Modeling and distributed computing of snow transport and delivery on meso-scale in a complex orography

Modelització i computació distribuïda de fenòmens de transport i dipòsit de neu a meso-escala en una orografia complexa

Thesis dissertation submitted in fulfillment of the requirements for the degree of Doctor of Philosophy
Programa de Doctorat en Societat de la Informació i el Coneixement

Alan Ward Koeck
Advisor: Dr. Josep Jorba Esteve

Distributed, Parallel and Collaborative Systems Research Group (DPCS)
Universitat Oberta de Catalunya (UOC)
Rambla del Poblenou 156, 08018 Barcelona

– Barcelona, 2015 –

In memoriam

Alan Ward, MA Oxon, PhD Dublin
1937-2014

Acknowledgements

A long-term commitment such as this thesis could not prosper on my own merits alone. I am deeply indebted to many people from several institutions.

At the Universitat Oberta de Catalunya (UOC), in the first place I would like to thank my advisor, Dr. Josep Jorba, for his constant help and guidance over these years. As they say, any merits of this work are probably his - and any errors most surely my own. Dr. Joan Manuel Marquès, co-leader of the Distributed, Parallel and Collaborative Systems (DPCS) research group at UOC has also been very enthusiastic in his reception of the initial idea, and supportive of the project as it evolved.

In Andorra, friendly and constructive discussions have taken place with investigators from the Observatori de Sostenibilitat d'Andorra (OBSA) and Institut d'Estudis Andorrans (IEA). Many colleagues and fellow-teachers at Universitat d'Andorra (UdA) and Escola Andorrana have been helpful in maintaining steam as the years passed by, as has my former boss at the Department of Higher Education, Dr. Joan-Marc Miralles.

I am also indebted to two anonymous reviewers who, by their comments, pointed out several weaknesses that have since been removed and in general helped give the thesis a better cohesion.

On a more personal note, my late father Dr. Alan Ward (Alan Mac an Bháird) has always been a role model for me, both as an academic of high standards and as a warm and caring human being. Unfortunately, he left us in February 2014 as the process of thesis redaction was getting underway.

Last, but not least, my partner Mercè Rey has been very supportive and patient both with my quirks and with the amount of time spent in research and investigation.

Abstract

Human activities in mountain terrain are increasing in scope, as are their impact on the natural environment, such as the effects of artificial snow generation. This study describes the working principles, development and validation of a Computational Fluid Dynamics (CFD) computer model of snowfall over a complex orography, with the aim of optimizing ski slope or other installations according to local weather patterns, thus helping the decision-making process.

In the first step, the spatial domain is discretized, with the main focus on challenging topography that tends to produce deformed mesh volumes. A novel measure of mesh deformation is then defined and applied to discuss different strategies of mesh optimization with the goal of facilitating parallel computer solutions of the Navier-Stokes fluid transport equations. These strategies are evaluated with regards to their implementation as a parallel computer algorithm.

In the second step, a computer model is designed to solve the Navier-Stokes incompressible turbulent fluid equations. Slip- and no-slip boundary layers are considered, modeling surface roughness with the k_s method. The efficiency of the CFD computational toolkit are discussed, as applied within the limits of a small or medium-sized commodity computation cluster using commercially available equipment.

Finally, the degree of coupling required between the snow- and air-phases of the fluid during the computer modeling of snowfall is discussed. A two-fluid (Euler-Lagrangian) methodology is implemented. The effects of tangent surface wind speed on primary and secondary snow transport are integrated into the model. An assessment is made of the application of parallel computing to the solution of Lagrangian movement of individual snow parcels. Experimental data is used to verify the suitability of computational techniques.

Additionally, real-world applications of such snowfall models are discussed in relation to ski-slope planning and high-altitude road snow clearing. An application of the model to wind energy production planning is presented.

