodes inside the domain.

We are interested in generating a set of points with higher density in the area close to the terrain. Thus, every node is to be placed in accordance with the function $z_i = a\,i^{\alpha} + b$, where z_i height corresponds to the i^{th} inserted point, in such a way that for $i = 0$ the height of the terrain is obtained, and for $i = n$ we get the height of the last introduced point. This last height must coincide with the altitude h of the upper plane that bounds the domain. The exponent α controls the concentration of points near the terrain surface, being greater as the exponent increases. The lowest value of alpha is one, corresponding a equally spaced distribution. In these conditions the number of points defined over the vertical is $n+1$. The z_i height can be written, for $i = 1,2,\dots,n$, in terms of the height of the previously inserted point

$$z_i = z_{i-1} + \frac{h - z_{i-1}}{n^\alpha - (i-1)^\alpha}\left[i^\alpha - (i-1)^\alpha\right] \tag{3.1}$$

In (3.1), once the values of α and n are fixed, the points to insert are completely defined. To maintain acceptable minimum quality of the generated mesh, the distance between the first inserted point ($i = 1$) and the surface of the terrain could be fixed. This will reduce the two degrees of freedom to one, either α or n. Consider the value of the distance d as a determined one, such that $d = z_1 - z_0 = \frac{h-z_0}{n^\alpha}$. If we fix α and set free the value of n we obtain

$$n = \left(\frac{h - z_0}{d}\right)^{1/\alpha} \tag{3.2}$$

Conversely, if we fix the value of n and set α free, we get

$$\alpha = \frac{\log \frac{h-z_0}{d}}{\log n} \tag{3.3}$$

In both cases, given one of the parameters, the other may be calculated by the expression (3.2) or (3.3), respectively. In this way, the point distribution on the vertical respects the distance d between z_1 and z_0. Moreover, if the distance between the last two introduced points is fixed, that is, $D = z_n - z_{n-1}$, then the parameters α and n are perfectly defined.

3.1.3 Determination of the set of points

The point generation will be carried out in three stages. In the first stage, we define a regular two-dimensional mesh τ_1 for the upper boundary of the domain with the required density of points. In the second phase, the mesh τ_1 will be globally refined and subsequently coarsened to obtain a two-dimensional mesh $\tau'_{m'}$ capable of fitting itself to the topography of the terrain. This last mesh defines the appropriate node distribution above ground level. In the third stage, we generate the set of points distributed between the upper boundary and the terrain surface. In order to do this, some points will be placed over the vertical of each node P of the terrain mesh $\tau'_{m'}$, attending to the vertical spacing function and to level j ($1 \le j \le m'$) where P is proper, i.e. the level of τ'_j where the node P was introduced. The vertical spacing function will be determined by the strategy used to define the following parameters: the topographic height z_0 of P; the altitude h of the upper boundary; the maximum possible number of points $n+1$ in the vertical of P, including both P and the corresponding upper boundary point, if there is one; the degree of the spacing function α; the distance between the two first generated points $d = z_1 - z_0$; and the distance between the two last generated points $D = z_n - z_{n-1}$. Thus, the height of the i^{th} point generated over the vertical of P is given by (3.1) for $i = 1,2,\dots,n-1$.

Regardless of the defined vertical spacing function, we shall use level j where P is proper to determine the definitive number of points generated over the vertical of P excluding the terrain and the upper boundary. We shall discriminate among the following cases:

1. If $j = 1$, that is, if node P is proper of the initial mesh τ_1, nodes are generated from (3.1) for $i = 1, 2, \ldots, n-1$.

2. If $2 \leq j \leq m' - 1$, we generate nodes for $i = 1, 2, \ldots, min(m' - j, n - 1)$.

3. If $j = m'$, that is, node P is proper of the finest level $\tau'_{m'}$, then any new node is generated.

This process has its justification, as mesh $\tau'_{m'}$ corresponds to the finest level of the sequence of nested meshes $T' = \{\tau_1 < \tau'_2 < \cdots < \tau'_{m'}\}$, obtained by the refinement/coarsening algorithm. Thus the number of introduced points decreases smoothly with altitude, and they are also efficiently distributed in order to build the three-dimensional mesh in the domain.

We set out a particular strategy where values of α and n are automatically determined for every point P of $\tau'_{m'}$, according to the size of the elements closest to the terrain and to the upper boundary of the domain. This strategy approximately respects both the required distances between the terrain surface and the first layer and the imposed distance between the last virtual layer and the upper boundary.

3.1.4 Three-dimensional mesh generation

Once the set of points has been defined, it will be necessary to build a three-dimensional mesh able to connect the points in an appropriate way and which conforms with the domain boundary, i.e., a mesh that respects every established boundary.

Although Delaunay triangulation is suitable to generate finite element meshes with a high regularity degree for a given set of points, this does not occur in the problem of conformity with the boundary, as it generates a mesh of the convex hull of the set of points. It may be thus usually impossible to recover the domain boundary from the faces and edges generated by the triangulation. To avoid this, we have two different sorts of techniques: *conforming Delaunay triangulation* [Murphy et al., 2000] and *constrained Delaunay triangulation* [George et al., 1991]. The first alternative is inadequate for our purpose, as we wish the resulting mesh to contain certain predetermined points. The second alternative could provide another solution, but it requires quite complex algorithms to recover the domain boundary [Chen et al., 2011].

To build the three-dimensional Delaunay triangulation of the domain points, we propose to start by resetting the points in an auxiliary parallelepiped, so that every point of the terrain surface is on the original coordinates x, y, but at an altitude equal to the minimum terrain height, z_{min}. In the upper plane of the parallelepiped we set the nodes of level τ_1 of the mesh sequence at altitude h. The triangulation of this set of points is done using a variant of Bowyer-Watson incremental algorithm [Escobar and Montenegro, 1996] that effectively solves the problems derived from the round-off errors made when working with floating point numbers.

Once the triangulation is built in the parallelepiped, the final mesh is obtained by re-establishing its original heights. This latter process can be understood as a compression of the global mesh defined in the parallelepiped, such that its lowest plane becomes the terrain surface. In this way, conformity is ensured.

Sometimes when re-establishing the position of each point to its real height, poor quality, or even inverted elements may occur. For this reason, we need a procedure to untangle and smooth the resulting mesh, as analysed in the next phase.

3.1.5 Mesh optimization with improved objective functions

In finite element simulation the mesh quality is a crucial aspect for good numerical behaviour of the method. The most usual techniques to improve the quality of a valid mesh, that is, one that does not have inverted elements, are based upon local smoothing. In short, these techniques consist of finding the new positions that the mesh nodes must hold, in such a way that they optimize an objective function. Such a function is based on a certain measurement of the quality of the local submesh, $N(v)$, formed by the set of tetrahedra connected to the *free node v*. The quality of an element is defined using several measures, we will use the shape distortion measure presented in Knupp [2001]. An ideal element E^I with the desired shape is defined, then the distortion between this element and the physical element E^P is computed. To compute the distortion we consider the unique affine mapping φ_E represented in Fig. 3.1.

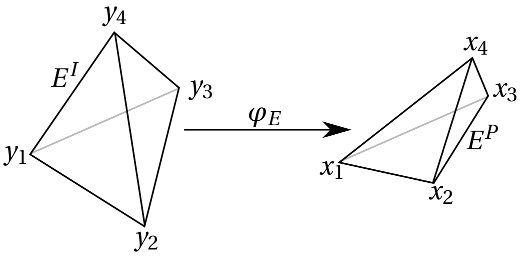


Figure 3.1: Mapping between the ideal element and the physical element

Being φ_E affine, its Jacobian J_{φ_E} is constant, so any change in size, rotation, or translation of the physical element does not change the value of the Jacobian. This property cause Jacobian to be a perfect candidate to be used in distortion measures, specifically the shape distortion measure [Knupp, 2001] is defined as

$$\eta(J_{\varphi_E}) = \frac{\|J_{\varphi_E}\|^2}{3|\sigma|^{\frac{2}{3}}} \tag{3.4}$$

where $\|\cdot\|$ is the Frobenius norm, and $\sigma = \det(J_{\varphi_E})$. This distortion measure has a value in the range $[1,\infty)$ where $\eta = 1$ is the ideal element. For this reason quality is defined as

$$q = \frac{1}{\eta} \tag{3.5}$$

being $q = 1$ the ideal element, and $q = 0$ a very poor quality element.

After repeating a local optimization process several times for all the nodes of the current mesh, quite satisfactory results can be achieved. Usually, objective functions are appropriate to improve the quality of a valid mesh, but they do not work properly when there are inverted elements. This is because they present singularities (barriers) when any tetrahedron of $N(v)$ changes the sign of its Jacobian determinant. To avoid this problem an optimization method consisting of two stages is proposed by Freitag and Plassmann [2000] and Freitag and Knupp [2002]. In the first one, the possible inverted elements are untangled by an algorithm that maximizes their negative Jacobian determinants; in the second one, the resulting mesh from the first stage is smoothed using another objective function based on a quality metric of the tetrahedra of $N(v)$.

We have used an alternative to this procedure [Escobar et al., 2003], such that the untangling and smoothing are carried out in the same stage. For this purpose, we use a suitable modification of the objective function such that it is regular all over $\mathbb{R}^3$. The regularization consists in replacing σ in equation (3.4) by

$$h(\sigma) = \frac{1}{2}\left(\sigma + \sqrt{\sigma^2 + 4\delta^2}\right) \tag{3.6}$$

where δ is a numerical parameter that has to be determined.

When a feasible region (subset of $\mathbb{R}^3$ where v could be placed, being $N(v)$ a valid submesh) exists, the minima of the original and modified objective functions are very close and, when this region does not exist, the minimum of the modified objective function is located in such a way that it tends to untangle $N(v)$. The latter occurs, for example, when the fixed boundary of $N(v)$ is tangled. With this approach, we can use any standard and efficient unconstrained optimization method to find the minimum of the modified objective function, see for example Bazaraa et al. [1993].

In conclusion, by using the modified objective function, we can untangle the mesh and, at the same time, improve its quality. More details about this mesh optimization procedure can be seen in Escobar et al. [2003].

3.2 Curvature based element size

In this approach, instead of using a refinement algorithm to adapt the mesh to the terrain, we are using an element size background mesh. The element size will be computed based on the curvature of the terrain. Using this strategy we can insert a layer at a certain height and control its element size with the curvature of this layer. Once all the layers are meshed we will tetrahedralize the whole volume.

To mesh the two-dimensional layers we have used Ez4u mesh generation environment [Roca et al., 2010, 2007, Roca, 2009]. Ez4u is a software developed at Laboratori de Càlcul Numèric (LaCàN). To mesh the three-dimensional domain we have used the Delaunay-based tetrahedral mesh generator Tetgen [Si, 2015].

This method can be divided in three different stages. First we will construct all the layers required, then each layer will be meshed, and finally the three-dimensional mesh will be generated. The insertion of elements in the mesh is also explained. Algorithm 3.2 describes the most important steps of this method.

Algorithm 3.2 Curvature based element size mesh generator

```
1: Construction of the layers
  1.1: Terrain triangulation
  1.2: Creation of the layers elevating and smoothing the curvature of the terrain
2: Layer mesh generation
  2.1: Creation of the background mesh using a curvature based element size
  2.2: Meshing of each layer using Ez4u
3: Three-dimensional mesh generation
  3.1: Creation of a piecewise linear complex (PLC) from all the layers
  3.2: Three-dimensional meshing of the PLC using a Delaunay-based algorithm
```

3.2.1 Construction of the layers

First of all we need to construct the stack of layers that will determine the volume of the mesh. The number of layers and its height are chosen by the user. When the mesh is used with CMAQ data is a good idea to mimic its layer distribution. So, we can define n_layers as the number of layers, and H^j the mean height of the layer $j \in [1, n\_layers]$.

In this stage we have to translate the terrain to the rest of the layers such that the curvature of the layers is smoothed successively. To better understand this translation we will split it in two parts.

First, we translate all the nodes of the terrain triangulation to a given layer j with the following formula

$$z_i^j = \max(z^1) + H^j \left(\frac{H^{n\_layers} - \max(z^1)}{H^{n\_layers}} \right) \qquad \forall i \in [0, n\_nodes] \tag{3.7}$$

being z_i^j the new z-coordinate of the i^{th} node in the layer j, n_nodes the number of nodes of the terrain triangulation, and $\max(z^1)$ is the maximum height of the terrain. This step just creates a planar surface in each layer.

In the second step, this planar surface is disturbed to smooth the curvature of the previous layer. This perturbation depends on the heights of the previous layer, so it has to be done sequentially one layer after the other starting at the bottom layer.

$$z_i^j = z_i^j - \left(\max(z^{j-1}) - z_i^{j-1} \right) \left(1 - \frac{z_i^j}{H^{n\_layers}} \right) \tag{3.8}$$

where z_i^{j-1} is the height of the node in the previous layer and $\max(z^{j-1})$ is the maximum height of the previous layer. In this step, the difference of the actual height for the i^{th} node in the previous layer and the maximum height of the previous layer is weighted with the ratio between this current layer height and the top layer height. This way, layers are flattened respectively until the top layer that is completely flat.

3.2.2 Layer mesh generation

To mesh the two-dimensional layers, the method used in this strategy consists in defining the element size of the mesh based on the curvature of the layers, and using a background mesh generate the resulting mesh.

A background mesh is a discretized grid that contains the element size desired in a geometry, the mesh generator considers this information to create a mesh adapted to this requirement; Persson [2006] presents the characteristics and usage of a background mesh in a mesh generator.

Unfortunately, the element sizes in a background mesh, sometimes, may conflict with themselves; i.e. asking for two very different sizes in an small region where just one of them fits. To solve this drawback the algorithm proposed by Ruiz-Gironés et al. [2014] preserves the smallest size during the mesh generation process.

Then, the first step is to construct the background mesh. To capture the terrain characteristics we will use the curvature, so the element size will be computed based on the curvature of the terrain following the formula proposed by Lee [2001].

The relation between the curvature and the size of the element is as stated in the following equation

$$l = \frac{1-\varepsilon}{\kappa} \sqrt{40\left(1 - (1 - 1.2\varepsilon)^{0.5}\right)} \tag{3.9}$$

where l is the size element, κ is the curvature value, $\varepsilon = (s - l)/l$ is the required precision, and s is the circumference arch. Figure 3.2 shows the geometrical relation between the curvature and the element size.

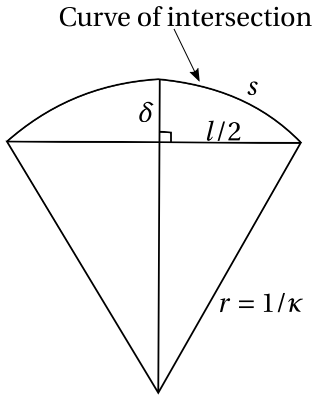


Figure 3.2: Geometric relation between element size and curvature

The curvature of the terrain is computed based on its triangulation. If the triangulation of the terrain comes from a uniform grid, e.g. a Digital Terrain Model (DTM), the curvature can be computed using a finite difference scheme, otherwise Garimella and Swartz [2003] presents several algorithms that can be applied to compute the curvature of the terrain.

Once the background mesh is computed, we will use the Ez4u mesh generation environment to create the two-dimensional mesh. The mesh generated in this step is a two-dimensional surface, so the domain to be meshed will be a flat surface covering all the desired space. The mesh generation algorithm used in Ez4u is a quadrilateral mesh generator that recursively divides the domain until the size of the background mesh is obtained [Sarrate and Huerta, 2000], then a smoothing technique is applied to improve the element quality [Sarrate and Huerta, 2001]. This quadrilateral mesh is converted to a triangular mesh by splitting each element into two triangles. This splitting is performed by inserting and edge from the vertex with the maximum angle of the element to its opposite vertex.

Once the mesh is generated in the flat domain, all the nodes heights will be relocated to the layers height by interpolating them.

At the end of this step all the layers are meshed with triangular meshes.

3.2.3 Three-dimensional mesh generation

Finally, we need to mesh the whole domain. For this goal we have used the Tetgen constrained Delaunay tetrahedralization mesh generator [Si, 2015].

Tetgen uses a simple model of a three-dimensional domain that may contain internal boundaries. The domain is described by a 3D piecewise linear complex (PLC) defined by a set of vertices, edges, polygons and polyhedra.

Following the definition of the PLC, all the three-dimensional domain is defined. The bottom and top layers define the top and bottom faces of the PLC. Depending on the internal structure of the final mesh, connectivities of the inner nodes will be inserted in the PLC. If connectivities are inserted, Tetgen will preserve the layers, otherwise Tetgen will preserve the inner layers nodes position but not the surface of the layers. Vertical walls can be defined by creating quadrilateral faces between layers containing all the boundary nodes.

Tetgen also let the user to define attributes to the nodes, and boundary markers to both nodes and faces. Attributes and markers allow to locate the new nodes and faces created by Tetgen. These markers will be very useful to define boundary conditions and to track some special elements. Tetgen also can refine adaptively the mesh defining a tetrahedral background mesh: a maximum volume on every tetrahedra of the mesh or an area on faces or a length in the edges; it also permits to insert nodes into an existing mesh. These facilities of Tetgen will prove very useful in next stages of the model, such as the plume rise perturbation (Chapter 5).

3.2.4 Insertion of anthropogenic elements

If some buildings or stacks have to be added to the domain, we need to change slightly the layer mesh generation step.

First of all, for all the layers that crosses the element, the domain of the two-dimensional mesh has to be cropped with the shape of the element. Then the element-size must be changed near this elements to take into account its dimensions. This correction of the element size has to be the same in all the layers that intersects the element.

This method will provide two-dimensional meshes with the shape of the element inserted into them, and the size adapted to its characteristics. The use of the same element-size will grant that the boundary of the element in all the layers will have the same number of vertices making it easier to sew the vertical walls.

To construct the top mesh an independent mesh has to be generated using the same element-size distribution as the background mesh. This will be a flat mesh that will be inserted in the PLC, and its boundary will be sewn with the boundary of the previous layer.

This method will allow to insert any number of elements inside the PLC before generating the three-dimensional mesh.

4 Wind field simulation

The wind field is the responsible of the convection of pollutants in the atmosphere, therefore its simulation is of paramount importance. In this Chapter, we describe a mass-consistent model that will be used to compute the wind field [Rodríguez, 2004]. A mass-consistent model forces null divergence for the wind field and requires impermeability in the terrain.

To solve the mass-consistent model we need a wind field in the whole domain. To obtain this wind field, we will interpolate it from the available data. Depending on the origin of the data, we propose two different interpolations one for data from measurement stations, and another for data from forecasting models such as CMAQ.

Algorithm 4.1 Wind field modelling

```
1: Node-by-node wind field interpolation
IF measurement stations data
  1.1: Horizontal interpolation at 10 m height
  1.2: Vertical extrapolation using log-linear vertical wind profile
ELSE IF CMAQ data
  1.1: Use algorithm 2.2
END IF
2: Approximation of the interpolated wind field with a mass-consistent model
  2.1: Impose a divergence null field
  2.2: Impose impermeability in the terrain
```

Algorithm 4.1 describes the steps that are necessary to compute the wind field. As you can see, when wind field data is from CMAQ we will use the algorithm described in Alg. 2.2, so in this Chapter we will describe the interpolation of data from measurement data. A description of the physical model and its discretization is described. The approach used in this work is based on the log-linear vertical wind profile model [Lalas and Ratto, 1996]. The algorithms described in this Chapter are implemented in the program Wind3D [Rodríguez et al., 2012], its source code is available at `http://www.dca.iusiani.ulpgc.es/Wind3D`.

4.1 Interpolation of the wind field from measurement stations data

In this Section, we describe how to construct the interpolated wind field ($\mathbf{u}_0$) from experimental data. To compute this wind field in the whole computational domain, we consider an horizontal interpolation and a vertical extrapolation of the available measurements. This method can also be used with data from numerical weather prediction models; In this case only the velocity at 10 m will be considered, and it will be treated as experimental measurements [Oliver et al., 2015b].

In the simulation of the wind field several parameters are uncertain, for this reason some calibration is required. Genetic algorithms are a good choice as the calibration method, and at the end of the chapter we will describe how to use them in our problem.

4.1.1 Horizontal interpolation

The most common technique of interpolation at a given point, placed at a height z_m above ground level, is formulated as a function of the inverse of the squared distance between that point and the measurement stations. On top of that, another strategy is to use the inverse of their height differences, as proposed by Palomino and Martín [1995] and Montero et al. [1998]. This approach takes into account the vertical variability which is important in very complex terrain, where phenomena such as drainage or diverted flows created by topographical obstacles can occur. In this work we propose a weighted arithmetic mean between both techniques

$$\mathbf{u}_0(z_m) = \xi \frac{\sum_{n=1}^{N} \frac{\mathbf{u}_n}{d_n^2}}{\sum_{n=1}^{N} \frac{1}{d_n^2}} + (1-\xi) \frac{\sum_{n=1}^{N} \frac{\mathbf{u}_n}{|\Delta h_n|}}{\sum_{n=1}^{N} \frac{1}{|\Delta h_n|}} \tag{4.1}$$

where the value of $\mathbf{u}_n$ is the velocity measured at station n, N is the number of stations considered in the interpolation, d_n is the horizontal distance from station n to the point of the domain where we are computing the wind velocity, $|\Delta h_n|$ is the height difference between station n and the studied point, and ξ is a weighting parameter ($0 \leq \xi \leq 1$), that allows to give more importance to one of these interpolation criteria, so when $\xi \to 1$ the inverse of the squared distance is the interpolation method, and when $\xi \to 1$ the inverse of the height differences is the interpolation method. Parameter ξ is one of the parameters that can be calibrated using genetic algorithms (Section 4.3)

4.1.2 Vertical extrapolation

To compute the wind field in the whole domain, a log-linear wind profile is considered [Lalas and Ratto, 1996]. This profile takes into account the horizontal interpolation and the effect of the terrain roughness. Some parameters of this vertical profile depend on the air stability (neutral, stable or unstable atmosphere) according to the Pasquill stability class. The profile differentiates three different regions of the atmosphere, the lower one defined between the roughness length and the surface layer, a middle region between the surface layer and the planetary boundary layer, and a upper region above the planetary boundary layer. Figure 4.1 shows schematically the proposed vertical profile.

The vertical profile depends on the stability of the atmosphere. This stability is related to the turbulence and the vertical temperature gradient. The atmospheric stability can be divided in unstable, stable and neutral. Since turbulence is difficult to measure, Pasquill [1961] laid out an stability classification scheme that depends on solar radiation and wind speed. This classification is shown in table 4.1.

To compute the vertical wind field, let's start with the lower region. The value of the wind field is defined by the following logarithmic equation

$$\mathbf{u}_0(z) = \frac{\mathbf{u}^*}{k} (\log \frac{z}{z_0} - \Phi_m) \qquad z_0 < z \leq z_{sl} \tag{4.2}$$

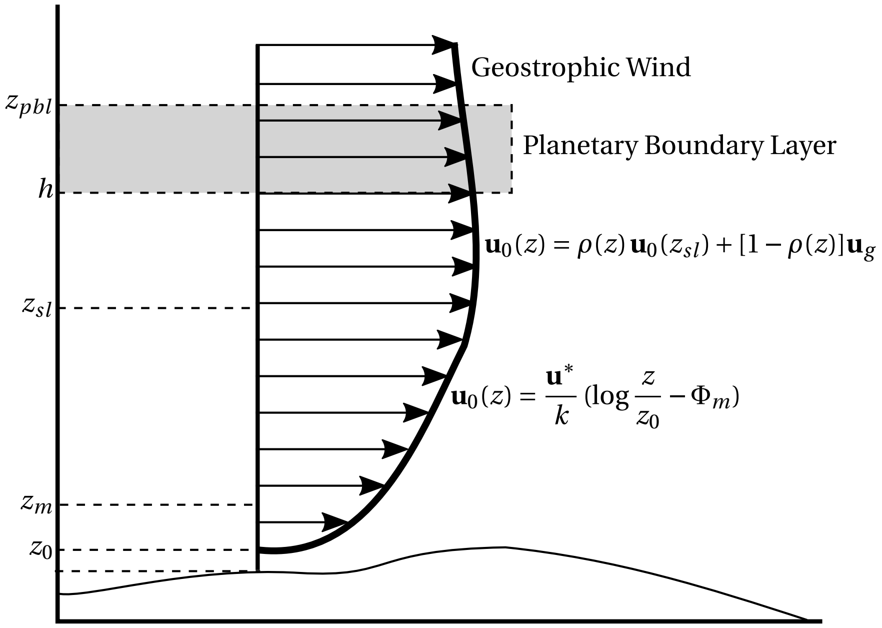

Geostrophic Wind
z_{pbl}
Planetary Boundary Layer
h
$\mathbf{u}_0(z) = \rho(z)\,\mathbf{u}_0(z_{sl}) + [1-\rho(z)]\mathbf{u}_g$
z_{sl}
$\mathbf{u}_0(z) = \frac{\mathbf{u}^*}{k}(\log\frac{z}{z_0} - \Phi_m)$
z_m
z_0

Figure 4.1: Vertical wind profile

	Daytime insolation			Night-time conditions	
Surface wind speed ($\mathrm{m\,s^{-1}}$)	Strong	Moderate	Slight	Thin overcast or > 4/8 low cloud	<= 4/8 cloudiness
< 2	A	A - B	B	E	F
2 - 3	A - B	B	C	E	F
3 - 5	B	B - C	C	D	E
5 - 6	C	C - D	D	D	D
> 6	C	D	D	D	D

Table 4.1: Meteorological conditions defining Pasquill stability classes

where $\mathbf{u}^*$ is the friction velocity, k is von Kármán's constant ($k = 0.41$), z_0 is the roughness length [McRae et al., 1982], and z_{sl} is the height of the surface layer. The values of Φ_m depend on the atmospheric stability. For neutral atmosphere its value is $\Phi_m = 0$, for stable atmosphere $\Phi_m = -5\frac{z}{L}$, and for unstable atmosphere

$$\Phi_m = \log\left[\left(\frac{\theta^2+1}{2}\right)\left(\frac{\theta+1}{2}\right)^2\right] - 2\arctan\theta + \frac{\pi}{2} \tag{4.3}$$

being $\theta = (1-16\frac{z}{L})^{1/4}$ and $\frac{1}{L} = az_0^b$, where a and b are coefficients that depend on the Pasquill stability class [Zannetti, 1990], see Table 4.2 for its values.

We must note that all the above parameters are known except the friction velocity, which is obtained from eq. (4.2). Using the velocity obtained from the horizontal interpolation $\mathbf{u}_0(z_m)$ at a given point (x, y), we can compute $\mathbf{u}^*$ as

$$\mathbf{u}^* = \frac{k\,\mathbf{u}_0(z_m)}{\log\dfrac{z_m}{z_0} - \Phi_m} \tag{4.4}$$

Pasquill stability class	a	b
A (Extremely unstable conditions)	-0.08750	-0.1029
B (Moderately unstable conditions)	-0.03849	-0.1714
C (Slightly unstable conditions)	-0.00807	-0.3049
D (Neutral conditions)	0.00000	0.0000
E (Slightly stable conditions)	0.00807	-0.3049
F (Moderately stable conditions)	0.03849	-0.1714

Table 4.2: Values of coefficients a and b according to the Pasquill stability class

The planetary boundary layer (PBL) is the lowest layer of the troposphere where the wind is influenced by friction. Above it, the wind field can be considered constant and equal to the geostrophic wind. The thickness of the PBL is not constant. At night and in the cool season the PBL tends to be lower in thickness while during the day and in the warm season it tends to have a higher thickness. As shown in Fig. 4.1, the thickness is defined between h and z_{pbl}. Since the PBL is influenced by the friction, its height (z_{pbl}) depends on the velocity friction.

$$z_{pbl} = \frac{\gamma\,|\mathbf{u}^*|}{f} \tag{4.5}$$

where $f = 2\Omega \sin\phi$ is the Coriolis parameter (Ω is the earth rotation and ϕ the latitude), and γ is a parameter that depends on the atmospheric stability which values varies between 0.15 and 0.3, due to this variation it is a parameter that can be calibrated.