Contents

List of Figures

Symbols and Abbreviations

notation	meaning	units
$\delta_{i,j}$	the Kronecker delta function	-
∇x	the gradient of quantity x	-
$\triangle x$	the divergence of quantity x	-
I	identity tensor	-
$\vec{n}$	normal vector	-
D	tensor of deformation	s^{-1}
e	specific internal energy	$J.kg^{-1}$
$\vec{f}$	heat flux vector	$J.m^{-2}.s^{-1}$
$\vec{f_S}$	contact forces exerted on surface	$N.m^{-2}$
$\vec{f_V}$	forces exerted on volume	$N.m^{-3}$
m	mass	kg
m_c	mass of carrier fluid in elementary volume	kg
m_p	mass of discrete particles in elementary volume	kg
$\vec{M}$	moment	$kg.m$
μ	coefficient of viscosity	$Pa.s$
p	pressure	Pa
$\tilde{q}$	internal heat source	$J.m^{-3}.s^{-1}$
RH	relative humidity	%
ρ	density	$kg.m^{-3}$
ρ_c	relative density of carrier fluid	$kg.m^{-3}$
ρ_p	relative density of discrete particles	$kg.m^{-3}$
s	specific entropy	$J.K^{-1}$
S	surface of control volume	m^2
σ	stress tensor	Pa
θ	temperature	K
τ	viscous stress tensor	Pa
$\vec{u}$	fluid speed	$m.s^{-1}$
V	volume of control volume	m^3
Φ_p	volume fraction of particles relative to carrier fluid	-

Chapter 1

Introduction and goals

1.1 Background

Human use of available land spaces has increased over the last century. Technological advances allow us to make use of difficult terrain types such as desert and mountain regions with greater ease than before. At the same time, the population of more developed countries have available to them longer periods of time that may be dedicated to leisure activities in such spaces.

However, at least two separate issues have been raised over the anthropic uses of mountain terrain. The first is related to the impact human activities may have on the natural environment. The effects of ski slope installation in area with rich and specific ecological systems has raised concerns for their effects on local fauna and flora (Jennifer W. Burt and Kevin J. Rice, 2009), including the reduction of habitat areas and alteration of wildlife behavior patterns. Water usage for usage in artificial snow generation systems has also been criticized as altering the availability of the resource for populations and ecosystems situated lower down the valleys (Carmen de Jong, 2007), specifically in times of the year when the availability of water is already reduced such as the months of January and February (in the Northern Hemisphere). In a similar fashion, the additives used in artificial snow production have also been noted to affect the environment (Christian Rixen, Veronika Stoeckli and Walter Ammann, 2003).

The second issue raised is that of climate change. Though this topic has been the object of much controversy concerning its global effects, there seems to be a consensus that its influence is specially remarkable in specific regions that include sites at high elevation (J. L. Innes, 1991; Martin Beniston, H. F. Diaz and R. S. Bradley, 1997). These effects combine the rising isotherms on the one hand, with changes in the composition of the atmosphere leading to alterations of ultraviolet expose patterns on the other to produce modifications of the limits in altitude between different natural environmental strata (Figure 1.1). Changes in vegetation such as the higher extension of forests (Cécile H. Albert *et al.*, 2008) that tend to replace alpine prairies also imply changes in ground albedo (Jan Esper *et al.*, 2012), thus altering even more local temperature response.

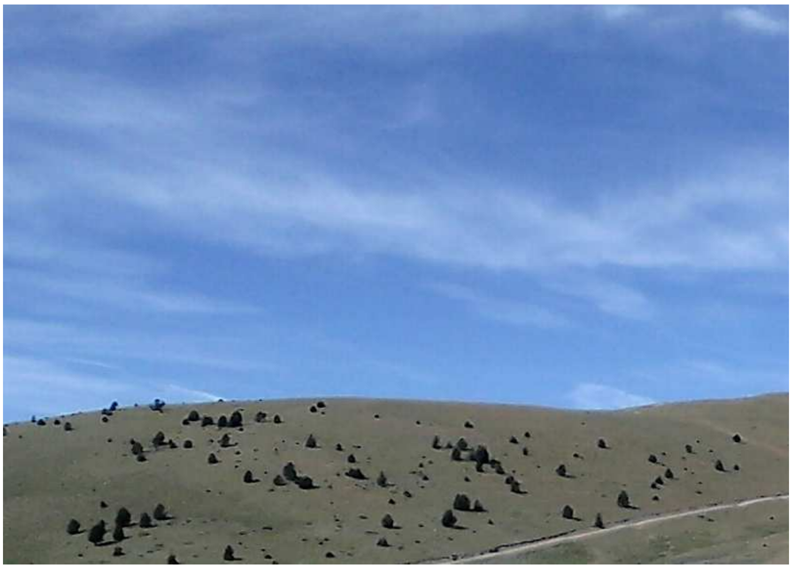


Figure 1.1: Colonization by *Pinus nigra* of the Alpine prairie stratum previously occupied by grasses and lichens. La Rabassa, Principality of Andorra.

Climate change at high altitude also may have effects on human activities, such as altering effective ski slope opening season lengths (Shardul Agrawala, 2007). Initial opening dates may need to be set back due to lack of snow, while closing dates may be affected by early spring snow melting, specially at the lower extent of ski slopes. This, in turn, has its effect on other economic processes related to winter tourism (e.g. hotel occupation, transport use patterns) (Marc Pons *et al.*, 2012).

These concerns are further enhanced by the fact that activities such as winter sports typically take place in installations that need large investments, both by private and public-sector investors. Ski slopes must be drawn up, and then constructed and moreover equipped with mechanical devices such as chairlifts or artificial snow production equipment. Road construction accessing the slope base stations is also a case in point. Such investments may see long amortization terms and need careful long-term planning.

Working conditions of the installations and in consequence the return on investment are affected by natural factors such as wind conditions and snowfall. Strong winds may increase the risk of avalanches on access roads, and prevent artificial snow production to take place on the ski slopes themselves. Natural snowfall may reduce a station's dependency on artificial snow and decrease production costs, or on the other hand load it may also increase snowdrift presence on access roads in ways that are not always easy to evaluate before the decision to invest in infrastructure is taken. This is where computer modeling comes in as a tool to help with the decision-making process.

Computer modeling is applied to many domains of human activity. Engineering and architectural applications include Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). Financial institutions use Computational Finance (CF) in recently developed applications such as high-frequency trading. As early as 1960, the implications of computing were evoked in relation to management decision-making processes (Herbert Simon, 1960). Computational Fluid Dynamics (CFD) is an area of computer-assisted model construction of fluid flows in general, that may be applied to the specific uses of wind circulation and snowfall.

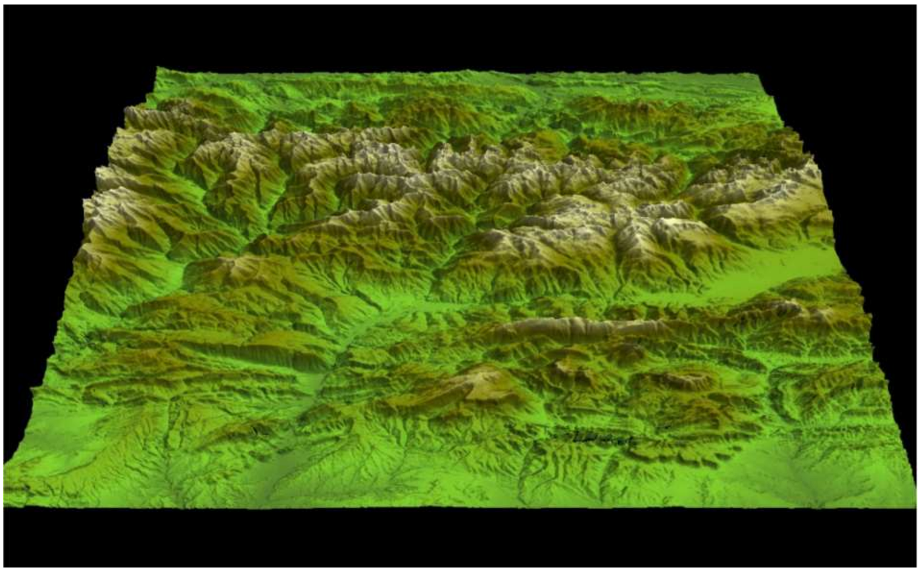


Figure 1.2: Digital Elevation Model of the Pyrenees Mountains, centered on the Principality of Andorra. Based on data from (T.G. Farr *et al.*, 2007).

High-quality topographic data is becoming available as public administrations share digital data collections with the general public, both as Digital Elevation Model data-sets (T.G. Farr *et al.*, 2007)(Figure 1.2) and in the form of vector-based multiple-layer information systems that combine topographical description with anthropic uses (Sigma map server, 2009).

At the same time, although perhaps the applicability of Moore's Law may in some cases be questioned since limitations such as interconnect speed (James D. Meindl, 2003) or power consumption (Ronald G. Dreslinski *et al.*, 2010) have arisen, it is clear still that computer equipment and processing power available to individuals and small organizations has evolved over the last decade and is still increasing year-on-year.