The lower height of the PBL (h) is considered to be equal to z_{pbl} in neutral and unstable conditions. In stable conditions, it is approximated by

$$h = \gamma' \sqrt{\frac{|\mathbf{u}^*|\,L}{f}} \tag{4.6}$$

where $\gamma' = 0.4$. The height of surface layer is defined as one tenth of h

$$z_{sl} = \frac{h}{10} \tag{4.7}$$

Then, for the region between z_{sl} and z_{pbl}, a linear interpolation with geostrophic wind $\mathbf{u}_g$ is carried out,

$$\begin{aligned} \mathbf{u}_0(z) &= \rho(z)\,\mathbf{u}_0(z_{sl}) + [1-\rho(z)]\mathbf{u}_g \qquad z_{sl} < z \le z_{pbl} && (4.8)\\ \rho(z) &= 1 - \left(\frac{z - z_{sl}}{z_{pbl} - z_{sl}}\right)^2 \left(3 - 2\frac{z - z_{sl}}{z_{pbl} - z_{sl}}\right) && (4.9) \end{aligned}$$

Finally, this model assumes $\mathbf{u}_0(z) = \mathbf{u}_g$ if $z > z_{pbl}$ and $\mathbf{u}_0(z) = 0$ if $z \le z_0$.

4.2 Mass-consistent model

We consider a mass-consistent model [Montero et al., 1998, 2005, Ferragut et al., 2010] to compute a wind field $\mathbf{u}$ in the three-dimensional domain Ω, with a boundary $\Gamma = \Gamma_a \cup \Gamma_b$, that verifies the continuity equation and the impermeability condition on the terrain Γ_a,

$$\begin{aligned} \nabla\cdot\mathbf{u} &= 0 \quad \text{in}\,\Omega \\ \mathbf{n}\cdot\mathbf{u} &= 0 \quad \text{on}\,\Gamma_a \end{aligned} \tag{4.10}$$

where $\mathbf{n}$ is the outward-pointing normal unit vector.

The model formulates a Least-Squares problem in the domain Ω to find a wind field $\mathbf{u} = (u, v, w)$, such that it is adjusted as much as possible to the interpolated wind field $\mathbf{u}_0 = (u_0, v_0, w_0)$ computed in the previous step. The adjusting functional for a field $\mathbf{v} = (\tilde{u}, \tilde{v}, \tilde{w})$ is defined as

$$E(\mathbf{v}) = \frac{1}{2}\int_\Omega (\mathbf{v}-\mathbf{u}_0)^t\,\mathbf{P}\,(\mathbf{v}-\mathbf{u}_0)\,d\Omega \tag{4.11}$$

where $\mathbf{P}$ is a 3×3 diagonal matrix with $P_{1,1} = P_{2,2} = 2\alpha_1^2$ and $P_{3,3} = 2\alpha_2^2$. So, α_1 will control the adjustment of the horizontal components of the wind, and α_2 the vertical component of the wind. These parameters are not know and they will be calibrated.

The Lagrange multiplier technique is used to minimize the functional (4.11), with the restrictions (4.10). Considering the Lagrange multiplier λ, the Lagrangian is defined as

$$L(\mathbf{v},\lambda) = E(\mathbf{v}) + \int_\Omega \lambda\nabla\cdot\mathbf{v}\,d\Omega \tag{4.12}$$

and the solution $\mathbf{u}$ is obtained by finding the saddle point $(\mathbf{u},\phi)$ of the Lagrangian (4.12). This resulting wind field verifies the Euler-Lagrange equation,

$$\mathbf{u} = \mathbf{u_0} + \mathbf{P}^{-1}\nabla\phi \tag{4.13}$$

where ϕ is the Lagrange multiplier. Since α_1 and α_2 are constant in Ω, the variational approach results in an elliptic problem in ϕ, by substituting (4.13) in (4.10), that is solved by using the finite element method.

$$-\nabla\cdot\left(\mathbf{P}^{-1}\nabla\phi\right) = \nabla\cdot\mathbf{u}_0 \qquad \text{in}\,\Omega \tag{4.14}$$

$$-\mathbf{n}\cdot\mathbf{P}^{-1}\nabla\phi = \mathbf{n}\cdot\mathbf{u}_0 \qquad \text{on}\,\Gamma_a \tag{4.15}$$

$$\phi = 0 \qquad \text{on}\,\Gamma_b \tag{4.16}$$

4.3 Parameter calibration using Genetic algorithms

To compute the unknown parameters of the model, we will use genetic algorithms. Genetic algorithms are optimization tools based on the natural evolution mechanism. They produce successive trials that have an increasing probability to obtain a global optimum. The use of this technique has been proposed in the forecasting of wind field in the local scale using the numerical weather prediction model HARMONIE [Oliver et al., 2015b].

In this work we apply the parallel genetic algorithm developed by Levine [Levine, 1994]. The most important aspects of genetic algorithms are the construction of an initial population, the evaluation of each individual in the fitness function, the selection of the parents of the next generation, the crossover of those parents to create the children, and the mutation to increase diversity.

The initial population has been randomly generated and we use iteration limit exceeded as stopping criterion. The fitness function plays the role of the environment; it evaluates each string of a population to measure how closed it is to achieve its goal. This is a measure, relative to the rest of the population, of how well that string satisfies a problem-specific metric. The values are mapped to a non-negative and monotonically increasing fitness value. Once each string is evaluated, the whole population is sorted in order to perform the following steps.

Two population replacements are commonly used. The first, the generational replacement, that replaces the entire population each generation [Holland, 1992]. The second, used in this work, is known as steady-state and only replaces a few individuals each generation [Whitley, 1989]. It only replaces the worst individuals to improve the quality of the solution.

The stochastic universal selection [Baker, 1987] and the uniform crossover operators [Spears and Jong, 1995] have been selected. The selection phase chooses the strings that will remain in the new generation or that will be crossed in the next stage. This selection relies on the fitness values. Then, the crossover operator combines two strings of the selected population in order to generate a new one with characteristics of both strings. The idea is that since the parents have good fitness values, the new strings will be promising.

The mutation operator is better used after crossover [Davis, 1991]. It allows to create individuals in the search space forcing evaluations that, probably, wouldn't take over otherwise. The mutation operator used in this work replaces the gene value with a random one within the initialization range. For this reason is very important to select the initializations carefully.

Now that we know the basis of the genetic algorithms, we are going to define the parameters that are going to be calibrated, and the fitness function for this problem.

The parameters that must be calibrated are α_1, α_2, ξ, and γ.

To simplify the process, instead of using α_1 and α_2 we will define a new parameter α as

$$\alpha = \frac{\alpha_1}{\alpha_2} \tag{4.17}$$

This parameter, α, defines the predominant component of the flow adjustment, when $\alpha > 1$ the predominant component is the vertical, and when $\alpha < 1$ the predominant component is the horizontal.

The second parameter to be calibrated is the weighting coefficient ξ in the equation (4.1). Note that $\xi = 1$ implies the inverse of the squared distance interpolation, while $\xi = 0$ stands for a height difference interpolation.

The parameter γ is related to the height of the boundary layer z_{pbl}, and depends on the atmospheric stability. As stated in Section 4.1.2, its value varies in a range between 0.15 and 0.3.

Depending on the problem, Pasquill stability class can also be calibrated. This calibration can be problematic because for different classes several parameters and formulae of the model change dramatically. For example, when atmospheric conditions are stable, the value of $\gamma\prime$ defines the height of the mixed layer, in the rest of conditions $\gamma\prime$ is not required for the simulation. To address this drawback we can rely on the relations given by Pasquill (Table 4.1) and to restrict the choices to just two or three stabilities.

The fitting function to be minimized is the root mean square error (RMS) of the wind velocities given by the model with respect to the wind field measures.

$$E(\alpha,\xi,\gamma) = \sqrt{\frac{\sum_{n=1}^{N}\left[\left(u_n^\star - u_n\right)^2 + \left(v_n^\star - v_n\right)^2\right]}{2N}} \tag{4.18}$$

where $u_n^\star$ and $v_n^\star$ are the X and Y components of the wind field in the measurement station n, u_n and v_n are the X and Y components of the resulting wind field of the mass-consistent model (4.10), and N is the number of measurements stations.

5 Plume rise

The plume rise is an important phenomena to model the air quality model in the local scale. The emissions are injected into the atmosphere creating a perturbation of the wind field, that modifies the trajectory of the pollutants in their transport in the atmosphere.

The trajectory of the plume rise has been widely studied in the past [Briggs, 1969b,a, Moore, 1974]. These works differentiates two kind of cases: *predominant buoyancy rise* and *predominant momentum rise*. The characterization of these types essentially depends on the ratio between the intensities of the pollutant emission velocity and the wind velocity at the top of the stack.

The previous studies are based on an important assumption, they consider the ambient wind field constant with null vertical component. This assumption produces that the mean trajectory of the plume is placed on a vertical plane that contains the stack. Therefore, the resulting plume rise trajectory cannot be directly applied in the case of complex terrains. In this case, the plume rise trajectory follows a bent curve due to the variation of the ambient wind field.

Algorithm 5.1 shows the steps needed to introduce the effect of the plume rise in the wind field. Two different descriptions of the plume trajectory are described. The first one is a simplified trajectory [Oliver et al., 2013], and the second one is a more realistic one using Briggs' equations [Oliver et al., 2015a].

Algorithm 5.1 Plume rise construction

```
1: Determine the effective height and final distance of the plume
2: Compute the elevation of the plume (from the stack to the effective height)
3: Compute the trajectory of the plume combining the elevation computed in step 2 and ambient
   wind field computed in previous Chapter.
4: Refine the mesh along the plume rise to a size that can capture the effect of it.
5: Modify the vertical velocities of the nodes inside the plume to ensure the pollutant follows the
   imposed trajectory
```

The first step is to discover the characteristics of the plume, i.e. the effective height and the final distance. With this values, in step 2 we need to construct a reasonable plume rise mean vertical trajectory that takes into account the Briggs' equations and the ambient wind field obtained in the previous Chapter. Two different approaches have been proposed, one with a simplification of the trajectory using the final point computed using Briggs' equations, and another approach that uses the trajectory predicted by Briggs.

Then, step three modifies this vertical trajectory taking into account the ambient wind field. The result is that the plume bends following the atmospheric wind. In this case, two different strategies are considered.

Once the plume trajectory is defined, the effect of the pollutant injection will be introduced by modifying the ambient vertical wind velocity (w) along the region of the plume rise trajectory, modifying the vertical velocity of the plume (w_0). As stated by Briggs [1969a], w_0 is a convenient variable with which to identify the plume. To ensure that the vertical velocity is correctly modified, we first need to refine the mesh in the plume zone (step 4). Finally, in step 5 we will modify the value of w_0 at any point in the surroundings of the bent plume trajectory.

5.1 Simple plume rise model

To determine the effective height of the plume we need to know the trajectory of the plume. We can model this trajectory by using a simple thermodynamic strategy. Let's consider the plume rise as a pipe that is expanded and bent due to the ambient wind. We assume that the temperature is uniformly distributed in a section of the pipe with a radius r. Since the surrounding air has a different temperature (T), a gradient of the temperature ($\Delta T = T_c - T$) exists. So, if that the gradient of the moment is equal to the buoyancy force, the following equation arises

$$\frac{d}{dt}(\rho w_c \pi r^2) = w_c \frac{d}{dz}(\rho w_c \pi r^2) = \frac{\pi r^2 g \rho \Delta T}{T} \tag{5.1}$$

where ρ is the air density and g is the gravity force. Assuming that no heat is lost in the plume

$$\frac{d}{dz}(\pi r^2 \rho c_p \Delta T) = 0 \tag{5.2}$$

where c_p is the specific heat capacity. Considering that the plume radius grows linearly with height

$$r = az$$

Then equations (5.1) and (5.2) are satisfied if

$$w_c = bz^{-1/2} \qquad \Delta T = cz^{-2}$$

where

$$c = \frac{Q_H}{\pi u a^2} \qquad b = \left(\frac{2gQ_H}{3\pi a^2 uT}\right)^{1/2}$$

being Q_H the heat flow in the cap of the stack, i.e. $Q_H = \pi r^2 w_c \Delta T$, assuming that, except those near the stack, the sections of the plume will move at the same velocity as the atmospheric wind, then

$$\frac{dz}{dt} = w_c \propto \left(\frac{Q_H}{uz}\right)^{1/2}$$

and finally, being $x = ut$

$$z(x) \propto \frac{Q_H^{1/3} x^{2/3}}{u} \tag{5.3}$$

With this simple model we have an idea about the trajectory of the plume, but we don't know the distance, and therefore the height, that it will reach. Another drawback with this simple model is that it does not take into account the atmospheric stability. These limitations are tackled in the following Section, using Briggs equations.

5.2 Determination of the effective height of the plume

Gaussian plume models [Olcese and Toselli, 2005] approximate the effective height of a plume z_H and the horizontal distance d_f from the stack to the point where the plume height reaches z_H, depending on the emission characteristics, the ambient wind and the atmospheric stability. The gas elevation depends on two factors. The first one is the difference in temperature between the emitted gas and the ambient air that determines the plume density (buoyancy rise). The second one is the difference of the emission velocity and the ambient wind (momentum rise).

In order to compute the effective height of the plume z_H, we have to deal with stack dip downwash [Boubel et al., 1994]. Stack tip downwash can occur where the ratio of the stack exit velocity to wind speed is small. In this case, the small velocity can block the exit. This phenomena is taken into account by modifying the stack height (z'_c) defined as

$$z'_c = z_c \qquad \text{if } w_c \geq 1.5\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right| \tag{5.4}$$

$$z'_c = z_c + 2D_c\left[\left(w_c/\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|\right) - 1.5\right] \qquad \text{if } w_c < 1.5\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right| \tag{5.5}$$

where (x_c, y_c, z_c) are the coordinates of the centre of the top of the stack, D_c is the diameter of the emission surface and w_c is the emission velocity of gases. From now on this modified height (z'_c) will be used.

5.2.1 Predominant buoyancy rise

When the buoyancy rise is predominant, i.e., $w_c \leq 4\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|$, the usual trajectory of the plume rise is represented in Fig. 5.1. In this case, Briggs' equations consider the buoyancy flow parameter

$$F = g\,w_c D_c^2 \frac{T_c - T}{4T_c} \tag{5.6}$$

where g is the gravitational acceleration, T_c is the temperature of the emitted gases in Kelvin scale and T is the ambient temperature in Kelvin scale.

For unstable or neutral atmospheric conditions, z_H and d_f may be approximated in metres depending on the F value.

$$z_H = z'_c + 21.425\frac{F^{3/4}}{\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|} \qquad d_f = 49F^{5/8} \qquad \text{If } F < 55 \tag{5.7}$$

$$z_H = z'_c + 38.71\frac{F^{3/5}}{\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|} \qquad d_f = 119F^{2/5} \qquad \text{If } F \geq 55 \tag{5.8}$$

For stable conditions Briggs' equations define the stability parameter $s = \dfrac{g}{T}\dfrac{\Delta\theta}{\Delta z}$, where $\dfrac{\Delta\theta}{\Delta z}$ is the mean potential temperature gradient from the top of the stack to the top of the plume. If $\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right| \geq 0.2746F^{1/4}s^{1/8}$, then

$$\begin{aligned} z_H &= z'_c + 2.6\left(\frac{F}{s\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|}\right)^{1/3} \\ d_f &= 2.07\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right| s^{-1/2} \end{aligned} \tag{5.9}$$

on the contrary, for low velocities of wind, i.e., $\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right| < 0.2746F^{1/4}s^{1/8}$

$$\begin{aligned} z_H &= z'_c + 4F^{1/4}s^{-3/8} \\ d_f &= 0 \end{aligned} \tag{5.10}$$

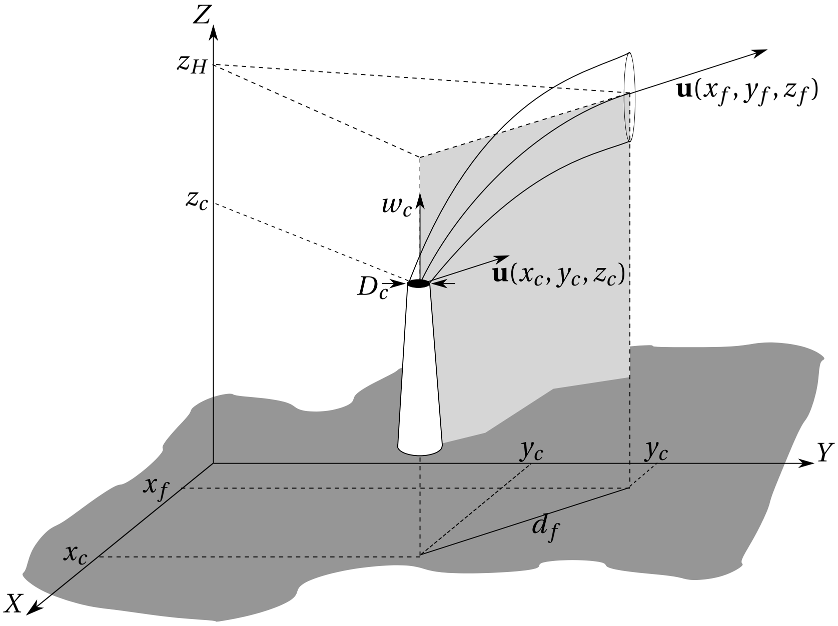


Figure 5.1: Usual trajectory of the plume rise with predominant buoyancy rise

5.2.2 Predominant momentum rise

If the momentum rise is predominant, i.e., $w_c > 4\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|$, the rise is completely vertical, $d_f = 0$. Figure 5.2 represents the trajectory of the plume in this case. The height of the rise z_H depends on atmospheric conditions.

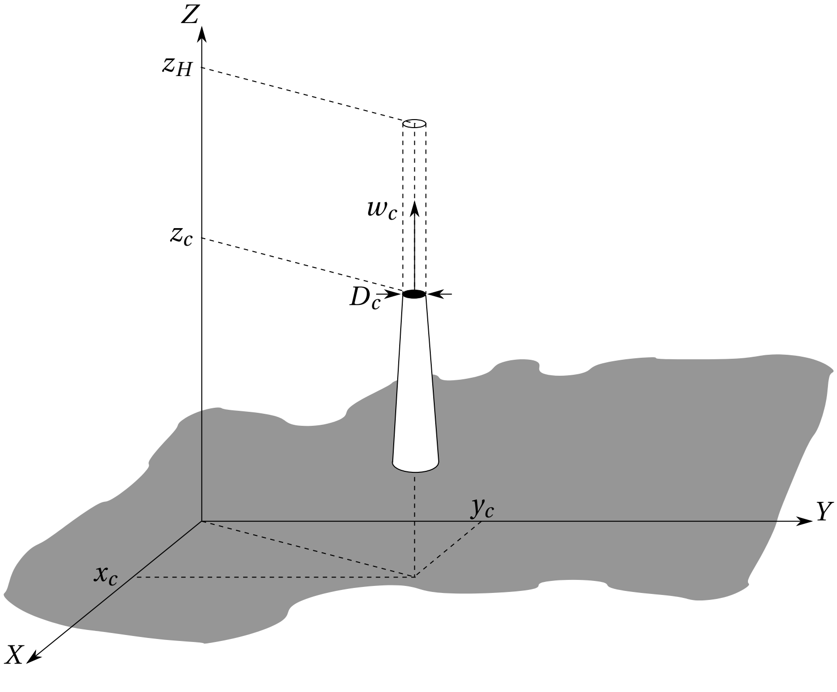


Figure 5.2: Trajectory of the plume rise with predominant momentum rise

For unstable or neutral conditions

$$z_H = z'_c + \frac{3D_c w_c}{\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|} \tag{5.11}$$

For stable conditions z_H is approximated by the lowest value of equations (5.11) and (5.12)

$$z_H = z'_c + 1.5\left[\frac{D_c^2 w_c^2 T}{4T_c\left|\mathbf{u}\left(x_c, y_c, z_c\right)\right|}\right]^{1/3} s^{-1/6} \tag{5.12}$$

5.3 Plume rise mean trajectory

Once we know the predominant force in the plume rise, and the values of z_H and d_f, a formulation to know the trajectory of the plume rise and the vertical component of the perturbed wind field along this trajectory has to be defined.

Firstly, we will consider separately the two different types of trajectories, bent-over or vertical, in the vertical plane defined by the top of the stack and the end of the plume.

Secondly, we will extend the previous results to three-dimensional trajectories in the case of a complex terrain.

5.3.1 Simplified plume trajectory and velocity

In all cases with d_f different from zero, that is equations (5.7), (5.8) and (5.9), the driving force is buoyancy, except for stable conditions and calm wind. To know the plume rise trajectory, we propose to combine an horizontal and a vertical motion, verifying certain known conditions.

The vertical motion along the mean trajectory of the plume is defined by an acceleration $a_0(t)$, a velocity $w_0(t)$ and $z(t)$, from the initial time $t=0$ to the final time $t=t_f$ when the plume reaches the effective height, verifying the following conditions

$$z(0) = z'_c \qquad z(t_f) = z_H \tag{5.13}$$

$$w_0(0) = w_c \qquad w_0(t_f) = 0 \tag{5.14}$$

Since there are four conditions on the vertical motion, we propose a cubic approximation of $z(t)$, and therefore a quadratic approximation of $w_0(t)$, and a linear approximation of $a_0(t)$ given by the following expressions

$$z(t) = z'_c + w_c t + \frac{-2w_c t_f + 3(z_H - z'_c)}{t_f^2} t^2 + \frac{w_c t_f - 2(z_H - z'_c)}{t_f^3} t^3 \tag{5.15}$$

$$w_0(t) = w_c + \frac{-4w_c t_f + 6(z_H - z'_c)}{t_f^2} t + \frac{3w_c t_f - 6(z_H - z'_c)}{t_f^3} t^2 \tag{5.16}$$

$$a_0(t) = \frac{-4w_c t_f + 6(z_H - z'_c)}{t_f^2} + \frac{6w_c t_f - 12(z_H - z'_c)}{t_f^3} t \tag{5.17}$$

To satisfy the conditions (5.14), and considering that $w_0(t)$ is a decreasing function in $[0, t_f]$, we impose $a_0(t) \le 0$ along the trajectory, obtaining the following condition for the unknown t_f

$$\frac{3}{2}\left(z_H - z'_c\right) \le w_c t_f \le 3\left(z_H - z'_c\right) \tag{5.18}$$

To determine the value of t_f, an additional assumption has to be done introducing the horizontal motion.

The horizontal motion is defined by a uniformly accelerated motion, with a constant positive acceleration vector $\mathbf{a_d} = (a_{dx}, a_{dy})$, a velocity $\mathbf{u_d}(t) = (u_d(t), v_d(t))$, and an horizontal relative position vector $\mathbf{d}(t) = (x(t) - x_c, y(t) - x_c)$ with respect to the centre of the stack, verifying the following conditions

$$|\mathbf{d}(t_f)| = d_f \tag{5.19}$$

$$\mathbf{u_d}(0) = \mathbf{u}(x_c, y_c, z_c) \tag{5.20}$$

Note that at the top of the stack the ambient wind $\mathbf{u}(x_c, y_c, z_c)$ has only horizontal components because of the imposition of the impermeability condition (4.10) at the top of the stack when solving the mass-consistent problem.

The horizontal trajectory can be expressed as

$$\begin{aligned} x(t) &= x_c + u(x_c, y_c, z_c)\,t + \frac{1}{2} a_{d_x} t^2 \\ y(t) &= y_c + v(x_c, y_c, z_c)\,t + \frac{1}{2} a_{d_y} t^2 \end{aligned} \tag{5.21}$$

From the previous conditions, we obtain an additional restriction for the unknown t_f

$$t_f = \frac{1}{a_d}\left(-\left|\mathbf{u}(x_c, y_c, z_c)\right| + \sqrt{\left|\mathbf{u}(x_c, y_c, z_c)\right|^2 + 2a_d d_f}\right) \tag{5.22}$$

where $a_d = |\mathbf{a_d}|$ is still an unknown. Imposing the condition (5.18) we obtain the following expression for a_d

$$a_d = (1+\delta)\frac{2w_c}{3(z_H - z_c')}\left[(1+\delta)\frac{w_c}{3(z_H - z_c')}d_f - \left|\mathbf{u}(x_c, y_c, z_c)\right|\right] \tag{5.23}$$

being $0 \le \delta \le 1$. For $\delta = 0$, the value of t_f is related to the upper bound in (5.18) and, for $\delta = 1$, to the lower bound. The case $\delta = 1/2$ corresponds to a value of t_f which produces a constant vertical acceleration a_0, a linear vertical component of velocity $w_0(t)$ and a quadratic vertical position $z(t)$.

Note that if the parameter δ is fixed, the horizontal and vertical motion is completely defined verifying strictly the conditions (5.19), (5.20), (5.13) and (5.14).

In all cases where d_f is equal to zero, that is when the driving force is momentum or when the driving force is buoyancy with calm wind (5.10), the horizontal motion of the plume until reaching the effective height can be considered negligible. Thus the trajectory of the gases is nearly vertical (see Fig. 5.2).