1.2 Goals and Contributions of this thesis

In this situation, it is attractive to use commercially available computing equipment -perhaps already installed in desktop computing or small research environments- to model specific situations as a tool to help decision-making when planning new sports and road installations in mountain areas. In this work, specific aims are modeling snowfall and secondary snow transport in scenarios including ski-slope planning and access road design, with a view to detecting situations in which snow accumulation or erosion could become inconvenient or even dangerous before construction or installation begins.

To do so, the first objective that must be addressed is to permit an efficient use of the consumer-grade computing equipment, that may be the only economically viable option when preparing small or medium-sized installations. This type of equipment has limitations both when used as individual computer platforms, and when used in a parallel computing environment. However, much attention has already been given to calculation efficiency in such an environment since the appearance of the Beowulf-type computing cluster (T. Sterling *et al.*, 1995). For this reason, this work will focus its attention on the making and verification of proposals regarding specific aspects of adapting existing techniques to modeling snowfall in mountain regions. Aspects treated include addressing the challenges posed by modeling mountain terrain, bridging the cap between large- and small-scale modeling, and handling the complex physical nature of snow particles and their transport.

1.2.1 Addressing the challenges posed by modeling the air volume above mountain terrain

Mountain terrain is complex in the sense that the ground layer in contact with the volume of air flowing over it may affect the direction and speed of airflow at a series of scales: large and small valleys, individual peaks and -at the lower end of the scale- individual large rocks or man-made obstacles.

These obstacles must be modeled when the air volume is discretized. However, the changes in surface normal direction give rise to elementary volume elements with uneven characteristics. These must be taken into account when the computer model is created. Such a model would need to represent in three dimensions the flow of air along the rock face, with its obstacles of various shapes and sizes.

> **Goal 1** of this thesis is *considering how a computer representation (mesh) may be built to accurately represent such complex terrain, and how to optimize the mesh to increase efficiency while solving the mathematical model applied using computers.*

1.2.2 Bridging the gap between large- and small-scale modeling

Local models of the effects of snowfall on individual buildings or installations have been constructed, such as (Sundsbø, 1998) or (Tominaga *et al.*, 2010). On the other hand, regional models also exist that concern snowfall across wide plains (Pomeroy *et al.*, 1993) or complete mountain slopes (Lehning *et al.*, 2006). However, little interaction has been observed between large- and small-scale models. In complex terrain, taking such interaction into account may give better understanding of the appearance of local phenomena.

To give an example, in mountain terrain it is common for enclosed valleys to have wind flow parallel to general valley orientation. A building sited at the center of such a valley can be foreseen to receive snowfall coming only from two privileged directions: either from the top or from the bottom of the valley. A similar building placed at the intersection of two converging valleys or in a higher and more exposed area may be expected to receive wind and snowfall from a larger variety of directions, thus complicating door and window placing during design.

For this reason, a computer model taking into account not only the immediate vicinity of the building but also the effects of orography in a wider area will give more precise results as to the interaction of wind flow and snow with the building itself.

> **Goal 2** of this thesis is *to study how a computer model created at regional level may be refined and applied to increasingly smaller areas, making an efficient use of existing CFD toolkits.*

1.2.3 Handling the complex physical nature of snow particles

Snowflakes form from the accumulation of water content, deposited onto ice crystals in the Troposphere. However, as has been known since the 17th century, snowflakes may take on a range of forms (Kepler, 1611). Many of these show the six-fold symmetry traditionally associated with snowflakes, such as the dendritic or star forms (Figure 1.3 a). These may break due to mechanical causes, forming simple needles or needle clusters (Figure 1.3 b).

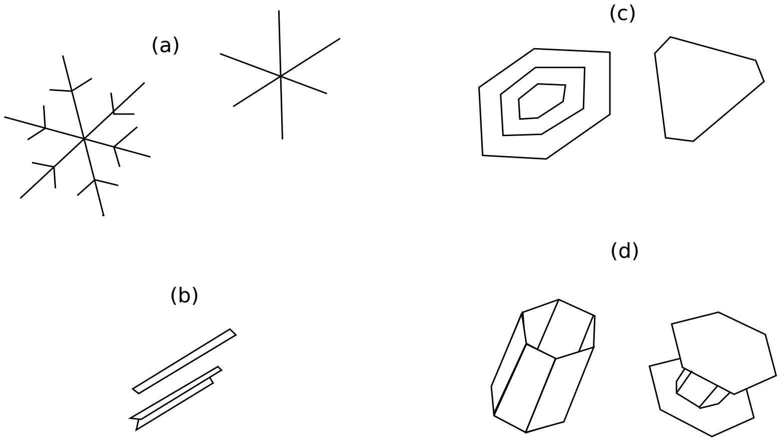


Figure 1.3: Various simple forms of snowflake. Adapted from (C. Magono and C. W. Lee, 1966)

Other snowflakes also show six-fold symmetry, though in the form of single or multiple planar shapes that may also exhibit complete or pseudo- triangle shapes (Figure 1.3 c). These have higher relative densities than dendritic forms. Hollow columnar forms are also possible, with either hexagonal or triangular bases and open or capped ends (Figure 1.3 d).