In this case, we propose a vertical motion along the trajectory of the plume with a constant negative acceleration a_0, a linear velocity $w_0(t)$ and a quadratic trajectory $z(t)$. Imposing the conditions (5.13) and (5.14) this vertical motion is completely defined.

$$t_f = \frac{2\left(z_H - z_c'\right)}{w_c} \tag{5.24}$$

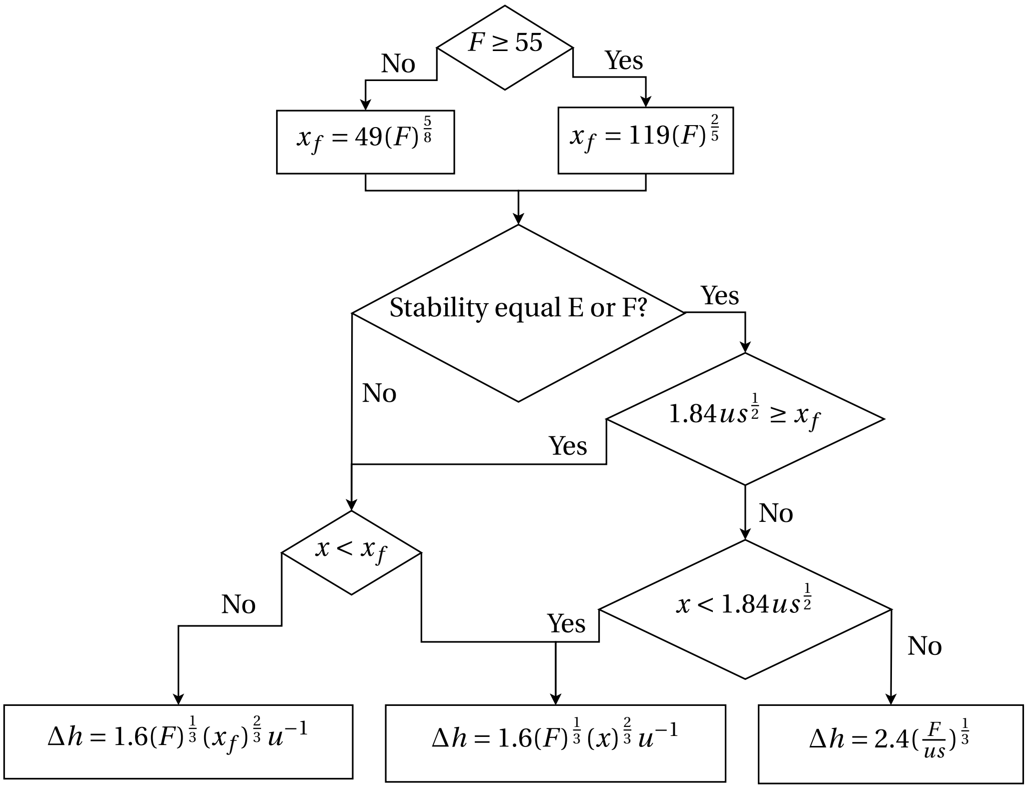
$$a_0 = \frac{-w_c}{t_f} \tag{5.25}$$

$$w_0(z) = w_c\sqrt{1 - \frac{2\left(z - z_c'\right)}{w_c t_f}} \tag{5.26}$$

We remark that using the value of t_f from (5.24) in the equation (5.15), the coefficient of its cubic term is null. As we have commented, this cancellation also occurs when $\delta = 1/2$ is chosen in (5.23). It is also important to note that using the value of t_f from (5.24) in the equation (5.17) results (5.25). Thus, fixing $\delta = 1/2$, the vertical motion in the bent-over plume case, given by equations (5.15), (5.16) and (5.17), and in the vertical plume case, given by equations (5.24), (5.25) and (5.26), are the same. This result makes compatible the vertical motion in both cases and justifies the election of $\delta = 1/2$.

5.3.2 Briggs plume trajectory and velocity

Briggs not only defined the effective height and final distance, but also determined the trajectory of the plume. Figure 5.3 shows the different trajectories a buoyant plume rise can take depending on the value of the buoyant flow parameter (F), the atmospheric stability, and the stability parameter (s). The, the trajectory used in this approach is the one determined by Briggs.



$F \geq 55$
No
Yes
$x_f = 49(F)^{\frac{5}{8}}$
$x_f = 119(F)^{\frac{2}{5}}$
Stability equal E or F?
Yes
No
$1.84us^{\frac{1}{2}} \geq x_f$
Yes
No
$x < x_f$
No
Yes
$x < 1.84us^{\frac{1}{2}}$
No
$\Delta h = 1.6(F)^{\frac{1}{3}}(x_f)^{\frac{2}{3}}u^{-1}$
$\Delta h = 1.6(F)^{\frac{1}{3}}(x)^{\frac{2}{3}}u^{-1}$
$\Delta h = 2.4(\frac{F}{us})^{\frac{1}{3}}$

Figure 5.3: Buoyant plume rise trajectory formulated by Briggs

In the case of the momentum plume the trajectory will be straight with the limit marked by the effective height computed in Section 5.2.

The vertical velocity in the plume has to ensure that the imposed trajectory. That is, if u_h is the ambient horizontal velocity, it has to satisfy

$$\frac{dz_t}{dt} = \frac{w_0}{u_h} \tag{5.27}$$

5.4 Extension of the plume mean trajectory to 3D over complex terrains

Several approximations can be considered to define the mean trajectory of bent curved plumes considering influence of complex terrains. Extension of vertical plumes is straightforward. In the case of buoyant plumes, to compute the trajectory we propose a new solution that takes into account the trajectories obtained in the vertical plane case, and the non uniform three-dimensional ambient wind obtained in Chapter 4.

Two different strategies are presented, the first one is specific for the simplified plume trajectory (Section 5.3.1), the second one is more generic and can be used for both the simplified trajectory or the Briggs trajectory (Section 5.3.2).

5.4.1 Simplified trajectory 3D extension

A three-dimensional bent trajectory is defined using an iterative process starting from the emission point (x_c, y_c, z_c). We uniformly subdivide the interval of time $[0, t_f]$ defining $n+1$ time instants $t_0, t_1, \ldots, t_n$, such that $t_i = i\Delta t$ for $i = 0, 1, \ldots, n$. For each time instant t_i a position of the mean trajectory is calculated.

Starting in time t_0 at position $\mathbf{x_0} = (x_c, y_c, z_c)$, and known the point $\mathbf{x_i} = (x_i, y_i, z_i)$, $\mathbf{x_{i+1}}$ can be computed from $\mathbf{x_i}$ with the following formulae

$$x_{i+1} = x_i + \frac{u(x_i, y_i, z_i)}{|(u(x_i, y_i, z_i), v(x_i, y_i, z_i))|}\left(u(x_c, y_c, z_c)\Delta t + \frac{1}{2} a_d \Delta t^2\right) \tag{5.28}$$

$$y_{i+1} = y_i + \frac{v(x_i, y_i, z_i)}{|(u(x_i, y_i, z_i), v(x_i, y_i, z_i))|}\left(v(x_c, y_c, z_c)\Delta t + \frac{1}{2} a_d \Delta t^2\right) \tag{5.29}$$

$$z_{i+1} = z(t_i) \tag{5.30}$$

where $z(t_i)$ is defined by (5.15), and the horizontal movement is deduced from (5.21).

The mean trajectory obtained by this method is a three-dimensional polygonal line, such that the longitude of its projection on the horizontal plane approximates the longitude d_f. In addition, the final height z_n coincides with the effective height z_H. Therefore, this method tries to verify the main values of the end of the plume considering Briggs' equations.

5.4.2 Briggs trajectory extension to 3D

This strategy is similar to the previous one, but two differences arises. The first one is that we are not relying in any particular definition of the distance, and we are using three-dimensional streamlines.

As in the previous strategy, we are defining a list of points, but in this case is just a list of distances that will be used in the computation of the streamline. Any list could work, but it is important to have more density near the stack that far from it; for example the following distribution will work.

$$d(i) = i^2 \tag{5.31}$$

where $d(i)$ is the i^{th} horizontal distance from the stack centre. Surely, we can also use the distance distribution that arises from the simplified trajectory strategy.

Once the distance distribution is known we have to compute the trajectory. This will be done in two steps. To compute a new point $\mathbf{x}_{i+1} = \{x_{i+1}, y_{i+1}, z_{i+1}\}$ from the previous one $\mathbf{x}_i = \{x_i, y_i, z_i\}$, we will compute the streamline that starts at point $\mathbf{x}_i$ with distance $d_{i+1} - d_i$, so the position at this step will be $\mathbf{x}_{i+1} = \{x^s_{i+1}, y^s_{i+1}, z^s_{i+1}\}$.

Then, we will compute the difference of height ($\Delta h(\mathbf{x}_{i+1})$) for that point using the equations from Section 5.3.2 (note that x in that formulas is the same as the distance d_{i+1}). Then, we will compute the difference with the previous difference height

$$\Delta h_{i+1} = \Delta h(\mathbf{x}_{i+1}) - \Delta h(\mathbf{x}_i) \tag{5.32}$$

This distance will be added to the z-coordinate of the previous point, so the final position will be $\mathbf{x}_{i+1} = \{x^s_{i+1}, y^s_{i+1}, z^s_{i+1} + \Delta h_{i+1}\}$

In the case of a momentum plume, the trajectory is the same as in the previous Section, so we will use those equations.

5.5 Local mesh refinement along a Gaussian plume

When we know the trajectory of the plume we need to modify the ambient vertical wind velocity (w) along the region of the plume rise. This perturbation is very local so we must have a sufficient mesh resolution in this area. For this reason, we propose to refine locally the mesh along the Gaussian plume [Green et al., 1980] until all the tetrahedra inside that region verify a size criteria.

The region can be defined in function of the diffusion coefficients of the Gaussian plume (σ_y, σ_z). To determine if a sample point $\mathbf{x}$ of a tetrahedron T is inside this region, the first step is to find its closest point $\mathbf{x_p} \in \mathbf{x_0}, \mathbf{x_1}, \ldots, \mathbf{x_n}$. The second step is to calculate the distance $d(t_p)$

$$d(t_p) = \sqrt{(x(t_p) - x_c)^2 + (y(t_p) - y_c)^2} \tag{5.33}$$

where the time $t_p = p\Delta t$, and $x(t_p)$, $y(t_p)$ are evaluated with equation (5.21). Then, we can compute the values $\sigma_z(d(t_p))$ and $\sigma_y(d(t_p))$ in function of the Pasquill stability class. These values define vertical ellipses (centred in $\mathbf{x_p}$) whose major and minor semi-axes are proportional to σ_y and σ_z respectively. Finally, fixing a value $K_1 > 1$, we consider that the sample point $\mathbf{x}$ of the tetrahedron T is inside the Gaussian plume region if $|\mathbf{x} - \mathbf{x_p}| < K_1\sigma_y$.

We mark for refinement the tetrahedron T if it simultaneously verifies the following two conditions: (i) T has at least one sample point inside the region and (ii) the length of the longest edge of T is lower than σ_z/K_2.

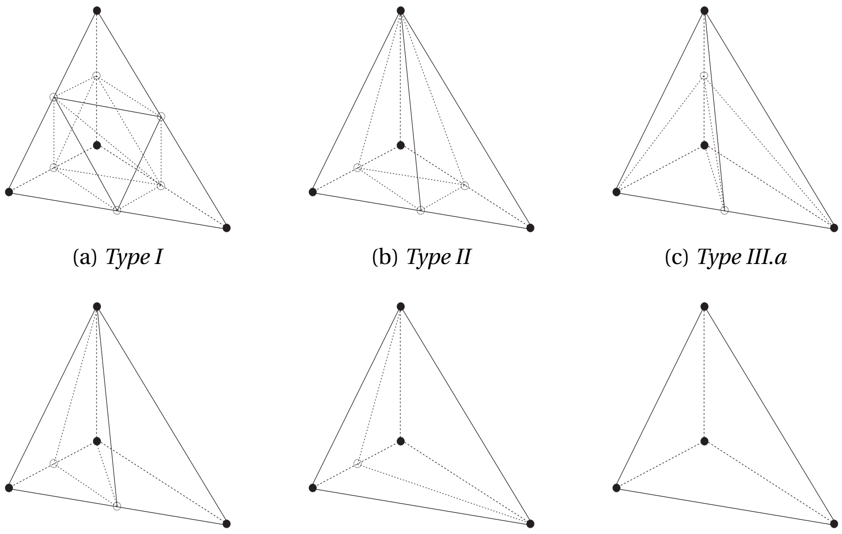
In the case of the vertical trajectory of the plume, we mark for refinement a tetrahedron T if it simultaneously verifies the following two conditions: (i) T has at least one sample point inside a vertical cylinder centred in (x_c, y_c, z_c) with the radius r_c of the top of the stack and a height $z_H - z_c$ and (ii) the length of the longest edge of T is lower than r_c/K_2.

Reasonable values of K_1 and K_2 are integer numbers between 1 and 3.

Once we know the elements to refine, depending on the mesh generator that we have used the refinement will be different.

In the case of the refinement-coarsening mesh generator (Section 3.1) to refine the marked tetrahedra, we use a local refinement algorithm [González-Yuste et al., 2004] based on the 8-subtetrahedron subdivision developed in [Löhner and Baum, 1992]. Figure 5.4 represents the different templates used in this process. An upper bound of the number of refinement steps can be fixed.

In the case of the curvature based mesh generator (Section 3.2) we will refine the mesh adding points into the mesh and using the Tetgen capability to refine an existing mesh adding these new points.



(a) *Type I* (b) *Type II* (c) *Type III.a*

(d) *Type III.b* (e) *Type IV* (f) *Type V*

Figure 5.4: Subdivision classification related to the new nodes (empty circles).

5.6 Wind field perturbation along the plume rise

Finally, a new ambient wind field **u** is obtained on the refined mesh with the mass-consistent model described in Chapter 4. The effect of the gas emission is introduced in this field by modifying its vertical component.

In the case of bent curve trajectory, the vertical component of the ambient wind velocity **u** is modified at any point **x** of the domain Ω located inside the region that was defined in the previous Section.

In the case of a bent trajectory, the velocity will be computed using the equations derived in Section 5.3. Depending on the trajectory used different velocities will arise.

In the case of vertical trajectory, the vertical component of the ambient wind velocity is modified inside a standard cylinder defined in the previous Section. That is, given a point $\mathbf{x} = (x, y, z)$ inside the cylinder, we impose the vertical velocity w of **u** by using equation (5.26) as $w_0(z)$.

6 Transport and reaction of pollutants in the atmosphere

The transport and reaction of pollutants in the atmosphere is the fundamental part of the air quality system. In this Chapter we will present the convection–diffusion–reaction equation that governs the problem. This equation will be solved with an Eulerian description using a finite element method stabilized with Least-Squares. To deal with the non-linear chemical reaction we propose to treat separately the transport of pollutants from the chemical reactions; to do that we will use a second order splitting operator. Finally to improve the solution of the problem an h-adaptivity scheme is applied. The h-adaptivity is based on an error indicator depending on the gradient of the logarithm of the concentration [Monforte and Pérez-Foguet, 2014a]. Adaptivity can lead to huge number of degrees of freedom when several species are considered. To handle this problem we propose to use a multimesh scheme where each pollutant specie is simulated in its own mesh [Monforte and Pérez-Foguet, 2014b]. Using this method, instead of having one mesh with a lot of degrees of freedom we have several meshes with a reasonable number of degrees of freedom. To use a mesh for each specie also allows to exploit parallelization. This multimesh approach has also been used in other problems [Li, 2005, Solin et al., 2010a,b].

6.1 Convection–diffusion–reaction equation

The air pollution simulation consists in solving the unsteady convection–diffusion–reaction formulation. The equation system governing the problem can be expressed with the following vectorial equation:

$$\frac{\partial \mathbf{c}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{c} = \nabla \cdot (\mathbf{K} \nabla \mathbf{c}) + \mathbf{e} + \mathbf{s}(\mathbf{c}) \tag{6.1}$$

for the spatial coordinates $\mathbf{x}$ and time t, $(\mathbf{x}, t) \in \Omega \times (0, t^{end}]$, with initial condition $\mathbf{c}(\mathbf{x}, 0) = \mathbf{c}^{ini}(\mathbf{x})$ on $\mathbf{x} \in \Omega$, and the following boundary conditions:

$$\begin{cases} \mathbf{c}(\mathbf{x}, t) = \mathbf{c}^{emi}(\mathbf{x}) & \text{in } \Gamma_S\text{: Top of the stack} \\ \mathbf{n} \cdot \mathbf{K} \nabla \mathbf{c} = -\mathbf{V}^d \mathbf{c} & \text{in } \Gamma_R\text{: Terrain} \\ \mathbf{n} \cdot \nabla \mathbf{c} = 0 & \text{in } \Gamma_{W_{out}}\text{: Outlet wind boundaries} \\ \mathbf{c}(\mathbf{x}, t) = \mathbf{c}^{out}(\mathbf{x}) & \text{in } \Gamma_{W_{in}}\text{: Inlet wind boundaries} \end{cases} \tag{6.2}$$

where ∇ is the gradient with respect $\mathbf{x}$, and $\mathbf{c}$, $\mathbf{u}$, $\mathbf{e}$ and $\mathbf{s}(\mathbf{c})$ are respectively the concentration, the perturbed wind velocity, the emission and the chemical vectors with a dimension n_c (the number of pollutant species), $\mathbf{K}$ is the diffusion matrix of dimension $3 \times n_c$, $\mathbf{V}^d$ is the deposition diagonal matrix with dimension n_c, and $\mathbf{n}$ is the outward-pointing normal unit vector, $\mathbf{c}^{emi}$ is the concentration of the emission in the top of the stack, and $\mathbf{c}^{out}$ the outside concentration at the inlet wind boundaries. Scalar product "$\cdot$" is applied n_c times: the first argument is multiplied by each one of the n_c components of the second argument.

The convection–diffusion–reaction equation depends basically on three factors; the wind field, the eddy diffusion, and the reaction of the pollutants.

The wind field has been computed in Chapter 4 and perturbed with the plume rise effect in Chapter 5.

The eddy diffusion term comprises all the processes that cause the mixing of pollutants in the atmosphere. The value of the eddy diffusion is not easy to know. The most simple model of the diffusion is to suppose a constant value for the whole domain. A more advanced technique is to use the K-theory to compute the vertical diffusion [Businger and Arya, 1974, Shir, 1973, Lamb and Durran, 1978]. Finally we can use the same algorithms as CMAQ where the vertical diffusion is computed using the Asymmetric Convective Model (ACM) [Pleim and Chang, 1992].

Regarding the reaction of the pollutants we want to note that the complete description of photochemical reaction of atmospheric species is highly complex [Finlayson-Pitts and Pitts, 1997, Kley, 1997, Andreae and Crutzen, 1997, Ravishankara, 1997]. For instance, detailed Volatile Organic Components decomposition involves hundreds of thousand reactions [Atkinson and Arey, 2003, Szopa et al., 2005]; that needs special methodologies to reduce the number of the modelled reactions and species. Reference models for gaseous phase reactions involve some tens of compounds [Jimenez et al., 2003, Kirchner, 2005]. The most simplified models just involve about ten reactive species [Zlatev, 1995]. On the other side, depending on the application, it can be necessary to take into account aqueous phase reactions, that involve several other reactions and species. The RIVAD model is one of the most simplified models that permit to simulate both processes, aqueous and gaseous, involving transport and reaction of four species [Scire et al., 2000]. The chemical mechanism used in CMAQ is the Carbon Bond CB05 [Yarwood et al., 2005].

6.2 Least-Squares finite element method discretization

To solve the convection–diffusion–reaction equation (eq. 6.1) we use a Galerkin finite element method stabilized with Least-Squares. To ease the development we will consider a linear chemical term, that is, $\mathbf{s}(\mathbf{c}) = \mathbf{A}\mathbf{c}$ where $\mathbf{A}$ is constant matrix.

The temporal discretization will solved with a Crank-Nicolson time integration scheme. So, concentrations $\mathbf{c}^n$ and $\mathbf{c}^{n+1}$ at times t^n and $t^{n+1} = t^n + \Delta t$ are related using a Crank-Nicolson scheme as

$$\mathbf{c}^{n+1} = \mathbf{c}^n + \frac{\Delta t}{2}\left[\frac{\partial \mathbf{c}^{n+1}}{\partial t} + \frac{\partial \mathbf{c}^n}{\partial t}\right] \tag{6.3}$$

To make notation compact, we define the differential operator $\mathscr{L}$ as

$$\mathscr{L} = \mathbf{u}\cdot\nabla - \nabla\cdot(\mathbf{K}\nabla) - \mathbf{A} \tag{6.4}$$

and a function $\mathscr{F}$ in Ω from the known $\mathbf{c}^n$

$$\mathscr{F} = \mathbf{c}^n - \frac{\Delta t}{2}\mathscr{L}\mathbf{c}^n + \frac{\Delta t}{2}\left[\mathbf{e}^{n+1} + \mathbf{e}^n\right] \tag{6.5}$$

Applying the Crank-Nicolson scheme, we can rewrite equation (6.1) as

$$\left[\mathscr{I} + \frac{\Delta t}{2}\mathscr{L}\right]\mathbf{c}^{n+1} = \mathscr{F} \tag{6.6}$$

where $\mathscr{I}$ is the identity operator.

Using Least-Squares we obtain a symmetric problem. The weak form of the equation (6.6) is

$$\left(\left[\mathscr{I}+\frac{\Delta t}{2}\mathscr{L}\right]v,\left[\mathscr{I}+\frac{\Delta t}{2}\mathscr{L}\right]\mathbf{c}^{n+1}\right)=\left(\left[\mathscr{I}+\frac{\Delta t}{2}\mathscr{L}\right]v,\mathscr{F}\right) \tag{6.7}$$

where v is the test function, and $(\mu, v) = \int_\Omega \mu v \, d\Omega$ is the inner product

Defining the next bilinear forms

$$a(v,\mathbf{c}) = (v,\mathbf{u}\cdot\nabla\mathbf{c}-\mathbf{A}\mathbf{c}) + (\nabla v,\mathbf{K}\nabla\mathbf{c}) + \int_{\Gamma_R}\mathbf{V}^d\mathbf{c}\,v\,d\Gamma_R \tag{6.8}$$

$$b(v,\mathbf{c}) = \sum_e(\mathbf{u}\cdot\nabla v-\mathbf{A}v,\mathbf{c})_e \tag{6.9}$$

$$g(v,\mathbf{c}) = \sum_e(\mathbf{u}\cdot\nabla v-\mathbf{A}v,\mathbf{u}\cdot\nabla\mathbf{c}-\mathbf{A}\mathbf{c})_e \tag{6.10}$$

where $(\mu, v)_e = \int_{\Omega_e} \mu v \, d\Omega$, and $\sum_e$ represents the sum over all the mesh elements, and applying the operators $\mathscr{L}$ and $\mathscr{F}$ in the weak form (6.7), we obtain

$$\begin{aligned}(v,\mathbf{c}^{n+1}) + \frac{\Delta t}{2}\left[a(v,\mathbf{c}^{n+1}) + b(v,\mathbf{c}^{n+1}) + \frac{\Delta t}{2}g(v,\mathbf{c}^{n+1})\right] = \\ (v,\mathbf{c}^{n}) - \frac{\Delta t}{2}\left[a(v,\mathbf{c}^{n}) - b(v,\mathbf{c}^{n}) + \frac{\Delta t}{2}g(v,\mathbf{c}^{n})\right] + \Delta t\left[(v,\mathbf{e}^{n+\frac{1}{2}}) + \frac{\Delta t}{2}b(v,\mathbf{e}^{n+\frac{1}{2}})\right]\end{aligned} \tag{6.11}$$

where $\mathbf{e}^{n+\frac{1}{2}} = \frac{\mathbf{e}^{n+1}+\mathbf{e}^{n}}{2}$.

Equation (6.11) can be written as a equation system

$$\mathbf{B}\boldsymbol{c}^{n+1} = \mathbf{f} \tag{6.12}$$

where $\boldsymbol{c}^{n+1}$ is the concentration vector approximation at t^{n+1} in the degrees of freedom of the finite element discretization, $\mathbf{f} = [2\mathbf{F}-\mathbf{B}]\boldsymbol{c}^n + \frac{\Delta t}{2}\mathbf{F}[\boldsymbol{e}^{n+1}+\boldsymbol{e}^n]$ and $\mathbf{B}$ and $\mathbf{F}$ are square matrices with dimension $(n_c \times n_{dof})$, being n_{dof} the number of degrees of freedom.

In order to solve this linear system it is necessary to find an efficient solver, using an sparse matrix storage. Since $\mathbf{B}$ is a symmetric positive definite matrix, we have considered a solver based on a conjugate gradient method preconditioned with an incomplete Cholesky factorization density type [Lin and Moré, 1999]. The left–preconditioning is used to improve the convergence of the conjugate gradient method. The original linear system is transformed into $\mathbf{T}^{-1}\mathbf{B}\boldsymbol{c}^{n+1} = \mathbf{T}^{-1}\mathbf{f}$, where $\mathbf{T}$ is the symmetric positive definite preconditioner obtained with the incomplete Cholesky factorization. The large fill-in of the complete (i.e. standard) Cholesky factorization is completely or partially avoided by discarding coefficients along the factorization process. We have considered an incomplete Cholesky factorization with no fill-in, such that the incomplete factor $\mathbf{L}$ has the same sparsity pattern as the lower triangle of matrix $\mathbf{B}$. The main advantage of the Cholesky method is that the incomplete factorization of matrix $\mathbf{B}$ can be amortized over many time-steps. More details about the implementation of this system equation solver can be found in Donea and Huerta [2003] and Rodríguez-Ferran and Sandoval [2007].

6.3 Non-linear chemical treatment

To deal with the non-linearity of the reactive term in the convection–diffusion–reaction equation (6.1), we have considered an splitting method that separates this equation into a convection–diffusion equation and a reaction equation. We will make use of the second order splitting operator (*Strang splitting*) proposed by Ropp et al. [2004]:

$$\frac{d\mathbf{c}^\star}{dt} = \mathbf{s}(\mathbf{c}^\star) \qquad \text{for } t \in \left[0, \frac{\Delta t}{2}\right] \text{ and } \mathbf{c}^\star(\mathbf{x},0) = \mathbf{c}^n(\mathbf{x}) \tag{6.13}$$

$$\frac{\partial \mathbf{c}^{\star\star}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{c}^{\star\star} = \nabla\cdot(\mathbf{K}\nabla\mathbf{c}^{\star\star}) + \mathbf{e} \qquad \text{for } t \in [0,\Delta t] \text{ and } \mathbf{c}^{\star\star}(\mathbf{x},0) = \mathbf{c}^\star(\mathbf{x},\frac{\Delta t}{2}) \tag{6.14}$$

$$\frac{d\mathbf{c}^{\star\star\star}}{dt} = \mathbf{s}(\mathbf{c}^{\star\star\star}) \qquad \text{for } t \in \left[\frac{\Delta t}{2}, \Delta t\right] \text{ and } \mathbf{c}^{\star\star\star}(\mathbf{x},\frac{\Delta t}{2}) = \mathbf{c}^{\star\star}(\mathbf{x},\Delta t) \tag{6.15}$$

Once we have split the equation (6.1), we solve three equations in different time steps; the reaction equation (6.13), the convection–diffusion equation (6.14), and the reaction equation (6.15), being finally $\mathbf{c}^{n+1}(\mathbf{x}) = \mathbf{c}^{\star\star\star}(\mathbf{x},\Delta t)$.

The transport (convection–diffusion) equation (6.14) is solved using the same method proposed in the previous Section, being $\mathbf{A} = 0$. The non-linear chemical equations (6.13) and (6.15) depends on the chemical mechanism chosen. In this thesis we will use the RIVAD model or the CB05 mechanism used in CMAQ.

6.3.1 RIVAD model

This model is a pseudo first-order chemical scheme for acid rain simulation, specially calibrated for being used on non-urban areas. The concentration **c** involves four species, $c_1 = [SO_2]$, $c_2 = [SO_4]$, $c_3 = [NO_2]$ and $c_4 = [NO_3]$, and the components of the reaction vector $\mathbf{s}(\mathbf{c})$ are:

$$s_1(\mathbf{c}) = -s_2(\mathbf{c}) = -\alpha_1(\mathbf{c})c_1 \tag{6.16}$$

$$s_3(\mathbf{c}) = -s_4(\mathbf{c}) = -\alpha_3(\mathbf{c})c_3 \tag{6.17}$$

where $\alpha_1(\mathbf{c}) = \gamma_1/(c_1+\delta_1 c_3)$ and $\alpha_3(\mathbf{c}) = \gamma_3/(c_3+\delta_3 c_1)$. Note that for values close to zero of the concentration of the primary species c_1 and c_3, both $\alpha_1(\mathbf{c})$ and $\alpha_3(\mathbf{c})$ requires a proper numerical treatment in order to avoid excessively high reaction rates.

To solve this term we use a second order Rosenbrock method (ROS2) [Verwer et al., 1999]. Rosenbrock solvers are a generalization of Runge–Kutta methods designed to deal with stiff ODEs. It uses the Jacobian square matrix of $\mathbf{s}(\mathbf{c})$ of dimension n_c to solve the problem.

$$\frac{\partial \mathbf{s}_1(\mathbf{c})}{\partial \mathbf{c}_1} = \frac{-\gamma_1\delta_1\mathbf{c}_3}{(\mathbf{c}_1+\delta_1\mathbf{c}_3)^2} = -\frac{\partial \mathbf{s}_2(\mathbf{c})}{\partial \mathbf{c}_1} \tag{6.18}$$

$$\frac{\partial \mathbf{s}_1(\mathbf{c})}{\partial \mathbf{c}_3} = \frac{\gamma_1\delta_1\mathbf{c}_1}{(\mathbf{c}_1+\delta_1\mathbf{c}_3)^2} = -\frac{\partial \mathbf{s}_2(\mathbf{c})}{\partial \mathbf{c}_3} \tag{6.19}$$

$$\frac{\partial \mathbf{s}_3(\mathbf{c})}{\partial \mathbf{c}_1} = \frac{\gamma_3\delta_3\mathbf{c}_3}{(\mathbf{c}_3+\delta_3\mathbf{c}_1)^2} = -\frac{\partial \mathbf{s}_4(\mathbf{c})}{\partial \mathbf{c}_1} \tag{6.20}$$

$$\frac{\partial \mathbf{s}_3(\mathbf{c})}{\partial \mathbf{c}_3} = \frac{-\gamma_3\delta_3\mathbf{c}_1}{(\mathbf{c}_3+\delta_3\mathbf{c}_1)^2} = -\frac{\partial \mathbf{s}_4(\mathbf{c})}{\partial \mathbf{c}_3} \tag{6.21}$$

6.3.2 CB05

The Carbon Bond chemical mechanism CB05 is a condensed mechanism of atmospheric oxidant chemistry that provides a basis for computer modeling studies of ozone, part iculate matter (PM), visibility, acid deposition and air toxics issues [Yarwood et al., 2005, Sarwar et al., 2007]. It is the chemical mechanism used in CMAQ, updated with the last development (CB05-TU) [Whitten et al., 2010]. The core CB05 mechanism has 51 species and 156 reactions, on top of that, CB05-TU adds 11 more species and 26 reactions. These reactions are modelled as ordinary differential equations (ODE), so a ODE system has to be solved.

To solve this system, three different models are available in CMAQ: Rosenbrock (ROS3) [Sandu et al., 1997a], SMVGEAR [Jacobson and Turco, 1994], and the Euler Backward Iterative (EBI) method [Hertel et al., 1993].

ROS3 is a third order solver of the Rosenbrock family, so it has the same properties and resolution method as the ROS2 method used to solve RIVAD model.

SMVGEAR is an improvement over GEAR-type code [Gear, 1971] to solve ODE in large grid domains. The improvements consists in using an sparse matrix storage, vectorize all of the loops. To solve the ODE, this method solves iteratively the system

$$\mathbf{P}\hat{\mathbf{e}} = \mathbf{B} \tag{6.22}$$

where $\hat{\mathbf{e}}$ is a vector correct the concentration values $\mathbf{c}$, $\mathbf{B}$ is the right hand side vector of the ODE with the current values of concentrations, and $\mathbf{P}$ is a matrix defined as

$$\mathbf{P} = \mathbf{I} + \Delta t \beta_0 \mathbf{J} \tag{6.23}$$

where $\mathbf{I}$ is the identity matrix, β_0 is an scalar that depends on the order, and $\mathbf{J}$ is the Jacobian matrix.

Equation 6.22 is solved, updating $\mathbf{B}$ with the new values of the concentration until the corrected vector $\hat{\mathbf{e}}$ becomes less than a predefined error.

The EBI method is based on the backward Euler approximation. Using EBI the approximation of the i^{th} ODE can be written as

$$c_i^{n+1} = c_i^n + P_i^{n+1}\Delta t - L_i^{n+1}\Delta t c_i^{n+1} \tag{6.24}$$

where c_i^{n+1} is the concentration of the i^{th} pollutant at t^{n+1}, c_i^{n+1} is the concentration of the i^{th} pollutant at t^n, P_i^{n+1} is the chemical production of the i^{th} pollutant at t^{n+1}, L_i^{n+1} is the chemical loss of the i^{th} pollutant at t^{n+1}. This equation can be rewritten as

$$c_i^{n+1} = \frac{c_i^n + P_i^{n+1}\Delta t}{1 + L_i^{n+1}\Delta t} \tag{6.25}$$

To compute c_i^{n+1}, the right hand side is calculated by using some explicit formulas and value of concentrations in the previous iterative step. This is done iteratively until the difference of concentration values in two consecutive steps is smaller than a defined error.

6.4 Adaptation and multimesh scheme

To improve the solution of the simulation, an h-adaptivity scheme is used. Error is approximated by using an error indicator. Several indicators are proposed in the literature [John, 2000]; we will use the error indicator proposed by Monforte and Pérez-Foguet [2014a].

$$\begin{cases} \eta_e = \|\nabla \log(c)\| & \text{if } c > \hat{c} \\ 0 & \text{Otherwise} \end{cases} \tag{6.26}$$

where $\hat{c}$ is a lower limit of the solution. Beyond this limit the mesh will no longer be refined.

We set a threshold of the error indicator to determine if the mesh has to be refined or coarsened. For η_e greater than this threshold the mesh will be refined, and for values smaller the mesh will be coarsened.

To adapt the mesh to the error indicator we are taking advantage of the capability of Tetgen [Si, 2015] to use a volume background mesh. So, we need to convert the error indicator to define a new volume.

In the case of refining the mesh, for a given element with a volume V_e, we will define the new volume V_e' as

$$V_e' = \frac{V_e}{1+\alpha\eta_e} \tag{6.27}$$

where α is a parameter that controls the refinement we want to obtain.

When we need to coarsen the mesh the following volume will be defined

$$V_e' = \min\left(\frac{V_e}{1+\alpha\eta_e}, \beta V_e^0\right) \tag{6.28}$$

where V_e^0 is the original volume of the element, and β is a parameter that controls the maximum volume that we want to allow with respect to the original mesh. This restriction ensures that we do not lose characteristics of the geometry or the wind field that can be important for the problem.

When the mesh is adapted in some cases it is important to recalculate the solution with the new mesh. In the problem at hand, we will require to recalculate at the initial step and every time some field changes, e.g. wind field, boundary conditions, etc.

When we use this adaptation scheme in a problem with multiple species, for a given element each specie will ask for a new volume. The most restrictive volume is the one used in the refinement. To chose the most demanding volume can lead to a very large mesh; to avoid this problem we will use a multimesh approach [Monforte and Pérez-Foguet, 2014b]. This scheme using a multimesh adaptation is described in Algorithm 6.1.

Algorithm 6.1 Multimesh adaptation scheme

```
FOR EACH timestep
    1: Solve the transport each specie in its mesh
    2: Solve the reaction in a common mesh
    IF adapt
        3.1: Compute the error indicator for each mesh
        3.2: Adapt the meshes where the indicator ask for it
        IF recalculate
            4.1: Project the recalculation initial concentration
            4.2: Reset the timestep
        ELSE
            4.1: Project the current concentration between meshes
        END IF
    END IF
END
```

In the multimesh approach, each specie will be computed in its own mesh. As stated before, the problem is composed by the transport and reaction steps. The transport step can be solved independently for each specie, but the reactive step gathers all the species, so we need to project all the species values to a common large mesh, where the reaction will be solved.

7 Diagnostic application: La Palma island (Canary islands, Spain)

In this Chapter a realistic example is presented. We propose to study a region in La Palma island. The topography of the island is real, we are using a digital elevation model, but the wind field data and the location of the stack and its emissions are simulated. This application can be seen in Oliver et al. [2013]

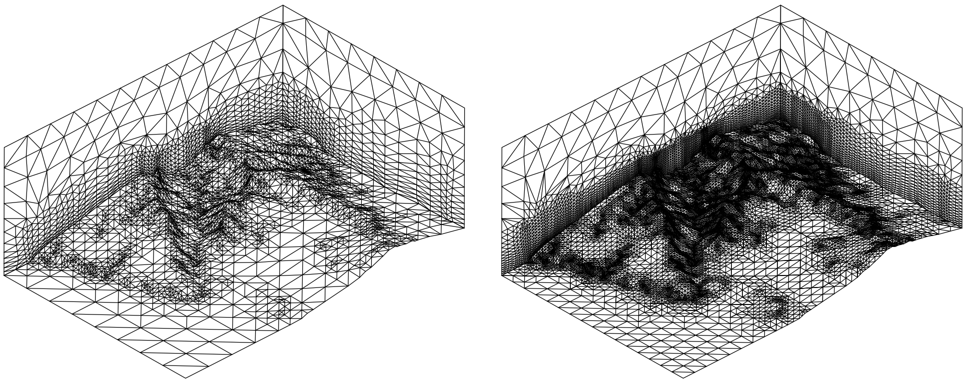
7.1 Adaptive tetrahedral mesh

The domain of interest is a rectangular area with dimensions 15 600 m × 22 803 m. The topography of the terrain is highly complex ranging from the sea level up to a maximum height of 2279 m with several deep valleys. The upper boundary of the domain has been placed at $h = 9000$ m. The digital terrain model of the area is defined over a uniform grid with a spacing step of 200 m in both directions x and y. The stack geometry is inserted into the topographical data. Let us consider a stack with a height of 150 m above ground level and the diameter at its top of 15 m. The location of the stack is indicated in Fig. 7.4.

For this application the mesh has been built using the refinement-coarsening method explained at Section 3.1. The mesh must be able to detect the details of the stack; therefore, we impose an element size of about 2 m × 2 m in the stack. If we start from a uniform mesh τ_1 of the rectangular area with an element size of about 2000 m × 2000 m, we must make ten refinement steps close to the stack. However, we make only five global refinement steps over the whole domain and five additional local refinements in the surroundings of the stack. The node distribution of τ_1 is the same as the mesh in the upper boundary of the domain. The tolerance of the adaptation process is $\varepsilon = 40$ m, i.e. the distance between the adapted mesh and the terrain surface is less than that value.

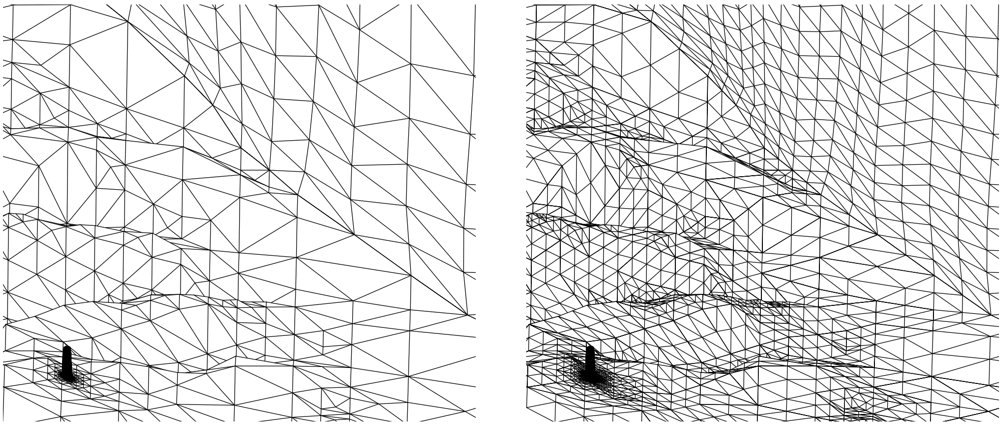
Figure 7.1 (a) shows the resulting tetrahedral mesh, with 198 146 nodes and 1 123 012 tetrahedra. Figure 7.2 (a) shows a detail of the surroundings of the stack. Figure 7.3 (a) shows the distribution of the mesh element quality using the mean ratio algebraic shape quality metric.

To have a more suitable mesh for the air quality problem, all the elements located below the height 3000 m have been refined. Figure 7.1 (b) shows the refined mesh with 455 953 nodes and 2 622 454 tetrahedra. Figure 7.2 (b) shows a detail of the surroundings of the stack and Fig. 7.3 (b) shows the distribution of the mesh element quality.



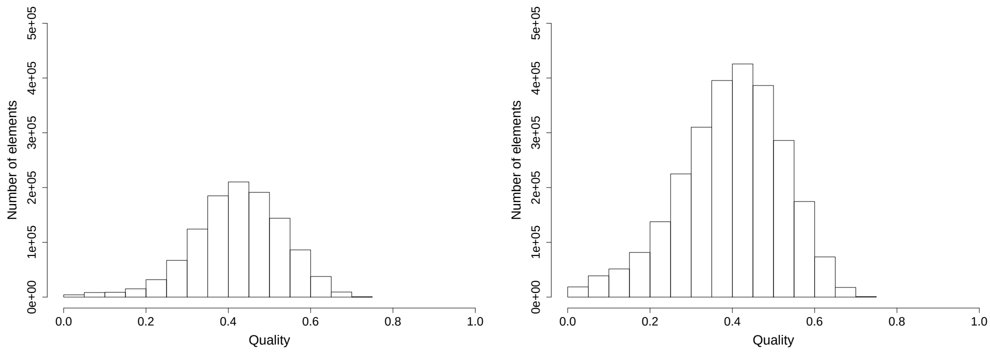
(a) (b)

Figure 7.1: La Palma tetrahedral mesh before (a) and after (b) the refinement under 3000 m



(a) (b)

Figure 7.2: Mesh detail around the stack before (a) and after (b) the refinement



(a) (b)

Figure 7.3: Mesh quality before (a) and after (b) the refinement

7.2 Ambient wind field

The wind field is constructed from the measurements of 4 stations. The locations of these stations are shown in Fig. 7.4, and the values of the corresponding wind velocities are listed in Table 7.1. The wind field has been computed by following the two steps described in Chapter 4. For the wind interpolation, we have considered a neutral Pasquill stability class, and the following parameter values: $\mathbf{u}_g = (-10, -1, 0)$, $\xi = 0.5$, and $\gamma = 0.15$. For the mass-consistent model an $\alpha = 1.0$ has been chosen.

Figures 7.5, 7.6 and 7.7 show the interpolated ($\mathbf{u}_0$) and the resulting ($\mathbf{u}$) ambient wind field at heights 20 m, 100 m and 500 m above ground level. The interpolated wind field preserves the measurements of the stations, but the terrain has no impact over it. Therefore, it can be seen in the figures how the wind field $\mathbf{u}_0$ crosses the terrain. However, the terrain has an strongly effect on the resulting wind field, verifying the conditions of the mass-consistent model. It can be noted that the wind field velocity is higher in the ridges of the mountains. In addition, the wind field is channelled in the ravines. It also can be noted, that the effect of the terrain decreases as the elevation increases, and the value of the wind velocity increases and tends to the uniform geostrophic wind field.

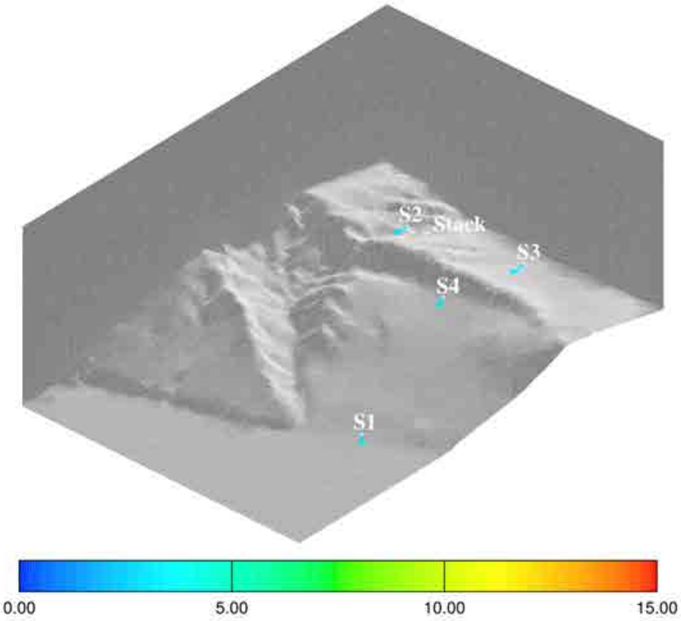
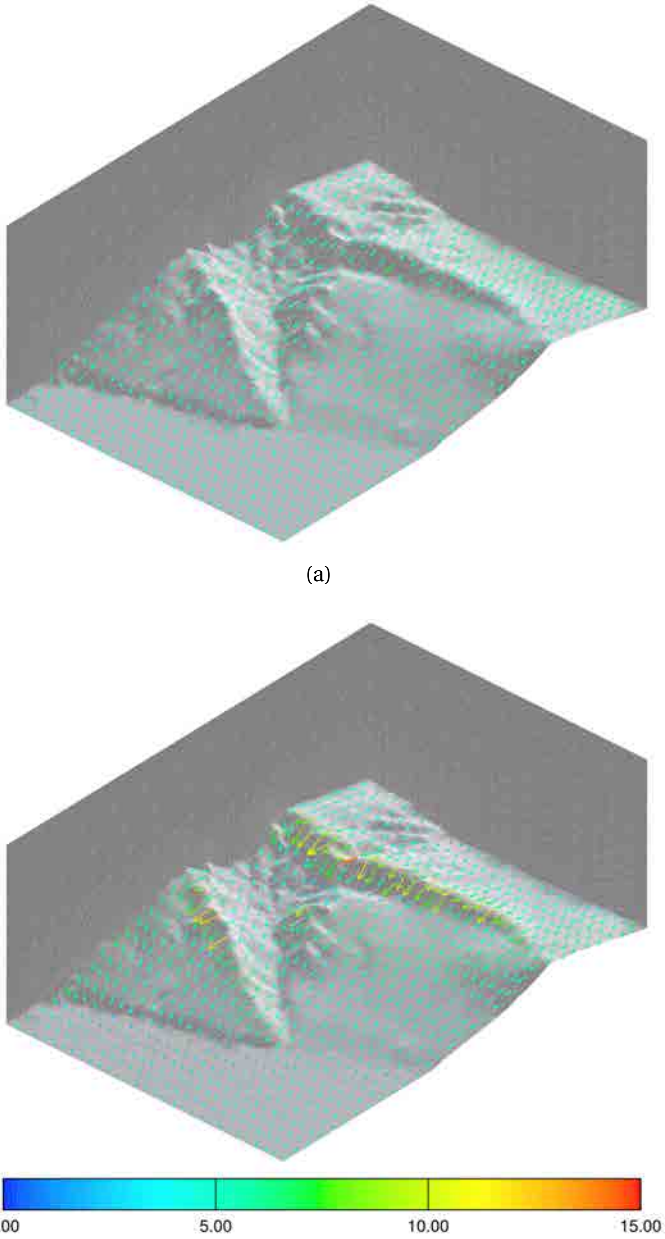


Figure 7.4: Location of the four measurement stations and its corresponding wind velocity. The colourmap represents the wind velocity in ($\mathrm{m\,s^{-1}}$)

Station	$\lvert V\rvert$ ($\mathrm{m\,s^{-1}}$)	$\varphi(V)$ (°)
LPA	3	225
MBI	3	270
MBII	3.5	265
MBIII	3	245

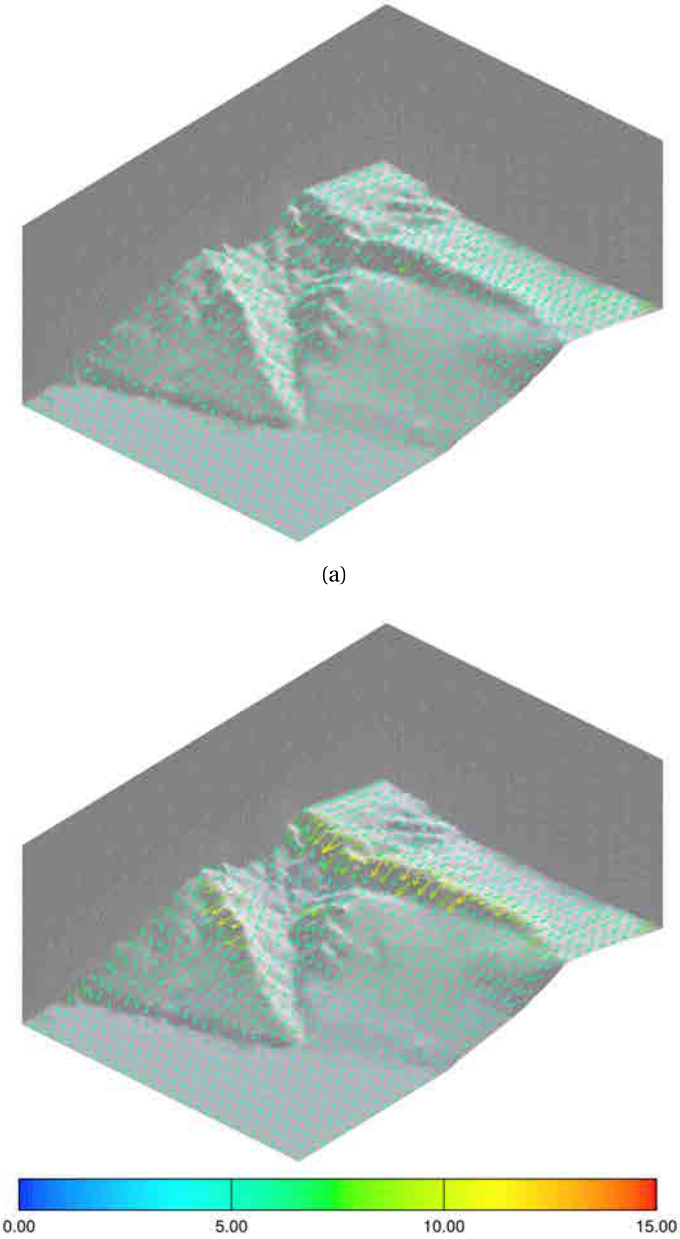
Table 7.1: Measurements of the wind velocity at the four stations



(a)

(b)

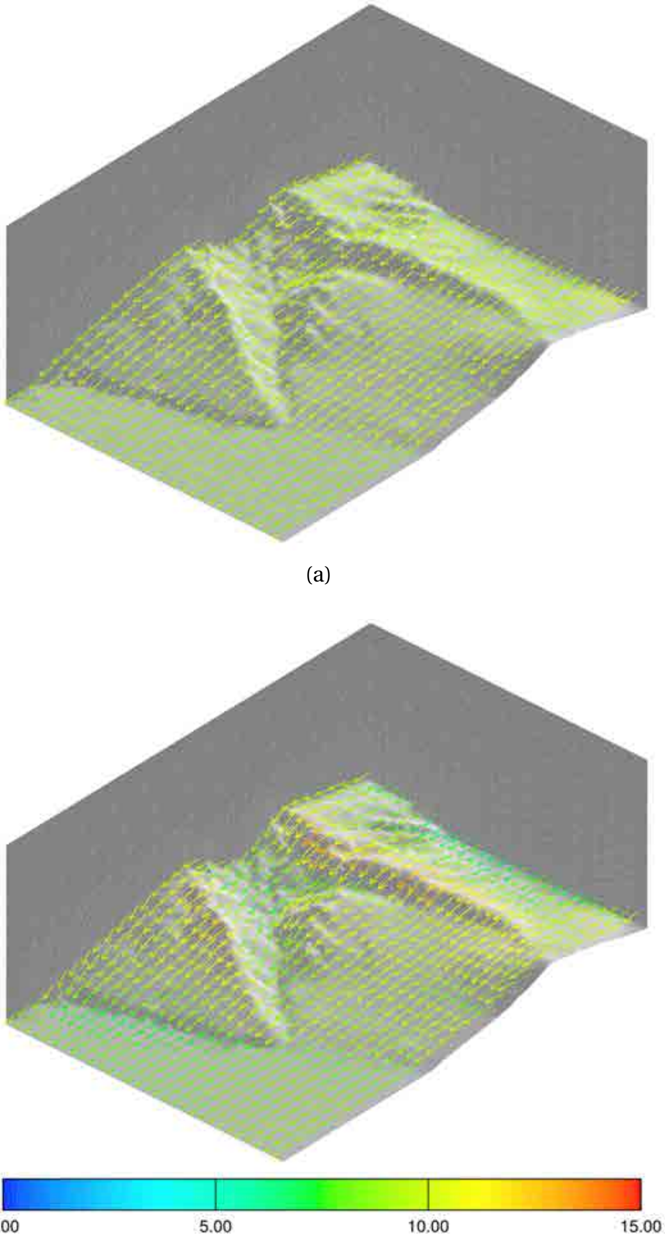
Figure 7.5: Interpolated (a) and resulting (b) wind field at 20 m above ground level. The colourmap represents the wind velocity in $(\mathrm{m\,s^{-1}})$



(a)

(b)

Figure 7.6: Interpolated (a) and resulting (b) wind field at 100 m above ground level. The colourmap represents the wind velocity in $(\mathrm{m\,s^{-1}})$



(a)

(b)

Figure 7.7: Interpolated (a) and resulting (b) wind field at 500 m above ground level. The colourmap represents the wind velocity in $(\mathrm{m\,s^{-1}})$

7.3 Plume rise

The stack emission characteristics are given by a gas velocity of $w_c = 5\,\mathrm{m\,s^{-1}}$ at a temperature of $T_c = 573\,\mathrm{K}$, and the ambient temperature is $T = 298\,\mathrm{K}$.

The velocity of the ambient wind at the top of the stack ($z_c = 1454\,\mathrm{m}$) has a magnitude of $|\mathbf{u}_c| = 7.13\,\mathrm{m\,s^{-1}}$, with horizontal components $u_c = -7.09\,\mathrm{m\,s^{-1}}$ and $v_c = -0.77\,\mathrm{m\,s^{-1}}$. So, a predominant buoyancy rise with a bent plume trajectory results in this case.

With the previous values, using the Briggs' equations, we obtain $z'_c = 1406\,\mathrm{m}$, $F = 5390\,\mathrm{m^4\,s^{-3}}$, $z_H = 2347\,\mathrm{m}$ and $d_f = 3700\,\mathrm{m}$.

The mean trajectory of the plume is then determined by using the method proposed in Section 5.3. Figure 7.8 shows the effect of the local refinement along the Gaussian plume after one (a) and six (b) refinement steps. The final mesh resolution along the plume is enough to capture the transport of pollutants. Note that the plume follows a bent trajectory.

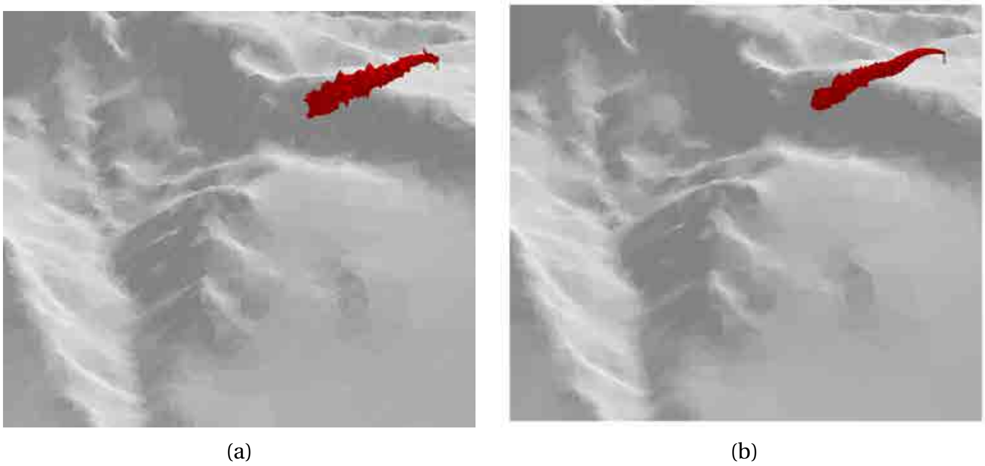


Figure 7.8: Refinement steps along the plume rise: (a) 1 step, (b) 6 steps

Once the mesh has been refined, a new mass-consistent wind field is computed in the new mesh. In order to capture the buoyancy rise effect, the resulting ambient wind field is perturbed modifying its vertical component as proposed in Section 5.6. Figure 7.9 represents the streamlines that start at the top of the stack. These streamlines follow a reasonable bent curve trajectory, that introduces the main physical effects of the emission problem over a complex terrain.

7.4 Air pollution

At this point we can compute the air pollution in the studied region. For the non-linear chemical part, the RIVAD model has been used. We simulate the concentrations of four species $c_1 = [SO_2]$, $c_2 = [SO_4]$, $c_3 = [NO_2]$ and $c_4 = [NO_3]$. The stack emission concentration ($\mathbf{c}^{emi}$) for the primary pollutants are fixed to $c_1^{emi} = c_3^{emi} = 6\,\mathrm{g\,m^{-3}}$, and for the secondary pollutants $c_2^{emi} = c_4^{emi} = 0$. We have considered an horizontal diffusion coefficient of $8 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$ and a vertical diffusion coefficient of $4 \times 10^{-6}\,\mathrm{m^2\,s^{-1}}$ for the four species, corresponding to a mean value for the air. All the four diagonal terms of the deposition matrix ($\mathbf{V}^d$) have been fixed to $1.3 \times 10^{-3}\,\mathrm{m\,s^{-1}}$. In addition, we have fixed null values for vectors $\mathbf{e}$, $\mathbf{c}^{ini}$ and $\mathbf{c}^{out}$. The time-dependent problem has been simulated with a time step $\Delta t = 10\,\mathrm{s}$.

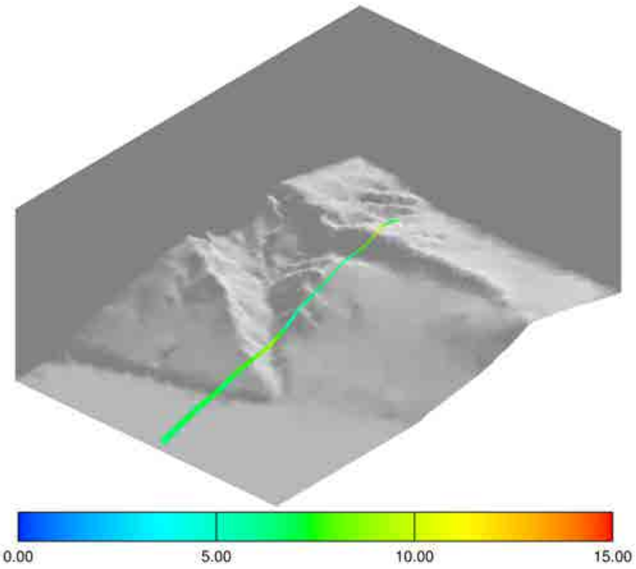


Figure 7.9: Perturbed wind field streamlines from the top of the stack. The colourmap represents the wind velocity in $(\mathrm{m\,s^{-1}})$

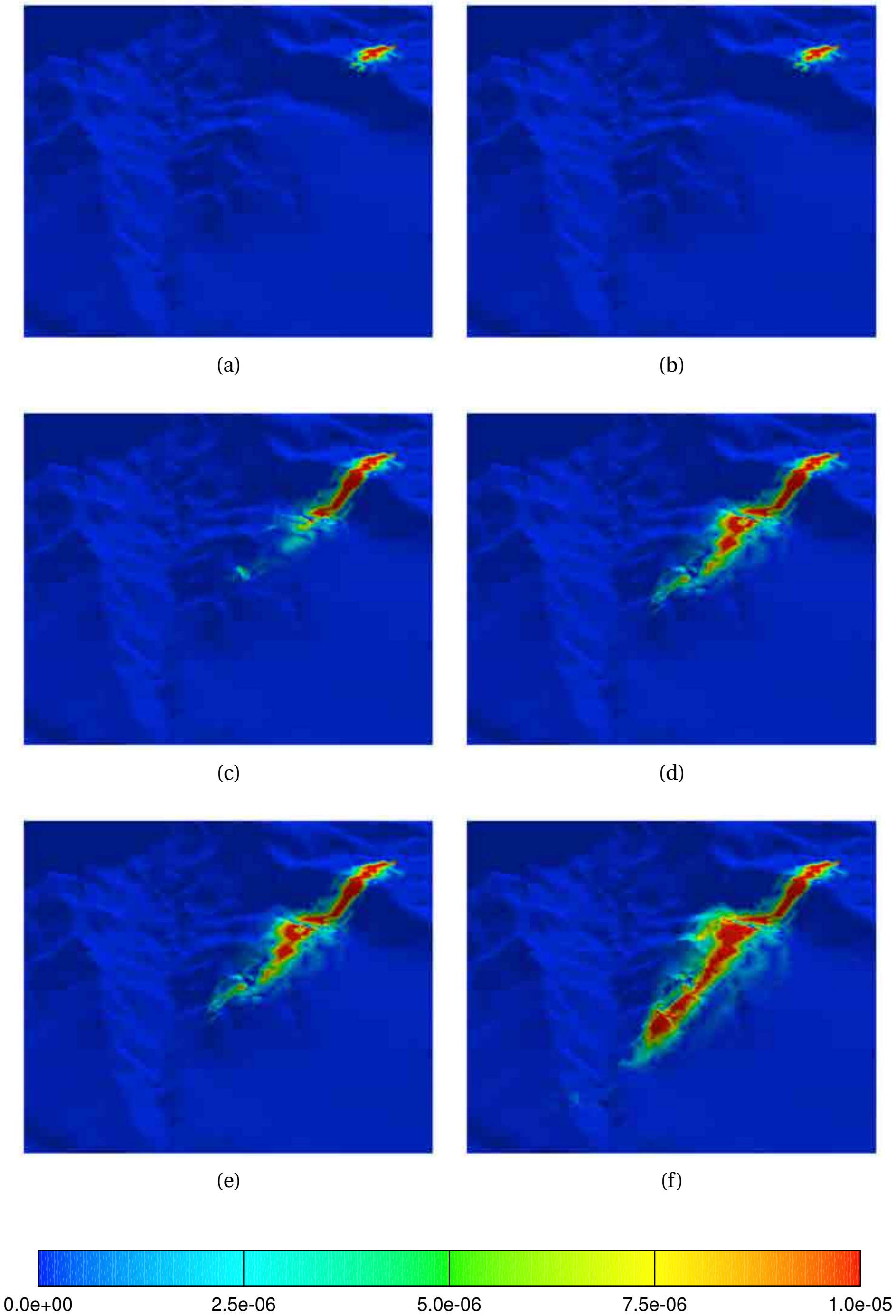
Figures 7.10 and 7.11 show the evolution of the immission concentration distributions for SO_2 and SO_4, respectively. These figures are interesting since them give information about the pollutant concentration at the ground-level. Note that primary pollutant tends to have the highest concentrations near the emission source, while the highest concentrations of the secondary pollutant are located further. This is consistent with the chemical reaction effect.

Figures 7.12 and 7.13 show the evolution of the isosurfaces corresponding to a concentration value of $10\,\mu\mathrm{g\,m^{-3}}$ for SO_2 and SO_4, respectively. Note that the secondary pollutant is more prevalent than the primary one in the front of the plume at the final time steps. It also can be noted that the main effect of the plume dispersion includes the orography, the wind and the emissions.

The evolution of the immission concentration has been studied in five points (A-E). Figure 7.14 shows their locations. These points are grouped in two separate sets. The horizontal distances from the stack to the first two points (A,B) are approximately 100 m and 1000 m, respectively. We note that there are no obstacles between the stack and this first group of points. The second group is formed by three points (C,D,E) that are located at different heights behind the crest of a mountain. The approximated horizontal distance from the stack to this three points are 3000 m,4000 m and 5000 m respectively.

Figure 7.15 shows the evolution of the immission concentration at the points (A-E). It can be observed the influence of the distance to the stack and the orography. The numerical results verifies the qualitative standard behaviour in the first group of points (concentration decreases with the distance). Moreover, reasonable values are reached behind the crest of the mountain. Note that the value of the steady concentration at the deepest point (C) is lower than the values at points D and E.

A final comment about the computational complexity of the evolution process should be done. For each time step we have to solve a finite element problem with a number of degrees of freedom about the number of nodes multiplied by the number of species, i.e. $455953 \times 4 = 1823812$. The number of time steps in the simulation period (30 min) is $30 \times 60/10 = 180$. Therefore, in the whole evolution process about 180 linear equation systems with 1823812 unknowns must be solved. The



(a)

(b)

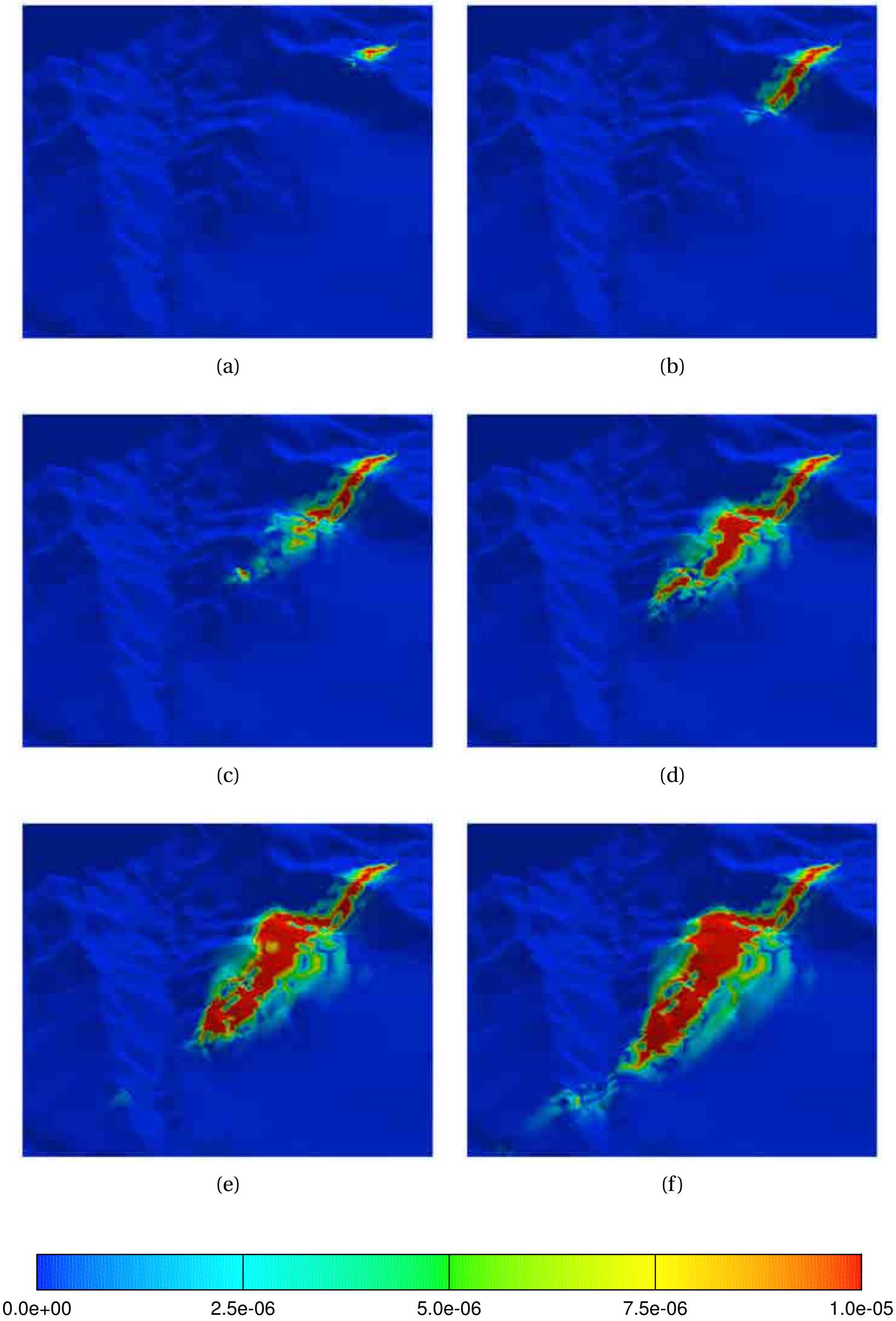
(c)

(d)

(e)

(f)

Figure 7.10: Evolution of the immission concentration distribution ($\mathrm{g\,m^{-3}}$) of the primary pollutant SO_2 after (a) 5 min, (b) 10 min, (c) 15 min, (d) 20 min, (e) 25 min and (f) 30 min



(a)

(b)

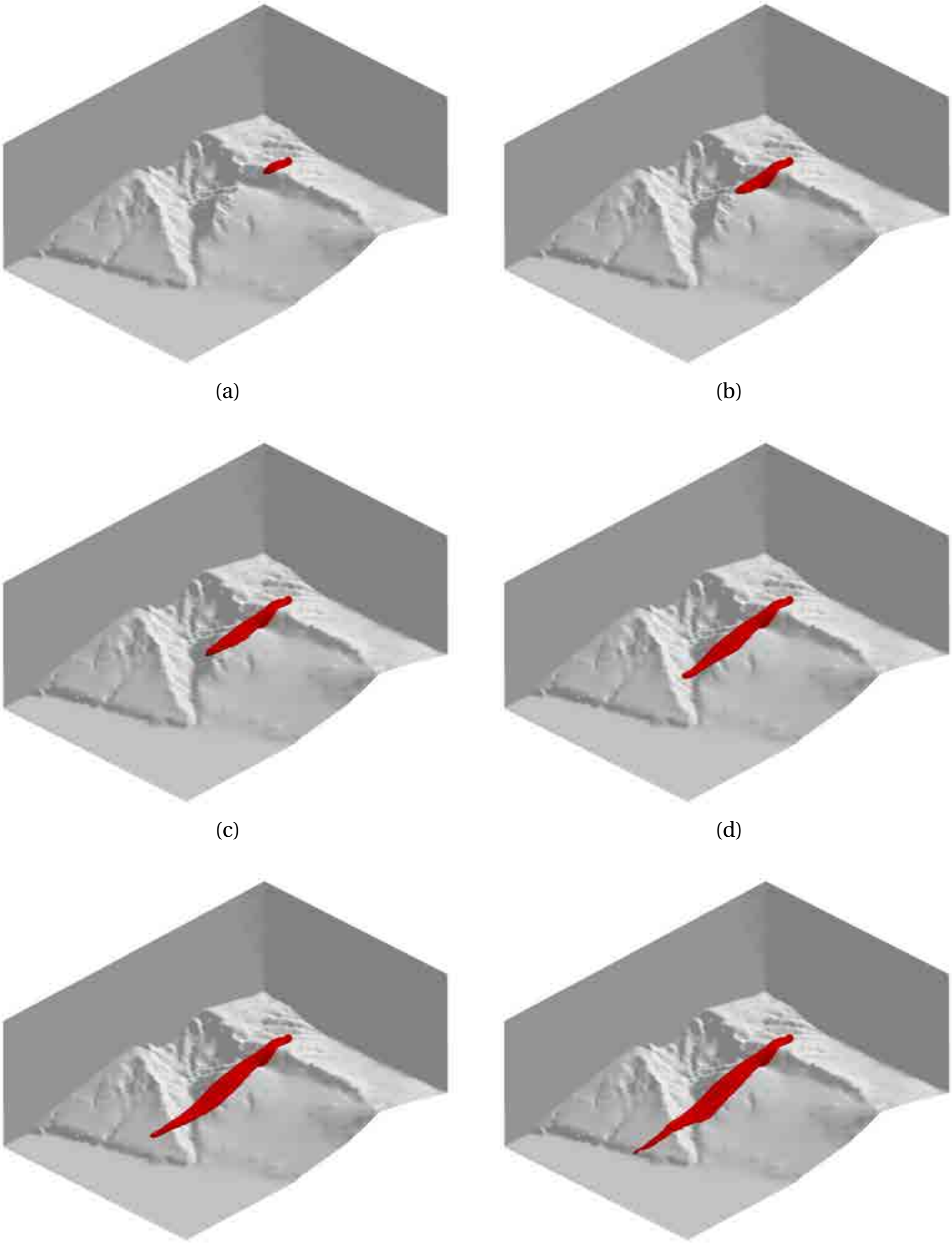
(c)

(d)

(e)

(f)

Figure 7.11: Evolution of the immission concentration distribution ($\mathrm{g\,m^{-3}}$) of the secondary pollutant SO_4 after (a) 5 min, (b) 10 min, (c) 15 min, (d) 20 min, (e) 25 min and (f) 30 min



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.12: Evolution of the isosurface corresponding to a concentration of 10 µg m^{-3} for the primary pollutant SO_2 after (a) 5 min, (b) 10 min, (c) 15 min, (d) 20 min, (e) 25 min and (f) 30 min

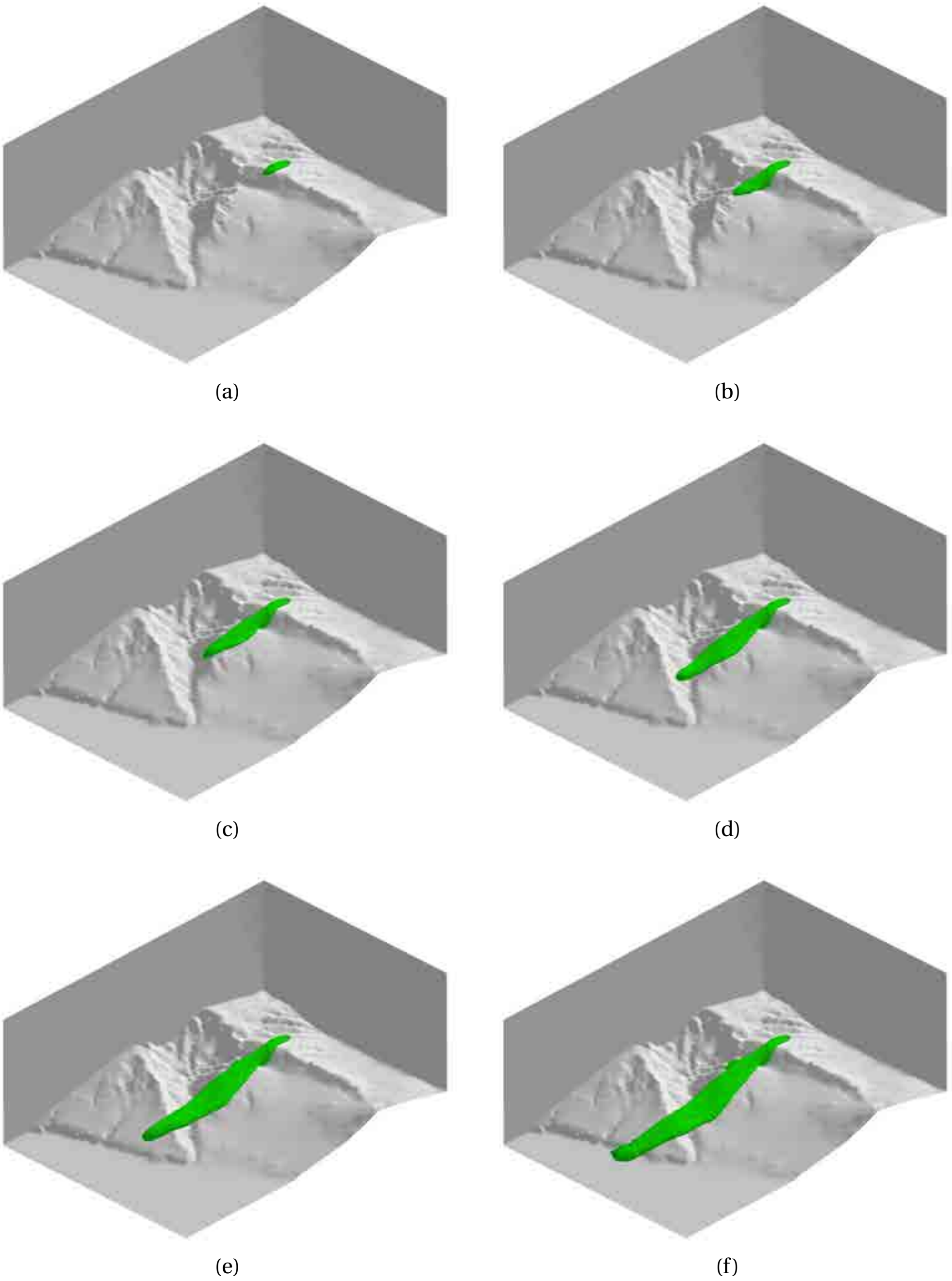

(a) (b) (c) (d) (e) (f)

Figure 7.13: Evolution of the isosurface corresponding to a concentration of 10 µg m^{-3} for the secondary pollutant SO_4 after (a) 5 min, (b) 10 min, (c) 15 min, (d) 20 min, (e) 25 min and (f) 30 min

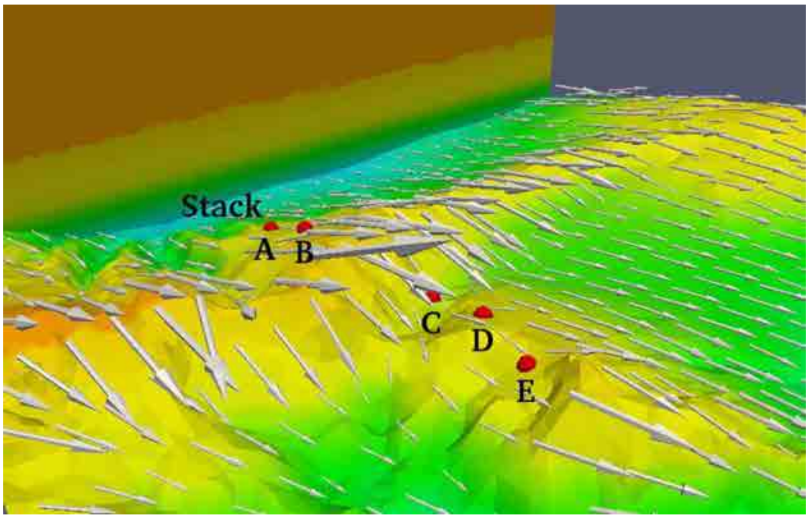


Figure 7.14: Wind field at 10 m above ground level, and location of the stack and the points A-E

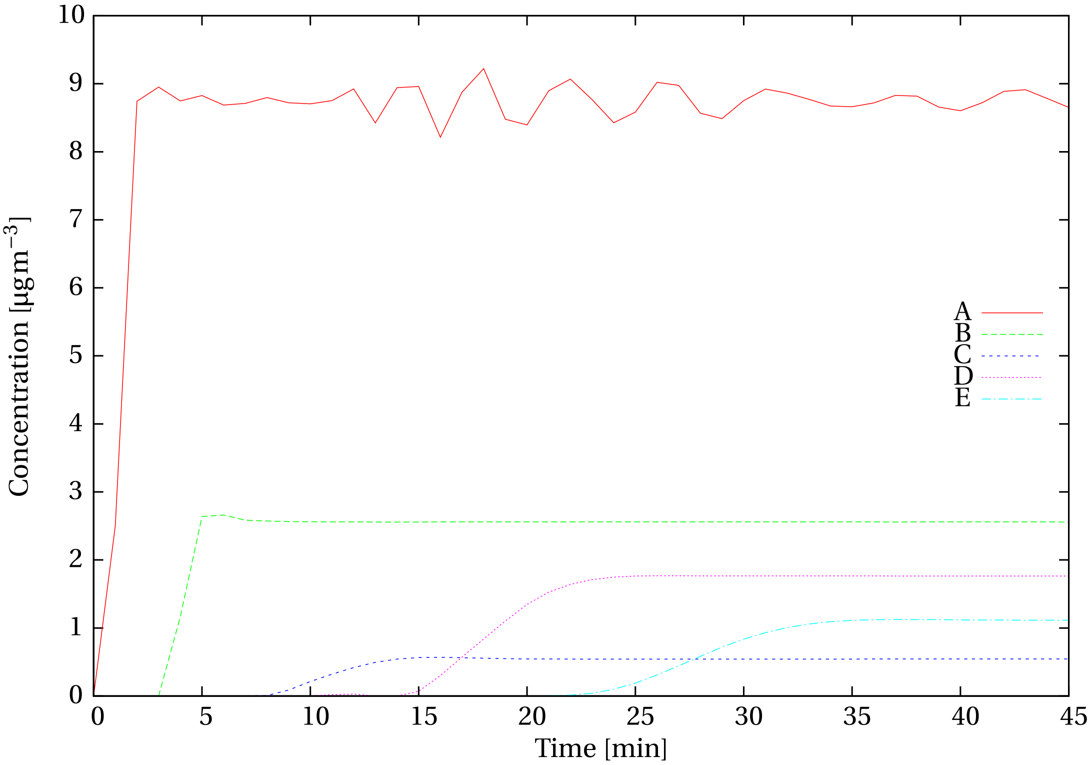


Figure 7.15: Immission concentration evolution of the primary pollutant at points A-E

computational cost corresponding to the mesh generation, wind simulation, and the resolution of the ordinary differential equations in the splitting method are insignificant with respect to the resolution of the unsteady convection–diffusion equation. In a computer with 128GB of RAM memory and 2.34 GHz, the total computing time is about 40 min.

8 Forecast application: CMAQ benchmark test case

In this Chapter, we present a forecast simulation using data from the CMAQ system. We will use the data from the CMAQ benchmark test [CMAQ Operational Guide, 2012]. The local simulation has been performed in the surroundings of Pineville Kentucky, where we simulate the transport and reaction of pollutants from two fictitious chimneys. We will simulate an episode starting at midnight on 22 July 2001 until noon, a 12 hour simulation. The development of this application has been described in Pérez-Foguet and Oliver [2008], Oliver and Pérez-Foguet [2012], Montenegro et al. [2013], and Oliver et al. [2015a].

8.1 Adaptive tetrahedral mesh

The domain of interest is a rectangular area with dimensions of about 16 km × 16 km. The topography of the terrain has different ridges and valleys, ranging from 200 m up to a maximum height of 800 m. The upper boundary of the domain has been placed at $h = 3000$ m above sea level. The terrain is described using the digital elevation model (DEM) produced in the shuttle radar topography mission [Farr et al., 2007]; specifically we have used the SRTM1 version defined over an uniform grid of $1'' \times 1''$ (approximately 30 m × 30 m), with a linear vertical absolute height error of less than 16 m.

Two identical stacks have been considered. The chimneys have a height of 150 m above ground level and the diameter at its top is 15 m. One stack has been placed in the Pineville city area, at coordinates 36°45′53″N 83°41′56″W, and the other one is located in the Fourmile community, at coordinates 36°47′31″N 83°44′31″W.

The mesh has been created using the curvature based method explained in Section 3.2. The required precision (ε) is equal to 0.1. The minimum element size of the mesh is restricted to 200 m except near the chimneys where the minimum size is 0.5 m necessary to capture the geometry of the stacks. Regarding the vertical distribution we have considered no internal layers, so the nodes between the terrain and the top will be the result of generating a quality mesh using Tetgen.

Figure 8.1 shows the resulting tetrahedral mesh, with 15 767 nodes and 56 536 tetrahedra. It captures the valleys and mountains of the terrain, and we can see that, vertically, the nodes have a higher density near the terrain that on top due to the difference in element size; this characteristic is a good one for our problem because one of the outputs of interest is the concentration near the surface. Figure 8.2 shows a detail of the surroundings of one the stacks where we can realize that the element size is fitted for the geometry of the stack. In Fig. 8.1 we cannot distinguish the mesh elements in the surroundings of the stacks, but in Fig. 8.2 we appreciate that there is an smooth

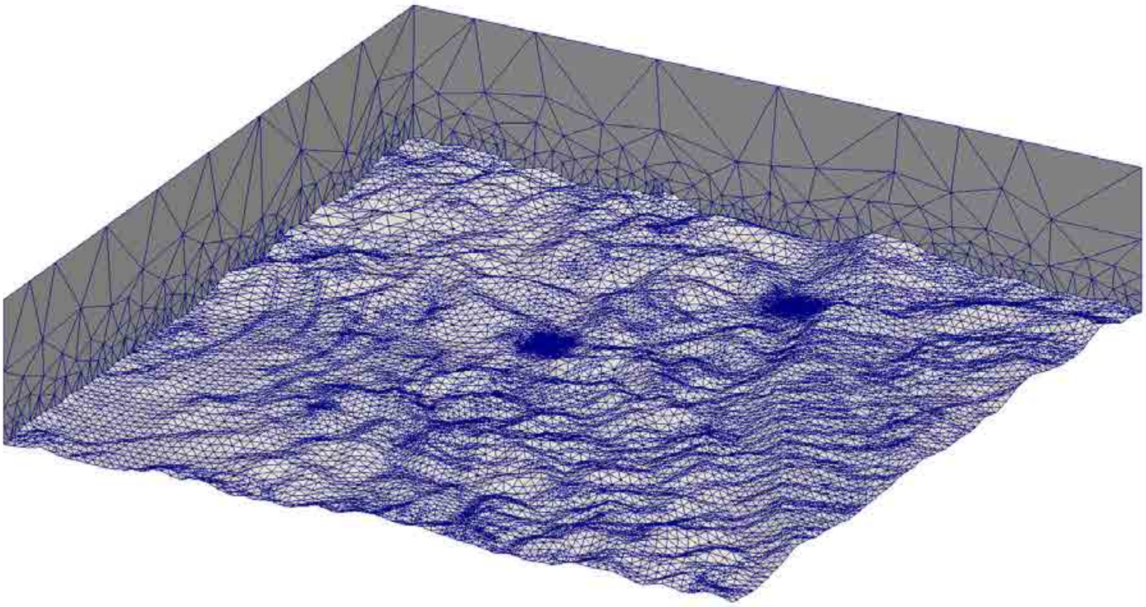
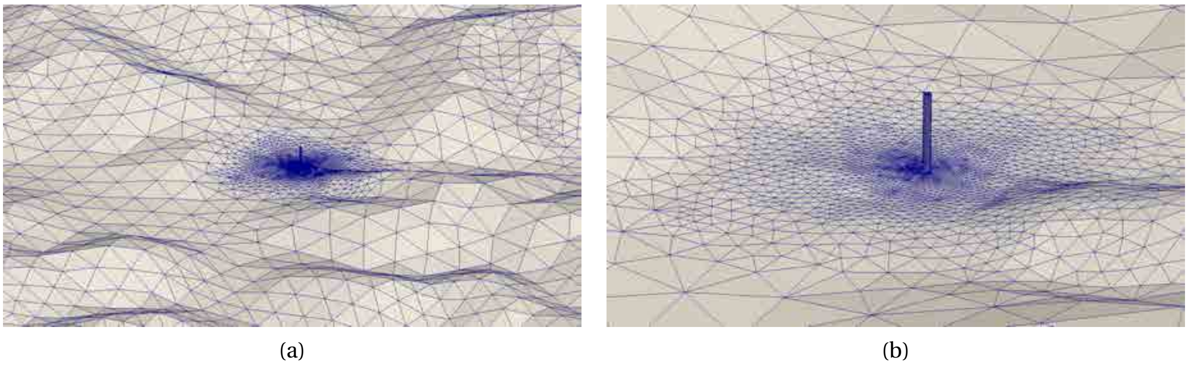


Figure 8.1: Tetrahedral mesh



(a)

(b)

Figure 8.2: Detail of the stacks (a) general view (b) detail view

transition from the element size needed in the terrain and the element size required to discretize the chimney.

Figure 8.3 shows the distribution of the mesh element quality using the mean ratio algebraic shape quality metric. We can observe that the number of elements below the quality 0.2 is low, meaning it is a suitable mesh for the simulation.

8.2 CMAQ interpolation

Once we have generated a mesh we have to interpolate the CMAQ data into this mesh. This step uses the method described in Section 2.3, where the nodes of the mesh are located in the CMAQ mesh and the new values are interpolated from the values in the CMAQ cell.

Figure 8.4 shows the interpolation of the temperature and pressure fields. They are very similar

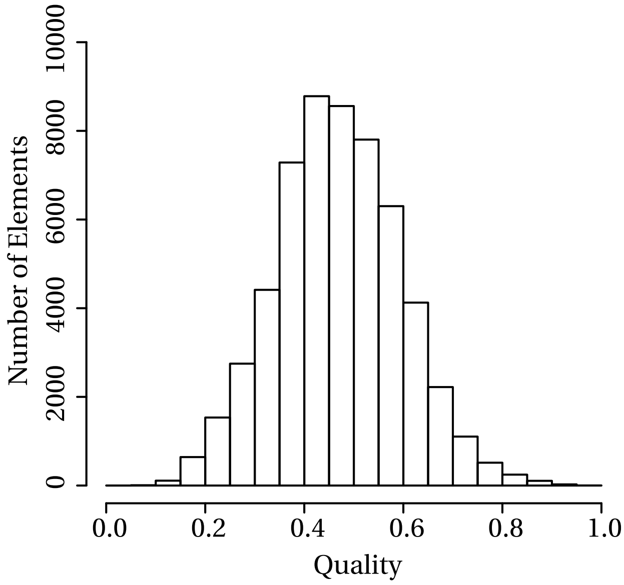
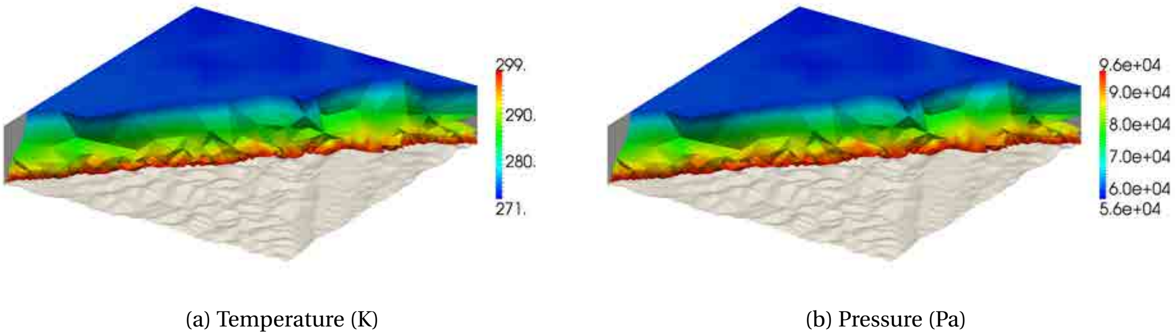


Figure 8.3: Mesh quality histogram



(a) Temperature (K)

(b) Pressure (Pa)

Figure 8.4: Interpolation from CMAQ of temperature and pressure fields

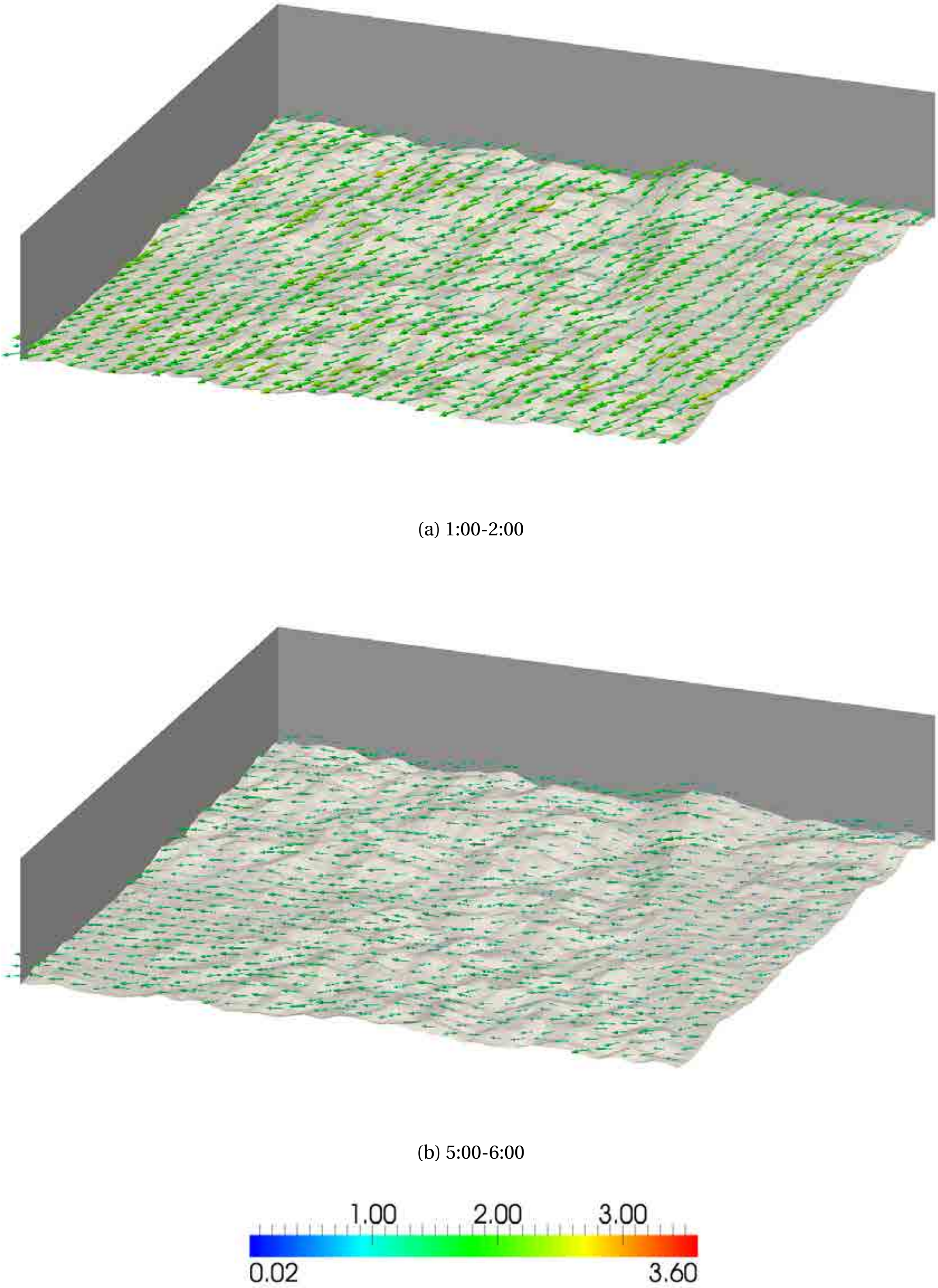
because below the tropopause both fields have a comparable profile that decrease when the height increases.

8.3 Ambient wind field

The wind field is constructed from the CMAQ wind field data. This data has hourly records, so we will compute a resulting wind field per hour that will be used in the simulation for the whole hour. To interpolate the wind field we will use the method described in Section 2.3. Once the interpolated wind field is computed, we impose the mass-consistent model of the wind field described in Section 4.2 considering $\alpha = 1.0$.

Figure 8.5 show the resulting ($\mathbf{u}$) ambient wind field at a height of 200 m above ground level at 1, 5, 7 and 11 hours. As stated in the diagnostic application (Section 7.2), the terrain has an strong effect on the resulting wind field, verifying the conditions of the mass-consistent model.

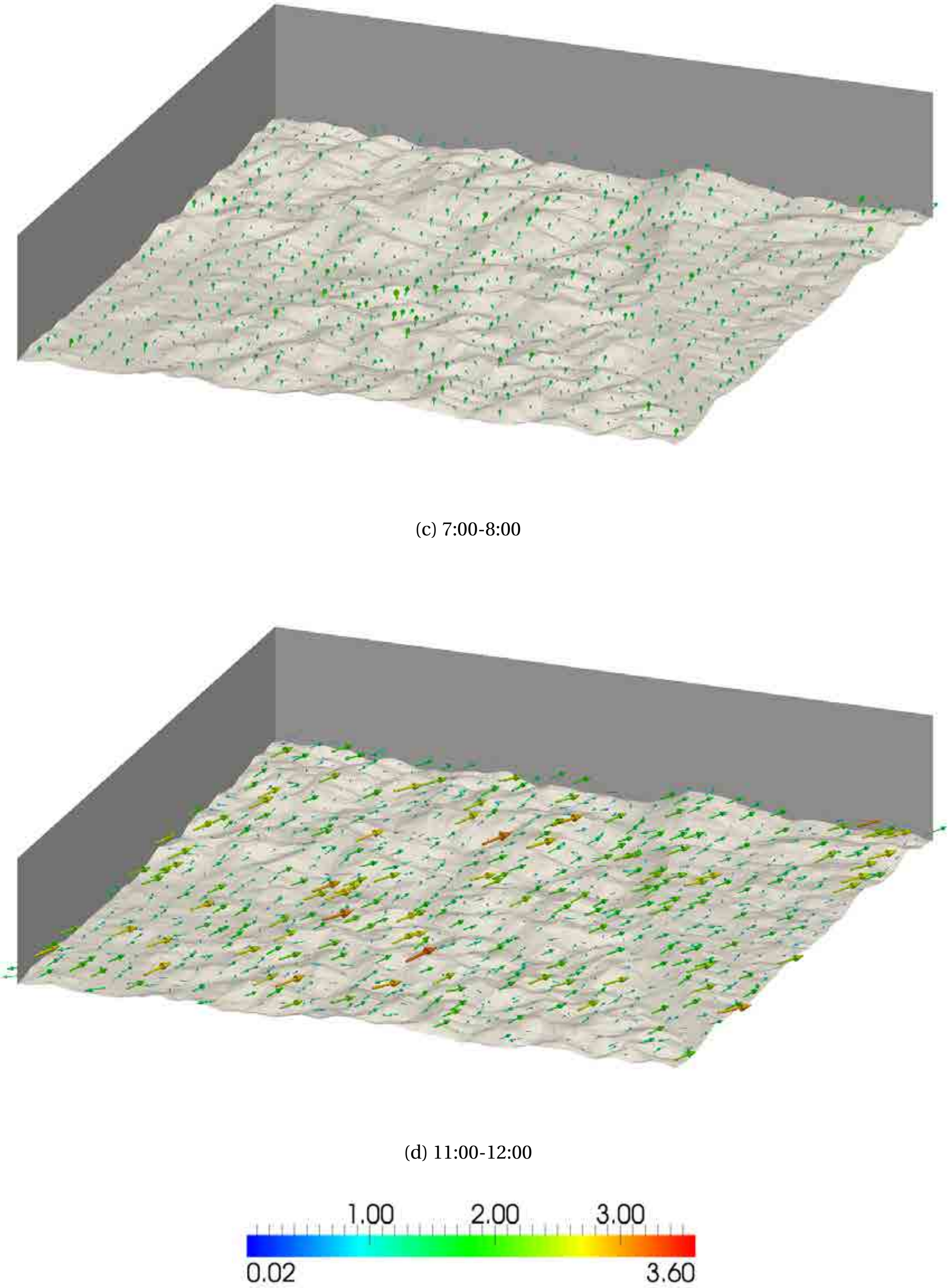
The wind field during this simulation, as can be observed in Fig. 8.5, changes velocity and di-



(a) 1:00-2:00

(b) 5:00-6:00

Figure 8.5: Resulting wind field at 200 m above ground level in different hours. The colourmap represents the wind velocity in $\mathrm{m\,s^{-1}}$



(c) 7:00-8:00

(d) 11:00-12:00

Figure 8.5: Resulting wind field at 200 m above ground level in different hours. The colourmap represents the wind velocity in $\mathrm{m\,s^{-1}}$

rection. At midnight, the velocity is about $2\,\mathrm{m\,s^{-1}}$ and then it decreases being really small between 4 and 8, from there it increases to its maximum at about $3.5\,\mathrm{m\,s^{-1}}$ at noon. The direction changes gradually between North-west to South-east clockwise. This change in velocity will be important in the plume rise computation.

8.4 Plume rise

In this section we compute the trajectory of the plume rise. In this case, both stacks have the same emissions characteristics: the velocity of the gas is $w_c = 5\,\mathrm{m\,s^{-1}}$ and its temperature $T_c = 573\,\mathrm{K}$. The velocity and the temperature of the ambient wind at the top of the stack varies every hour, depending on the data from CMAQ. The mean trajectory of the plume is then determined by using the Briggs trajectory method proposed in Section 5.3.2 taking into account the ambient wind field.

With these emissions, the trajectory of the plume rise will be momentum predominant from 4:00 to 8:00, and buoyancy predominant in the rest of the hours.

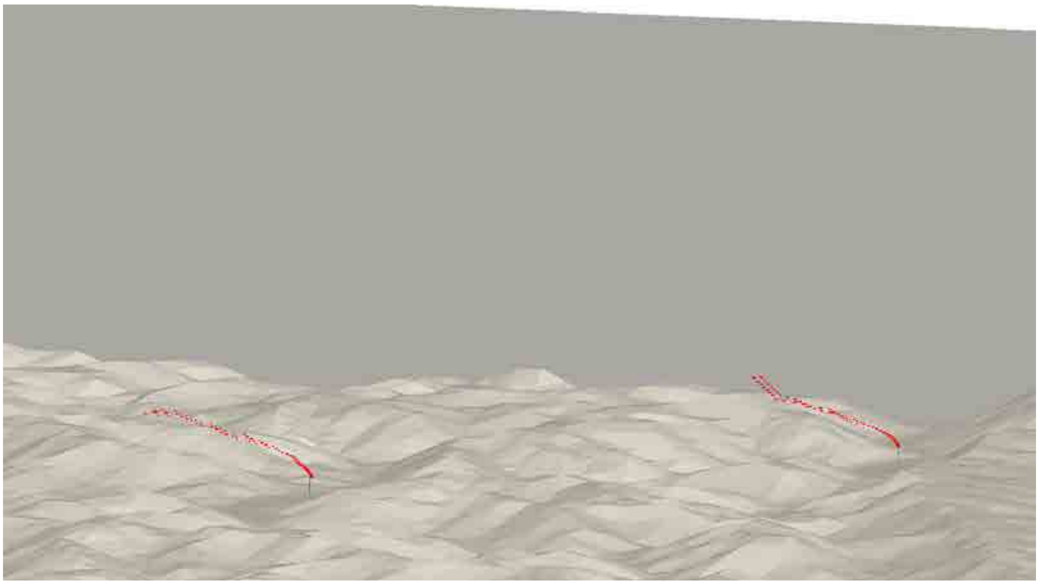


Figure 8.6: Plume rise trajectory at noon

Figure 8.6, and 8.7 show the plume rise trajectory at noon and at 11:00, respectively. Both trajectories are different because the wind and temperature fields are different. In Fig. 8.6 we can note that the vertical trajectory bends according to the ambient wind field; in Fig. 8.7 we can view the same effect on the horizontal trajectory.

Once the plume trajectory is defined we have to introduce it into the mesh. Figure 8.8 shows the mesh generated at Section 8.1 and the mesh refined capturing the plume rise at 3:00. This new mesh has 52 034 nodes and 255 101 elements. This mesh size will be similar for all the meshes with a buoyant plume rise, while the meshes with a momentum rise will have 25 432 nodes and 94 198 elements.

Finally, the vertical component of the wind field is modified to introduce the effect of the injection.

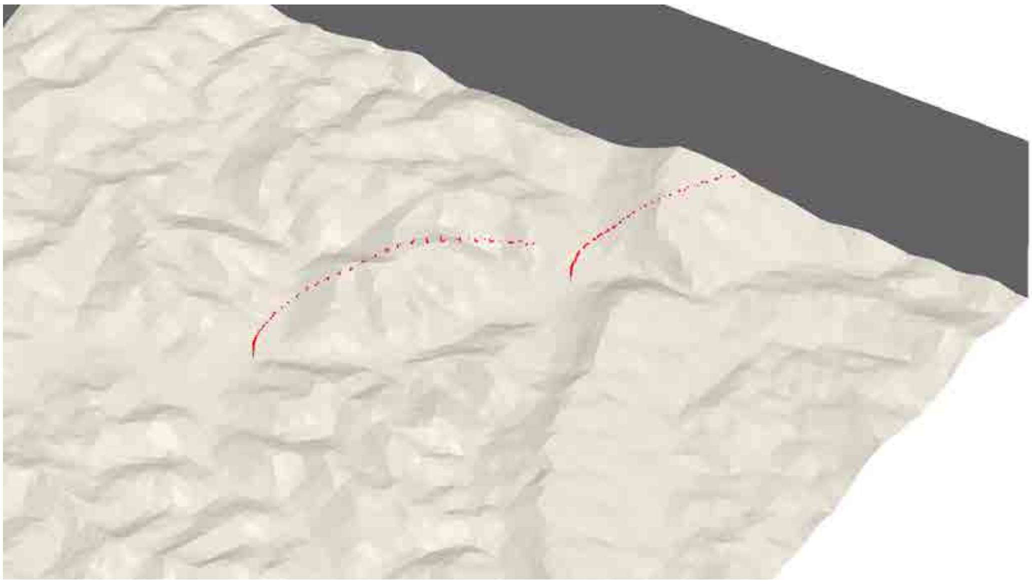
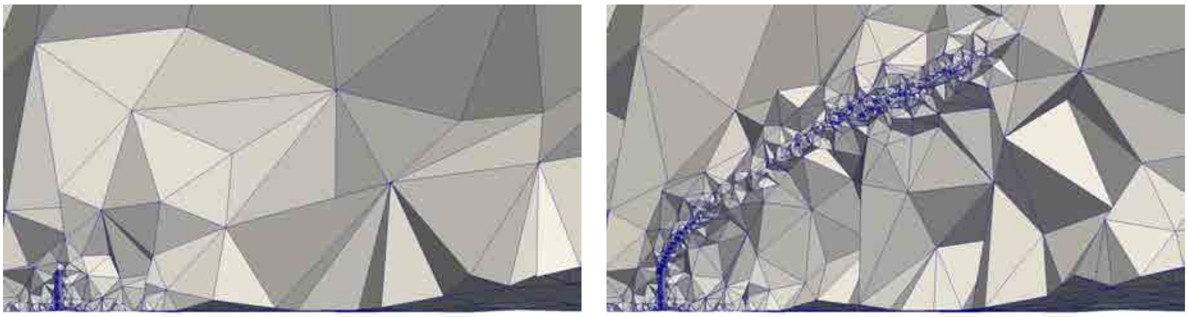


Figure 8.7: Plume rise trajectory at 11:00



(a) Original mesh

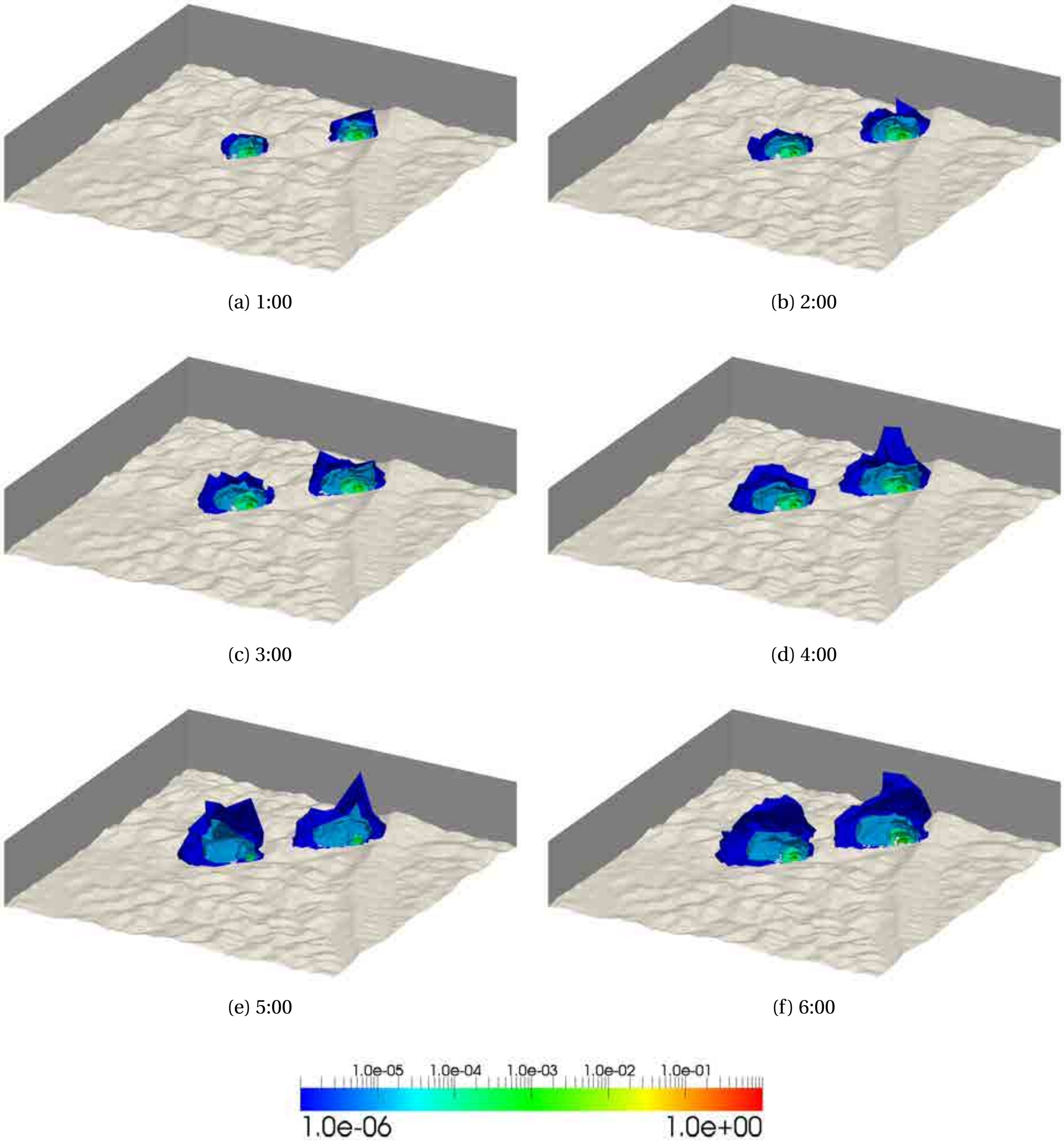
(b) Refined mesh

Figure 8.8: Mesh refinement around the plume rise at 3:00

8.5 Air pollution

At this point we have generated the meshes and fields (wind, temperature, pressure, initial conditions, etc.) for every hour of the simulation. We will simulate the transport and reaction of six chemical species that contain oxygen and nitrogen (i.e. NO_2, NO, O, O_3, NO_3, N_2O_5), following the CB05-06 example presented in Monforte and Pérez-Foguet [2014b]. To deal with the emissions, in this case, we will use a Neumann boundary condition. The emission rate in both stacks is the same, emitting $10\,\mathrm{g\,m^{-2}\,s^{-1}}$ of oxygen and $5\,\mathrm{g\,m^{-2}\,s^{-1}}$ of NO. The initial and boundary conditions are fixed using the concentration values of the interpolated CMAQ data. The diffusion term has a horizontal value of $100\,\mathrm{m^2\,s^{-1}}$, and the vertical diffusion is computed using Shir [1973] formulas for neutral conditions [Monforte and Pérez-Foguet, 2014b]. The terms of the deposition matrix ($\mathbf{V}^d$) have been fixed to $1.3 \times 10^{-3}\,\mathrm{m\,s^{-1}}$. The time-dependent problem has been simulated with a time step $\Delta t = 5\,\mathrm{s}$.

This problem has been solved using the multimesh scheme, so we have used 6 meshes, one for each component. The adaptation step is set to 10 min. The first adaption step of each hour is recomputed in order to capture the influence of the change of the reference mesh, the wind field, and the boundary conditions between hours.



(a) 1:00

(b) 2:00

(c) 3:00

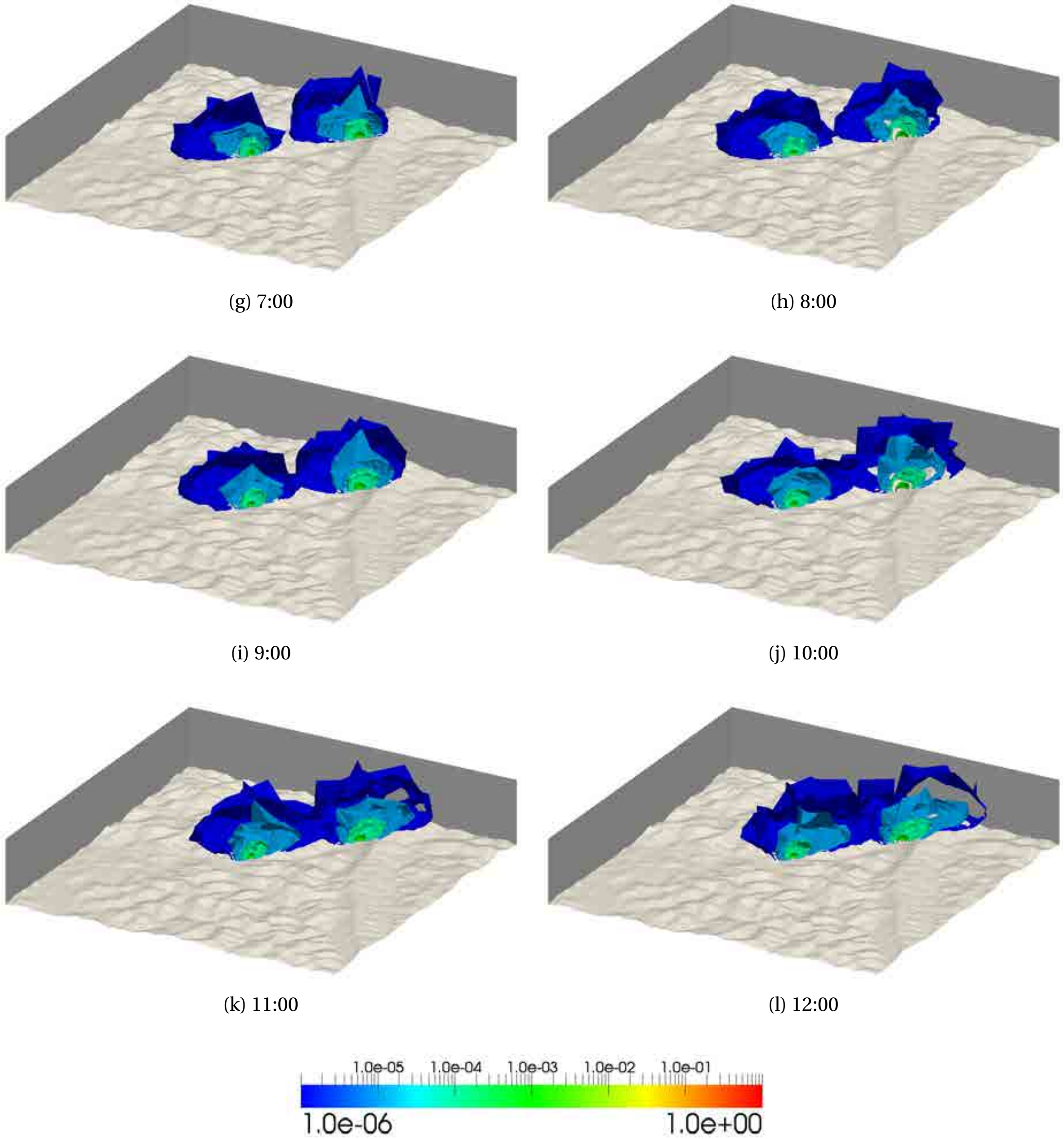
(d) 4:00

(e) 5:00

(f) 6:00

Figure 8.9: Evolution of different isosurfaces corresponding to the concentration of oxygen. The colourmap represents the concentration in $\mathrm{g\,m^{-3}}$

Figure 8.9 shows the evolution of the concentration of oxygen. We can observe the isosurfaces ranging from $1.0 \times 10^{-6}\,\mathrm{g\,m^{-3}}$ to $1\,\mathrm{g\,m^{-3}}$. We can observe how the plume of the concentration changes direction accordingly to the wind field direction observed in Section 8.3. Between 4 and 8,



(g) 7:00

(h) 8:00

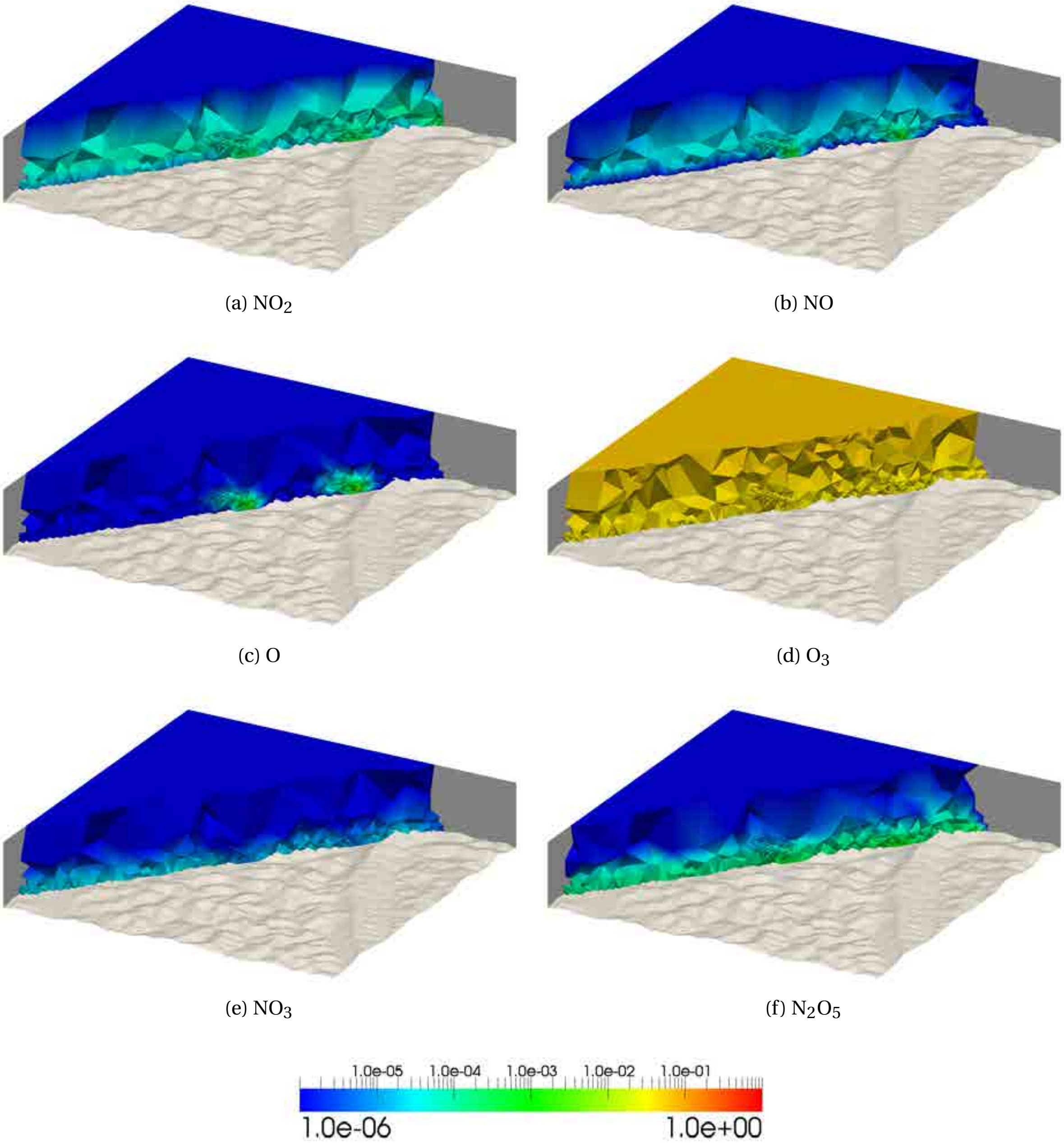
(i) 9:00

(j) 10:00

(k) 11:00

(l) 12:00

Figure 8.9: Evolution of different isosurfaces corresponding to the concentration of oxygen. The colourmap represents the concentration in $\mathrm{g\,m^{-3}}$



(a) NO_2

(b) NO

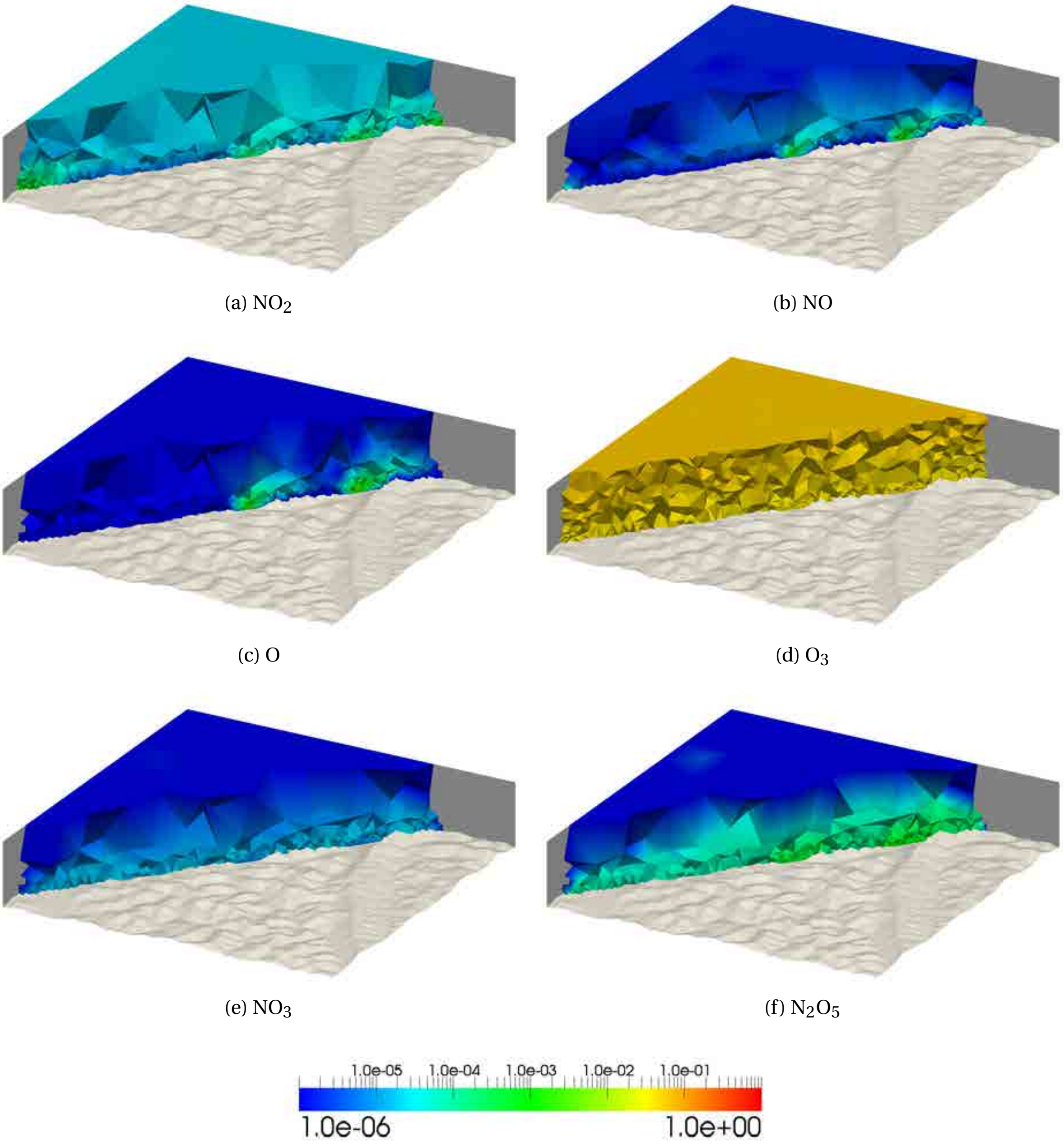
(c) O

(d) O_3

(e) NO_3

(f) N_2O_5

Figure 8.10: Concentration of the different chemical species at 3:00. The colourmap represents the concentration in $\mathrm{g\,m^{-3}}$



(a) NO_2

(b) NO

(c) O

(d) O_3

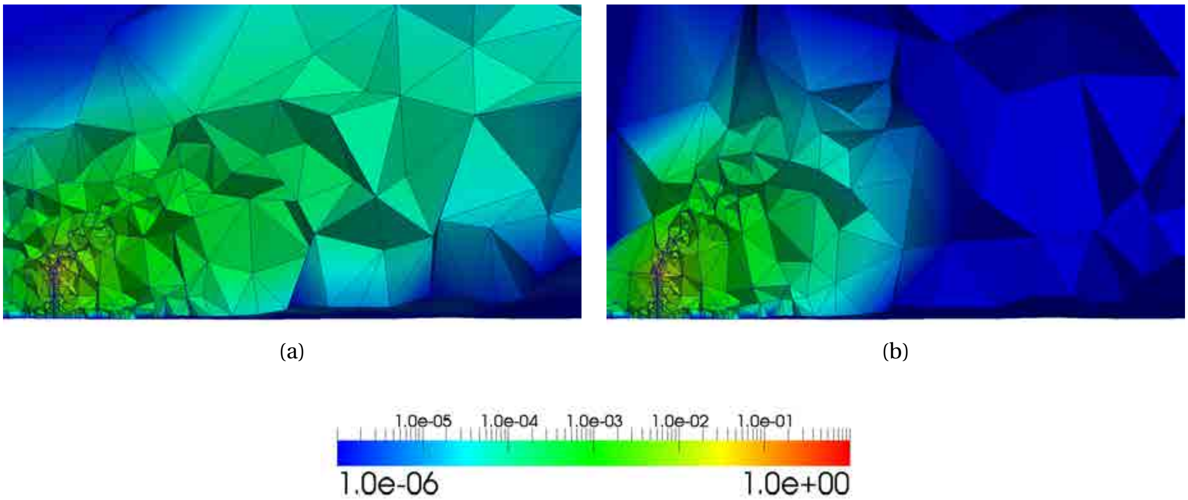
(e) NO_3

(f) N_2O_5

Figure 8.11: Concentration of the different chemical species at 12:00. The colourmap represents the concentration in $\mathrm{g\,m^{-3}}$

we can also see that the diffusion plays an important role since the wind velocity is small. From 9 the velocity is higher and the plumes from both stacks overlap.

Figures 8.10 and 8.11 show the different chemical species at 3 and 12. NO and O are the species that are emitted; if we look closer at NO we can see how the ambient concentration that can be seen at 3 has disappeared at 12. O_3 do not react with any specie and the concentration is almost steady. Near the emitters, NO_3, specially at 3, gets exhausted; while NO_2 and N_2O_5, on the contrary, is created due to chemical reactions.


(a)

(b)

Figure 8.12: Solution before the adaptation (a) and after the adaptation (b) for oxygen at 1:10. The colourmap represents the concentration in $\mathrm{g\,m^{-3}}$

Figure 8.12 shows the solution before adapting the mesh, and after adapting and recalculating the mesh. We can observe that the mesh has been refined in a different trajectory resembling the new trajectory of the plume rise.

This problem has been solved in a cluster using 6 Hexa-Core Xeon L5640 CPU. The total time for solving the 12 hours of the simulation has been 10 h and 30 min. Thus, the ratio computing time divided by simulated real time is slightly less than 1. This ratio is similar to the one obtained in the diagnostic example (Chapter 7), slightly over one. This time is promising for producing an operational system.

9 Conclusions and future work

We have introduced a new methodology for solving air pollution problems over a complex terrain. The main contribution of the work is the construction of a system to model the air quality in the local scale, and specially in zones with a complex orography. This system consists of the following parts that are described briefly.

- *Adaptive three-dimensional mesh generation.* We have proposed two different strategies to discretize domains defined over complex terrains including several stack geometries, with a minimal user intervention and low computational cost.

- *Mass-consistent wind field computation.* The mass-consistent model obtains an ambient wind field that takes into account the complex orography.

- *Plume rise.* The local mesh refinement along the streamlines of the injection of the emission into the ambient wind allows to simulate the influence of the plume initial rise and its development near the emitter.

- *Convection–diffusion–reaction equation.* An stabilized finite element method has been proposed to solve the equation obtaining the values of concentration for all the pollutants in the whole three-dimensional domain.

- *Multimesh adaptive scheme.* To improve the solution an h-adaptation has been used, using a mesh for every specie. The proposed methodology uses the necessary mesh resolution to consider the main effects of the physical phenomena that are involved in air pollution process.

With all this components we have built a system able to solve a diagnostic problem (Chapter 7), and a predictive problem (Chapter 8).

9.1 Future work

- *Operational numerical air quality forecasting system.* The main description of an operational system is one that can be routinely used for both operational and research applications. Usually it is executed every once or twice per day with a forecast length of 48 to 72 hours. To this end, all of the interactions with the CMAQ model and the coupling between all the processes has to be automatized. A user has to be able to choose the outputs of interest and final values. So, a system of scripts, executables, support libraries, documentation and tools have to be provided to the final user.

- *Parallelization.* To provide an operational system the simulation time has to be the be reduced enough to effectively operate the system. To reduce this simulation time parallelization of the code is a good idea. Several parallelization opportunities arise in this problem; e.g. chemical reactions can be solved node by node separately, domain decomposition of the mesh, etc. To choose the right parallelization opportunities, it is important to study the scalability of the code.

- *Application and validation in real scenarios.* Application of the local scale system in a real scenario gives the opportunity to validate and calibrate the model. We can also monitor and dynamically control emitters to fulfil environmental regulations. To do this we can take advantage of of wind pattern analysis [Pérez-Foguet, 2014] and to perform the prediction of Air Quality using improved downscaled meteorological predictions. The model can be adapted to special scenarios and specific physical models can be added.

Bibliography

P. H. Abelson. Air pollution and acid rain. *Science*, 230(4726):617–617, 1985. doi:10.1126/science.230.4726.617. [Cited on page 1]

N. N. Ahmad, D. P. Bacon, M. S. Hall, and A. Sarma. Application of the multidimensional positive definite advection transport algorithm (MPDATA) to environmental modelling on adaptive unstructured grids. *International Journal for Numerical Methods in Fluids*, 50(10):1247–1268, 2006. doi:10.1002/fld.1113. [Cited on page 2]

M. O. Andreae and P. J. Crutzen. Atmospheric aerosols: Biogeochemical sources and role in atmospheric chemistry. *Science*, 276(5315):1052–1058, 1997. doi:10.1126/science.276.5315.1052. [Cited on page 42]

R. Atkinson and J. Arey. Atmospheric degradation of volatile organic compounds. *Chemical Reviews*, 103(12):4605–4638, 2003. doi:10.1021/cr0206420. [Cited on page 42]

J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 14–21, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc. ISBN 0-8058-0158-8. URL `http://dl.acm.org/citation.cfm?id=42512.42515`. [Cited on page 30]

M. Bazaraa, H. Sherali, and C. Shetty. *Nonlinear Programing: Theory and Algorithms*. John Wiley and Sons, Inc., New York, 1993. doi:10.1002/0471787779. [Cited on page 20]

A. Bott. A positive definite advection scheme obtained by nonlinear renormalization of the advective fluxes. *Monthly Weather Review*, 117:1006–1016, 1989. doi:10.1175/1520-0493(1989)117<1006:APDASO>2.0.CO;2. [Cited on page 7]

R. Boubel, D. Fox, D. Turner, and A. Stern. *Fundamentals of Air Pollution*. Academic Press, San Diego, 1994. [Cited on page 33]

C.-A. Bourque and P. Arp. Simulating sulfur dioxide plume dispersion and subsequent deposition downwind from a stationary point source: A model. *Environmental Pollution*, 91(3):363–380, 1996. doi:10.1016/0269-7491(95)00041-0. [Cited on page 1]

J. P. Boyd. Multiscale numerical algorithms for weather forecasting and climate modeling: Challenges and controversies. *SIAM News*, 41(9), 2008. URL `http://www.siam.org/news/news.php?id=1456`. [Cited on page 1]

G. A. Briggs. Plume rise. Technical Report TID-25075, U.S. Atomic Energy Commission, Division of Technical Information, 1969a. URL http://www.osti.gov/scitech/biblio/4743102-17XJH8. doi:10.2172/4743102. [Cited on pages 31 and 32]

G. A. Briggs. Optimum Formulas for Buoyant Plume Rise. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences (1934-1990)*, 265(1161):197–203, 1969b. URL http://www.jstor.org/stable/73726. [Cited on page 31]

J. A. Businger and S. Arya. The height of the mixed layer in the stably stratified planetary boundary layer. *Advances in geophysics*, 18(A):73–92, 1974. doi:10.1016/s0065-2687(08)60453-7. [Cited on page 42]

D. Byun and J. Ching. Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System. Technical Report EPA/600/R-99/030, U.S. Environmental Protection Agency, Office of Research and Development, 1999. URL http://www.epa.gov/asmdnerl/CMAQ/CMAQdocumentation.html. [Cited on pages xiii, 2, 7, and 8]

D. Byun and K. L. Schere. Review of the governing equations, computational algorithms, and other components of the models-3 community multiscale air quality (CMAQ) modeling system. *Applied Mechanics Reviews*, 59(2):51, 2006. doi:10.1115/1.2128636. [Cited on page 7]

J. M. Cascón, E. Rodríguez, J. M. Escobar, and R. Montenegro. Comparison of the meccano method with standard mesh generation techniques. *Engineering with Computers*, 31(1):161–174, 2013. doi:10.1007/s00366-013-0338-6. [Cited on page 15]

J. Chen, D. Zhao, Z. Huang, Y. Zheng, and S. Gao. Three-dimensional constrained boundary recovery with an enhanced Steiner point suppression procedure. *Computers & Structures*, 89(5-6):455–466, 2011. doi:10.1016/j.compstruc.2010.11.016. [Cited on page 18]

D. P. Chock, M. J. Whalen, S. L. Winkler, and P. Sun. Implementing the trajectory-grid transport algorithm in an air quality model. *Atmospheric Environment*, 39(22):4015–4023, 2005. doi:10.1016/j.atmosenv.2005.03.037. [Cited on page 2]

P. Colella and P. R. Woodward. The Piecewise Parabolic Method (PPM) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984. doi:10.1016/0021-9991(84)90143-8. [Cited on page 7]

Community Modeling and Analysis System and the Institute for the Environment University of North Carolina at Chapel Hill Chapel Hill, NC. *Operational Guidance for the Community Multiscale Air Quality (CMAQ) Modeling System, Version 5.0 (February 2012 Release)*, 2012. URL http://www.airqualitymodeling.org/cmaqwiki/index.php?title=CMAQ_version_5.0_%28February_2010_release%29_OGD. [Cited on pages 7 and 61]

L. Davis. *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York, NY, USA, 1991. ISBN 0442001738. [Cited on page 30]

J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. John Wiley and Sons Ltd, West Sussex., 2003. doi:10.1002/0470013826. [Cited on pages 2 and 43]

ENVIRON International Corporation. *CAMx Comprehensive Air Quality Model with extensions. User's Guide v. 6.2*. ENVIRON International Corporation. Novato, California., 2015. URL http://www.camx.com/files/camxusersguide_v6-20.pdf. [Cited on page 2]

J. M. Escobar and R. Montenegro. Several aspects of three-dimensional Delaunay triangulation. *Advances in Engineering Software*, 27(1-2):27–39, 1996. doi:10.1016/0965-9978(96)00006-3. [Cited on page 18]

J. M. Escobar, E. Rodríguez, R. Montenegro, G. Montero, and J. M. González-Yuste. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2775–2787, 2003. doi:10.1016/S0045-7825(03)00299-8. [Cited on pages 15, 19, and 20]

T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf. The shuttle radar topography mission. *Reviews of Geophysics*, 45(2), 2007. doi:10.1029/2005rg000183. RG2004. [Cited on page 61]

L. Ferragut, R. Montenegro, and A. Plaza. Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems. *Communications in Numerical Methods in Engineering*, 10 (5):403–412, 1994. doi:10.1002/cnm.1640100506. [Cited on page 16]

L. Ferragut, R. Montenegro, G. Montero, E. Rodríguez, M. Asensio, and J. M. Escobar. Comparison between 2.5-D and 3-D realistic models for wind field adjustment. *Journal of Wind Engineering and Industrial Aerodynamics*, 98(10-11):548–558, 2010. doi:10.1016/j.jweia.2010.04.004. [Cited on pages 2 and 28]

B. J. Finlayson-Pitts and J. N. Pitts. Tropospheric air pollution: Ozone, airborne toxics, polycyclic aromatic hydrocarbons, and particles. *Science*, 276(5315):1045–1051, 1997. doi:10.1126/science.276.5315.1045. [Cited on page 42]

M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003. doi:10.1016/S0167-8396(03)00002-5. [Cited on page 15]

L. A. Freitag and P. M. Knupp. Tetrahedral mesh improvement via optimization of the element condition number. *International Journal for Numerical Methods in Engineering*, 53(6):1377–1391, 2002. doi:10.1002/nme.341. [Cited on page 19]

L. A. Freitag and P. Plassmann. Local optimization-based simplicial mesh untangling and improvement. *International Journal for Numerical Methods in Engineering*, 49(1-2):109–125, 2000. doi:10.1002/1097-0207(20000910/20)49:1/2<109::AID-NME925>3.0.CO;2-U. [Cited on page 19]

R. V. Garimella and B. K. Swartz. Curvature estimation for unstructured triangulations of surfaces. Technical Report LA-UR-03-8240, Los Alamos National Laboratory, 2003. URL `http://math.lanl.gov/Research/Publications/Docs/garimella-2003-curvature.pdf`. [Cited on page 22]

C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1971. ISBN 0136266061. [Cited on page 45]

P. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering*, 92(3):269–288, 1991. doi:10.1016/0045-7825(91)90017-Z. [Cited on page 18]

J. M. González-Yuste, R. Montenegro, J. M. Escobar, G. Montero, and E. Rodríguez. Local refinement of 3-D triangulations using object-oriented methods. *Advances in Engineering Software*, 35(10-11):693 – 702, 2004. doi:10.1016/j.advengsoft.2003.07.003. [Cited on page 39]

T. L. Greaver, T. J. Sullivan, J. D. Herrick, M. C. Barber, J. S. Baron, B. J. Cosby, M. E. Deerhake, R. L. Dennis, J.-J. B. Dubois, C. L. Goodale, A. T. Herlihy, G. B. Lawrence, L. Liu, J. A. Lynch, and K. J. Novak. Ecological effects of nitrogen and sulfur air pollution in the US: what do we know? *Frontiers in Ecology and the Environment*, 10(7):365–372, 2012. doi:10.1890/110049. [Cited on page 1]

A. E. S. Green, R. P. Singhal, and R. Venkateswar. Analytic extensions of the gaussian plume model. *Journal of the Air Pollution Control Association*, 30(7):773–776, 1980. doi:10.1080/00022470.1980.10465108. [Cited on page 39]

G. Grell, J. Dudhia, and D. Stauffer. A description of the fifth-generation Penn State/NCAR Mesoscale Model (MM5). Technical Report NCAR/TN-398+STR, National Center for Atmospheric Research, 1994. doi:10.5065/D60Z716B. [Cited on page 7]

K. Hanjalić and S. Kenjereš. Dynamic simulation of pollutant dispersion over complex urban terrains: A tool for sustainable development, control and management. *Energy*, 30(8):1481 – 1497, 2005. doi:10.1016/j.energy.2004.05.001. [Cited on page 2]

K. He, Y. Lei, X. Pan, Y. Zhang, Q. Zhang, and D. Chen. Co-benefits from energy policies in china. *Energy*, 35(11):4265 – 4272, 2010. doi:10.1016/j.energy.2008.07.021. [Cited on page 2]

J. Hernandez, L. Cremades, and J. Baldasano. Dispersion modelling of a tall stack plume in the spanish mediterranean coast by a particle model. *Atmospheric Environment*, 29(11):1331–1341, 1995. doi:10.1016/1352-2310(94)00346-M. [Cited on page 1]

O. Hertel, R. Berkowicz, J. Christensen, and Ø. Hov. Test of two numerical schemes for use in atmospheric transport-chemistry models. *Atmospheric Environment. Part A. General Topics*, 27(16): 2591–2611, 1993. doi:10.1016/0960-1686(93)90032-t. [Cited on page 45]

G. M. Hidy. Multiscale impact of fuel consumption on air quality. *Energy & Fuels*, 16(2):270–281, 2002. doi:10.1021/ef0101659. [Cited on page 2]

J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-58111-6. [Cited on page 30]

C. Hua. An inverse transformation for quadrilateral isoparametric elements: Analysis and application. *Finite Elements in Analysis and Design*, 7(2):159–166, 1990. doi:10.1016/0168-874x(90)90007-2. [Cited on page 13]

P. J. Hurley, W. L. Physick, and A. K. Luhar. TAPM: a practical approach to prognostic meteorological and air pollution modelling. *Environmental Modelling & Software*, 20(6):737–752, 2005. doi:10.1016/j.envsoft.2004.04.006. [Cited on page 1]

M. Z. Jacobson and R. P. Turco. SMVGEAR: A sparse-matrix, vectorized gear code for atmospheric models. *Atmospheric Environment*, 28(2):273–284, 1994. doi:10.1016/1352-2310(94)90102-3. [Cited on page 45]

P. Jimenez, J. M. Baldasano, and D. Dabdub. Comparison of photochemical mechanisms for air quality modeling. *Atmospheric Environment*, 37(30):4179–4194, 2003. doi:10.1016/S1352-2310(03)00567-3. [Cited on page 42]

V. John. A numerical study of a posteriori error estimators for convection–diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 190(5-7):757–781, 2000. doi:10.1016/s0045-7825(99)00440-5. [Cited on page 45]

F. Kirchner. The chemical mechanism generation programme CHEMATA – Part 1: The programme and first applications. *Atmospheric Environment*, 39(6):1143–1159, 2005. doi:10.1016/j.atmosenv.2004.09.086. [Cited on page 42]

D. Kley. Tropospheric chemistry and transport. *Science*, 276(5315):1043–1044, 1997. doi:10.1126/science.276.5315.1043. [Cited on page 42]

P. M. Knupp. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.*, 23(1):193–218, 2001. doi:10.1137/S1064827500371499. [Cited on page 19]

I. Kossaczký. A recursive approach to local mesh refinement in two and three dimensions. *Journal of Computational and Applied Mathematics*, 55(3):275–288, 1994. doi:10.1016/0377-0427(94)90034-5. [Cited on page 15]

I. Lagzi, D. Kármán, T. Turányi, A. S. Tomlin, and L. Haszpra. Simulation of the dispersion of nuclear contamination using an adaptive eulerian grid model. *Journal of Environmental Radioactivity*, 75(1):59–82, 2004. doi:10.1016/j.jenvrad.2003.11.003. [Cited on page 2]

D. Lalas and C. Ratto. *Modelling of Atmospheric Flow Fields.* World Scientific Publishing, Singapore, 1996. [Cited on pages 25 and 26]

R. G. Lamb and D. R. Durran. Eddy diffusivities derived from a numerical model of the convective planetary boundary layer. *Il Nuovo Cimento C*, 1(1):1–17, 1978. doi:10.1007/bf02508043. [Cited on page 42]

C. K. Lee. On curvature element-size control in metric surface mesh generation. *International Journal for Numerical Methods in Engineering*, 50(4):787–807, 2001. doi:10.1002/1097-0207(20010210)50:4<787::AID-NME51>3.0.CO;2-O. [Cited on page 21]

S.-M. Lee, H. J. Fernando, M. Princevac, D. Zajic, M. Sinesi, J. L. McCulley, and J. Anderson. Transport and diffusion of ozone in the nocturnal and morning planetary boundary layer of the phoenix valley. *Environmental Fluid Mechanics*, 3:331–362, 2003. doi:10.1023/A:1023680216173. [Cited on page 2]

D. Levine. A parallel genetic algorithm for the set partitioning problem. Technical Report ANL-94/23, Argonne National Laboratory, 1994. URL `http://140.221.6.23/pub/tech_reports/reports/ANL9423.pdf`. [Cited on page 29]

R. Li. On multi-mesh h-adaptive methods. *Journal of Scientific Computing*, 24(3):321–341, 2005. doi:10.1007/s10915-004-4793-5. [Cited on page 41]

C.-J. Lin and J. J. Moré. Incomplete Cholesky factorizations with limited memory. *SIAM Journal on Scientific Computing*, 21(1):24–45, 1999. doi:10.1137/S1064827597327334. [Cited on page 43]

R. Löhner and J. D. Baum. Adaptive h-refinement on 3d unstructured grids for transient problems. *International Journal for Numerical Methods in Fluids*, 14(12):1407–1419, 1992. doi:10.1002/fld.1650141204. [Cited on page 39]

M. J. Martín, D. E. Singh, J. C. Mouriño, F. F. Rivera, R. Doallo, and J. D. Bruguera. High performance air pollution modeling for a power plant environment. *Parallel Computing*, 29(11-12):1763–1790, 2003. doi:10.1016/j.parco.2003.05.018. [Cited on page 2]

D. L. Mauzerall, B. Sultan, N. Kim, and D. F. Bradford. NOx emissions from large point sources: variability in ozone production, resulting health damages and economic costs. *Atmospheric Environment*, 39(16):2851–2866, 2005. doi:10.1016/j.atmosenv.2004.12.041. [Cited on page 2]

G. J. McRae, W. R. Goodin, and J. H. Seinfeld. Development of a second-generation mathematical model for urban air pollution–I. Model formulation. *Atmospheric Environment (1967)*, 16(4):679–696, 1982. doi:10.1016/0004-6981(82)90386-9. [Cited on page 27]

L. Monforte and A. Pérez-Foguet. Esquema adaptativo para problemas tridimensionales de convección-difusión. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 30(1):60–67, 2014a. doi:10.1016/j.rimni.2012.11.003. [Cited on pages 2, 3, 41, and 45]

L. Monforte and A. Pérez-Foguet. A multimesh adaptive scheme for air quality modeling with the finite element method. *Int. J. Numer. Meth. Fluids*, 74(6):387–405, 2014b. doi:10.1002/fld.3855. [Cited on pages 2, 3, 41, 46, and 67]

R. Montenegro, G. Montero, J. M. Escobar, E. Rodríguez, and J. M. González-Yuste. 3-D Adaptive Wind Field Simulation Including Effects of Chimney Emissions. In *Proceedings of WCCM VI/APCOM'04, Beijing, China*. Tsinghua University Press and Springer-Verlag, 2004. [Cited on page 2]

R. Montenegro, J. M. Cascón, J. M. Escobar, E. Rodríguez, and G. Montero. An automatic strategy for adaptive tetrahedral mesh generation. *Applied Numerical Mathematics*, 59(9):2203–2217, 2009. doi:10.1016/j.apnum.2008.12.010. [Cited on page 15]

R. Montenegro, J. M. Cascón, E. Rodríguez, J. M. Escobar, and G. Montero. The meccano method for automatic three-dimensional triangulation and volume parametrization of complex solids. In B. Topping, J. Adam, F. Pallarés, R. Bru, and M. Romero, editors, *Developments and Applications in Engineering Computational Technology*, chapter 2, pages 19–48. Saxe-Coburg Publications, Stirlingshire, seventh edition, 2010. ISBN 978-1-874672-48-7. doi:10.4203/csets.26.2. [Cited on page 15]

R. Montenegro, A. Oliver, E. Rodríguez, J. M. Escobar, G. Montero, and A. Pérez-Foguet. Local scale finite element modelling of stack pollutant emissions. In *Conference on Computational Methods for Coupled Problems in Science and Engineering*, Ibiza, 2013. [Cited on page 61]

G. Montero, R. Montenegro, and J. M. Escobar. A 3-D diagnostic model for wind field adjustment. *Journal of Wind Engineering and Industrial Aerodynamics*, 74-76(0):249–261, 1998. doi:10.1016/S0167-6105(98)00022-1. [Cited on pages 2, 26, and 28]

G. Montero, R. Montenegro, J. M. Escobar, E. Rodríguez, and J. M. González-Yuste. Velocity field modelling for pollutant plume using 3-D adaptive finite element method. In M. Bubak, G. van Albada, P. Sloot, and J. Dongarra, editors, *Computational Science - ICCS 2004*, volume 3037 of *Lecture Notes in Computer Science*, pages 642–645. Springer Berlin / Heidelberg, 2004. ISBN 978-3-540-22115-9. doi:10.1007/978-3-540-24687-9_92. [Cited on page 2]

G. Montero, E. Rodríguez, R. Montenegro, J. M. Escobar, and J. M. González-Yuste. Genetic algorithms for an improved parameter estimation with local refinement of tetrahedral meshes in a wind model. *Advances in Engineering Software*, 36(1):3–10, 2005. doi:10.1016/j.advengsoft.2004.03.011. [Cited on pages 2 and 28]

D. J. Moore. A comparison of the trajectories of rising buoyant plumes with theoretical/empirical models. *Atmospheric Environment (1967)*, 8(5):441 – 457, 1974. doi:10.1016/0004-6981(74)90060-2. [Cited on page 31]

R. Morris, G. Yarwood, C. Emery, and G. Wilson. Recent advances in photochemical air quality modeling using the CAMx Model: Current update and ozone modeling of point source impacts. In *Proceedings of The Air & Waste Management Association's 95th Annual Conference & Exhibition*. A&WMA, 2002. paper #43180. [Cited on pages 1 and 2]

M. Murphy, D. M. Mount, and C. W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 67–74, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics. ISBN 0-89871-453-2. URL `http://dl.acm.org/citation.cfm?id=338219.338236`. [Cited on page 18]

L. E. Olcese and B. M. Toselli. Development of a model for reactive emissions from industrial stacks. *Environmental Modelling & Software*, 20(10):1239–1250, 2005. doi:10.1016/j.envsoft.2004.08.008. [Cited on pages 1 and 33]

A. Oliver and A. Pérez-Foguet. Local scale air quality modelling based on cmaq forecast data. In *12^{th} EMS Annual Meeting & 9^{th} European Conference on Applied Climatology*, Łódź, 2012. [Cited on page 61]

A. Oliver, G. Montero, R. Montenegro, E. Rodríguez, J. M. Escobar, and A. Pérez-Foguet. Finite element simulation of a local scale air quality model over complex terrain. *Advances in Science and Research*, 8:105–113, 2012. doi:10.5194/asr-8-105-2012. [Cited on pages 2 and 3]

A. Oliver, G. Montero, R. Montenegro, E. Rodríguez, J. M. Escobar, and A. Pérez-Foguet. Adaptive finite element simulation of stack pollutant emissions over complex terrains. *Energy*, 49(0):47–60, 2013. doi:10.1016/j.energy.2012.10.051. [Cited on pages 2, 3, 31, and 47]

A. Oliver, A. Pérez-Foguet, E. Rodríguez, G. Montero, and R. Montenegro. Local scale air quality model with several pollutant sources. In *Fourth African Conference on Computational Mechanics (AfriCOMP15)*, Marrakech, 2015a. [Cited on pages 2, 31, and 61]

A. Oliver, E. Rodríguez, J. M. Escobar, G. Montero, M. Hortal, J. Calvo, J. M. Cascón, and R. Montenegro. Wind forecasting based on the HARMONIE model and adaptive finite elements. *Pure and Applied Geophysics*, 172(1):109–120, 2015b. doi:10.1007/s00024-014-0913-9. [Cited on pages 2, 26, and 29]

T. L. Otte and J. E. Pleim. The meteorology-chemistry interface processor (MCIP) for the CMAQ modeling system: updates through MCIPv3.4.1. *Geoscientific Model Development*, 3(1):243–256, 2010. doi:10.5194/gmd-3-243-2010. [Cited on page 8]

I. Palomino and F. Martín. A simple method for spatial interpolation of the wind in complex terrain. *J. Appl. Meteor.*, 34(7):1678–1693, 1995. doi:10.1175/1520-0450-34.7.1678. [Cited on page 26]

M. Pascal, M. Corso, O. Chanel, C. Declercq, C. Badaloni, G. Cesaroni, S. Henschel, K. Meister, D. Haluza, P. Martin-Olmedo, and S. Medina. Assessing the public health impacts of urban air pollution in 25 european cities: Results of the aphekom project. *Science of The Total Environment*, 449:390–400, 2013. doi:10.1016/j.scitotenv.2013.01.077. [Cited on page 1]

F. Pasquill. The estimation of the dispersion of windborne material. *Meteorol. mag*, 90(1063):33–49, 1961. [Cited on page 26]

J. A. Patz, D. Campbell-Lendrum, T. Holloway, and J. A. Foley. Impact of regional climate change on human health. *Nature*, 438(7066):310–317, 2005. doi:10.1038/nature04188. [Cited on page 1]

M. T. Pay, F. Martínez, M. Guevara, and J. M. Baldasano. Air quality forecasts on a kilometer-scale grid over complex Spanish terrains. *Geoscientific Model Development*, 7(5):1979–1999, 2014. doi:10.5194/gmd-7-1979-2014. [Cited on page 1]

A. Pérez-Foguet. Characterization of local wind patterns in complex mountain valleys. *International Journal of Climatology*, 34(6):1741–1759, 2014. doi:10.1002/joc.3798. [Cited on page 74]

A. Pérez-Foguet and A. Oliver. Coupling regional and local air quality models for short-time prediction around punctual pollutant sources using finite elements. In *8^th^ World Congress on Computational Mechanics and 5^th^ European Congress on Computational Methods in Applied Sciences and Engineering*, Venezia, 2008. [Cited on page 61]

A. Pérez-Foguet, A. Oliver, J. M. Escobar, and E. Rodríguez. Finite element simulation of chimney emissions: A proposal for near field impact assessment in highly complex terrains. In B. Topping, G. Montero, and R. Montenegro, editors, *Procedings of the Fifth International Conference on Engineering Computational Technology*, 2006. doi:10.4203/ccp.84.101. paper #101. [Cited on pages 2 and 3]

P.-O. Persson. Mesh size functions for implicit geometries and PDE-based gradient limiting. *Engineering with Computers*, 22(2):95–109, 2006. doi:10.1007/s00366-006-0014-1. [Cited on page 21]

A. Plaza, R. Montenegro, and L. Ferragut. An improved derefinement algorithm of nested meshes. *Advances in Engineering Software*, 27(1-2):51–57, 1996. doi:10.1016/0965-9978(96)00005-1. [Cited on page 16]

J. E. Pleim and J. S. Chang. A non-local closure model for vertical mixing in the convective boundary layer. *Atmospheric Environment. Part A. General Topics*, 26(6):965–981, 1992. doi:10.1016/0960-1686(92)90028-j. [Cited on page 42]

V. Ramanathan and Y. Feng. Air pollution, greenhouse gases and climate change: Global and regional perspectives. *Atmospheric Environment*, 43(1):37–50, 2009. doi:10.1016/j.atmosenv.2008.09.063. [Cited on page 1]

A. R. Ravishankara. Heterogeneous and multiphase chemistry in the troposphere. *Science*, 276 (5315):1058–1065, 1997. doi:10.1126/science.276.5315.1058. [Cited on page 42]

M.-C. Rivara. A grid generator based on 4-triangles conforming mesh-refinement algorithms. *International Journal for Numerical Methods in Engineering*, 24(7):1343–1354, 1987. doi:10.1002/nme.1620240710. [Cited on page 16]

X. Roca. *Paving the path towards automatic hexahedral mesh generation*. PhD thesis, Universitat Politècnica de Catalunya – BarcelonaTech, 2009. hdl:10803/5858. [Cited on page 20]

X. Roca, J. Sarrate, and E. Ruiz-Gironés. A graphical modeling and mesh generation environment for simulations based on boundary representation data. In *Communications in Num. Meth. Engrg.*, Porto, 2007. [Cited on page 20]

X. Roca, E. Ruiz-Gironés, and J. Sarrate. EZ4U: Mesh generation environment. `http://www.lacan.upc.edu/ez4u.htm`, 2010. [Cited on page 20]

E. Rodríguez. *Modelización numérica de campos de viento mediante elementos finitos en 3-D*. PhD thesis, Universidad de Las Palmas de Gran Canaria, 2004. `http://www.dca.iusiani.ulpgc.es/proyecto2012-2014/pdfnuevo/tesis_erb.pdf`. [Cited on page 25]

E. Rodríguez, G. Montero, J. M. Escobar, R. Montenegro, and A. Oliver. Wind3D. `http://www.dca.iusiani.ulpgc.es/Wind3D`, 2012. [Cited on page 25]

A. Rodríguez-Ferran and M. L. Sandoval. Numerical performance of incomplete factorizations for 3D transient convection–diffusion problems. *Advances in Engineering Software*, 38(6):439–450, 2007. doi:10.1016/j.advengsoft.2006.09.003. [Cited on pages 2 and 43]

D. L. Ropp, J. N. Shadid, and C. C. Ober. Studies of the accuracy of time integration methods for reaction–diffusion equations. *Journal of Computational Physics*, 194(2):544–574, 2004. doi:10.1016/j.jcp.2003.08.033. [Cited on pages 3 and 44]

E. Ruiz-Gironés, X. Roca, and J. Sarrate. Size-preserving size functions and smoothing procedures for adaptive quadrilateral mesh generation. *Engineering with Computers*, pages 1–16, 2014. doi:10.1007/s00366-014-0371-0. [Cited on page 21]

T. B. Ryerson, M. Trainer, J. S. Holloway, D. D. Parrish, L. G. Huey, D. T. Sueper, G. J. Frost, S. G. Donnelly, S. Schauffler, E. L. Atlas, W. C. Kuster, P. D. Goldan, G. HÃijbler, J. F. Meagher, and F. C. Fehsenfeld. Observations of ozone formation in power plant plumes and implications for ozone control strategies. *Science*, 292(5517):719–723, 2001. doi:10.1126/science.1058113. [Cited on page 2]

R. San José, J. L. Pérez, and R. M. González. An operational real-time air quality modelling system for industrial plants. *Environmental Modelling & Software*, 22(3):297–307, 2007. doi:10.1016/j.envsoft.2005.07.030. [Cited on page 1]

A. Sandu, J. Verwer, J. Blom, E. Spee, G. Carmichael, and F. Potra. Benchmarking stiff ode solvers for atmospheric chemistry problems II: Rosenbrock solvers. *Atmospheric Environment*, 31(20): 3459–3472, 1997a. doi:10.1016/s1352-2310(97)83212-8. [Cited on page 45]

A. Sandu, J. Verwer, M. V. Loon, G. Carmichael, F. Potra, D. Dabdub, and J. Seinfeld. Benchmarking stiff ODE solvers for atmospheric chemistry problems-I. Implicit vs explicit. *Atmospheric Environment*, 31(19):3151–3166, 1997b. doi:10.1016/S1352-2310(97)00059-9. [Cited on page 2]

J. Sarrate and A. Huerta. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, 49(10):1327–1350, 2000. doi:10.1002/1097-0207(20001210)49:10<1327::AID-NME996>3.0.CO;2-L. [Cited on page 22]

J. Sarrate and A. Huerta. An improved algorithm to smooth graded quadrilateral meshes preserving the prescribed element size. *Communications in Numerical Methods in Engineering*, 17(2):89–99, 2001. doi:10.1002/1099-0887(200102)17:2<89::AID-CNM357>3.0.CO;2-E. [Cited on page 22]

G. Sarwar, D. Luecken, and G. Yarwood. Chapter 2.9 developing and implementing an updated chlorine chemistry into the community multiscale air quality model. In *Developments in Environmental Science*, pages 168–176. Elsevier BV, 2007. doi:10.1016/s1474-8177(07)06029-9. [Cited on page 44]

R. D. Saylor and G. D. Ford. On the comparison of numerical methods for the integration of kinetic equations in atmospheric chemistry and transport models. *Atmospheric Environment*, 29(19): 2585–2593, 1995. doi:10.1016/1352-2310(95)00187-4. [Cited on page 2]

J. S. Scire, D. G. Strimaitis, and R. J. Yamartino. *A User's Guide for the CALPUFF Dispersion Model (Version 5)*. Earth Tech., Inc, Concord, MA., 2000. URL `http://www.src.com/calpuff/download/CALPUFF_UsersGuide.pdf`. [Cited on pages 2 and 42]

C. Seigneur. Air pollution: Current challenges and future opportunities. *AIChE Journal*, 51(2):356–364, 2005. doi:10.1002/aic.10458. [Cited on page 1]

C. Shir. A preliminary numerical study of atmospheric turbulent flows in the idealized planetary boundary layer. *Journal of the Atmospheric Sciences*, 30(7):1327–1339, 1973. doi:10.1175/1520-0469(1973)030<1327:apnsoa>2.0.co;2. [Cited on pages 42 and 67]

H. Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2): 11:1–11:36, 2015. doi:10.1145/2629697. [Cited on pages 20, 22, and 46]

W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. Duda, X.-Y. Huang, W. Wang, and J. G. Powers. A Description of the Advanced Research WRF Version 3. Technical Report NCAR/TN-475+STR, National Center for Atmospheric Research, 2008. doi:10.5065/D68S4MVH. [Cited on page 7]

P. Solin, J. Cerveny, L. Dubcova, and D. Andrs. Monolithic discretization of linear thermoelasticity problems via adaptive multimesh hp-FEM. *Journal of Computational and Applied Mathematics*, 234(7):2350–2357, 2010a. doi:10.1016/j.cam.2009.08.092. [Cited on page 41]

P. Solin, L. Dubcova, and J. Kruis. Adaptive hp-FEM with dynamical meshes for transient heat and moisture transfer problems. *Journal of Computational and Applied Mathematics*, 233(12):3103–3112, 2010b. doi:10.1016/j.cam.2009.07.025. [Cited on page 41]

M. Souto, J. Souto, V. Pérez-Muñuzuri, J. Casares, and J. Bermúdez. A comparison of operational lagrangian particle and adaptive puff models for plume dispersion forecasting. *Atmospheric Environment*, 35(13):2349–2360, 2001. doi:10.1016/S1352-2310(00)00537-9. [Cited on page 1]

W. Spears and K. D. D. Jong. On the virtues of parameterized uniform crossover. Technical Report ADA293985, Naval Research Lab Washington DC, 1995. URL `http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA293985`. [Cited on page 30]

S. Szopa, B. Aumont, and S. Madronich. Assessment of the reduction methods used to develop chemical schemes: Building of a new chemical scheme for VOC oxidation suited to three-dimensional multiscale HO_x-NO_x-VOC chemistry simulations. *Atmospheric Chemistry and Physics*, 5(9):2519–2538, 2005. doi:10.5194/acp-5-2519-2005. [Cited on page 42]

M. Taghavi, S. Cautenet, and J. Arteta. Impact of a highly detailed emission inventory on modeling accuracy. *Atmospheric Research*, 74(1-4):65–88, 2005. doi:10.1016/j.atmosres.2004.06.007. [Cited on pages 1 and 2]

A. Tomlin, S. Ghorai, G. Hart, and M. Berzins. 3-D multi-scale air pollution modelling using adaptive unstructured meshes. *Environmental Modelling & Software*, 15(6-7):681–692, 2000. doi:10.1016/S1364-8152(00)00038-4. [Cited on page 2]

J. G. Verwer, E. J. Spee, J. G. Blom, and W. Hundsdorfer. A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM Journal on Scientific Computing*, 20(4):1456–1480, 1999. doi:10.1137/S1064827597326651. [Cited on pages 3 and 44]

C. J. Walcek. Effects of wind shear on pollution dispersion. *Atmospheric Environment*, 36(3):511–517, 2002. doi:10.1016/S1352-2310(01)00383-1. [Cited on page 1]

H. Walton, D. Dajnak, S. Beevers, M. Williams, P. Watkiss, and A. Hunt. Understanding the Health Impacts of Air Pollution in London. Technical Report TFL 90419, Environmental Research Group, School of Biomedical Sciences, King's College London, 2015. URL `http://www.kcl.ac.uk/lsm/research/divisions/aes/research/ERG/research-projects/HIAinLondonKingsReport14072015final.pdf`. [Cited on page 1]

D. Whitley. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 116–121. Morgan Kaufmann, 1989. [Cited on page 30]

G. Z. Whitten, G. Heo, Y. Kimura, E. McDonald-Buller, D. T. Allen, W. P. Carter, and G. Yarwood. A new condensed toluene mechanism for carbon bond: CB05-TU. *Atmospheric Environment*, 44 (40):5346–5355, 2010. doi:10.1016/j.atmosenv.2009.12.029. [Cited on page 44]

G. Winter, G. Montero, L. Ferragut, and R. Montenegro. Adaptive strategies using standard and mixed finite elements for wind field adjustment. *Solar Energy*, 54(1):49–56, 1995. doi:10.1016/0038-092X(94)00100-R. [Cited on page 2]

R. J. Yamartino. Nonnegative, conserved scalar transport using grid-cell-centered, spectrally constrained blackman cubics for applications on a variable-thickness mesh. *Monthly Weather Review*, 121:753–763, 1993. doi:10.1175/1520-0493(1993)121<0753:NCSTUG>2.0.CO;2. [Cited on page 7]

G. Yarwood, S. Rao, M. Yocke, and G. Whitten. Updates to the Carbon Bond chemical mechanism: CB05. Technical Report RT-0400675, Final report to the US EPA, 2005. URL `http://www.camx.com/publ/pdfs/cb05_final_report_120805.pdf`. [Cited on pages 2, 8, 42, and 44]

P. Zannetti. *Air Pollution Modeling*. Computational Mechanics Publications, Boston, 1990. [Cited on page 27]

Z. Zlatev. *Computer Treatment of Large Air Pollution Models*. Kluver Academic Publishers, Dordrecht., 1995. [Cited on page 42]