Finally, small snowflakes or the ice crystals resulting from breakage may become soldered together with supercooled droplets of liquid water. Upon contact, the water turns into ice thus permitting the accretion of crystals into various types of composite snowflakes or clusters. The most dense clusters are known as graupel. A complete and encyclopedic, description of ice crystals and snowflake formation may be found in (Vijay P. Singh, Pratap Singh and Umesh K. Haritashya, 2011), p.558 onward.

All these forms of snowflake exhibit different physical characteristics, such as average dimensions and density, that must be taken into account when snow is transported by the air flow.

To take a practical example, consider a section of road surface that is exposed to winds of similar intensity throughout the winter season. In the Northern Hemisphere, in November and December cold air produces relatively light snow that will be deposited on the road bed, but could very well be swept back off the road when snowfall has ended due to wind erosion. However, late winter and early spring snowfalls may exhibit larger and wetter snowflakes that will retain sufficient density so as not to be removed by the wind.

> **Goal 3** of this thesis is *identifying which parameters must be taken into account and built into the computer model to correctly represent the relationship between the snow flake and its physical characteristics, and the supporting airflow.*

1.3 Outline of thesis

This thesis is set out as follows:

In Chapter 2, a general theory of snow transport and deposition is presented. The Navier-Stokes equations for conservation of mass, momentum and energy are derived from basic physical principles, and applied to air flow for the specific objectives of this work. Possible simplifications are set out and discussed. Primary and secondary snow transport by the air flow is described.

Chapter 3 presents schemes for the discretization of the domain to be modeled using 2D and 3D mesh types. The relationship between Digital Elevation Model data and 3D meshes is discussed. The need for defining a measure of mesh quality is addressed. Such a measure is proposed, and used to compare existing optimization algorithms applied to mesh refinement. Experimental results of mesh optimization of air volumes above complex terrain are presented.

The OpenFOAM toolkit for Computer Fluid Dynamics is adapted for use in modeling transport fluid over mountain terrain in Chapter 4. General solver structure is explored, and the inner working of the toolkit PISO solver examined. Implications of its use in parallel computing environments are discussed. Experimental results concerning the computer workload arising from deformed mesh elements in contact with complex terrain formations are presented. Mesh decomposition strategies are discussed to take into account the varying workload caused by areas of different characteristics when designing a complete airflow model.

In Chapter 5, mixed and multiphase fluid flows are presented, and the degree of coupling between components of a mixed flow is analyzed with application to the mixed fluid formed by air and snow flakes. Computational aspects related to modeling snow parcel trajectories in a parallel computing environment are evoked. Experimental results are presented related to a specific test site.

Chapter 6 presents three case studies. In the first, the effect of existing air flow around snow making equipment on ski slopes are modeled, showing how local wind patterns and obstacles nearby can affect snow cannon planning and installation. The second case study concerns high altitude road management. A model of snowdrift formation is presented, and the influence of roadside slope shape on primary and secondary snow transport and snowdrift formation is analyzed. The effects of a large, stationary vehicle on local drift formation are modeled. Finally, in the third case study the transport fluid model is applied to the study of a Horizontal-Axis Wind Turbine implantation in a site on a high-mountain ridge. In conjunction with a complementary Blade Element Model of the wind turbine itself, it is shown that care must be taken in the choice of the implantation site to avoid the apparition of cyclic stress within the turbine structure leading to decreased power output and increased maintenance needs.

Chapter 2

Snow transport and deposition theory

In this chapter, a general theory of snow transport and deposition is presented, in preparation for the analysis of the objectives of this thesis undertaken in subsequent chapters. The Navier-Stokes equations for the movement of continuous flows will be studied and their possible simplifications in this specific application. The mechanisms of primary and secondary snow transport will be presented.

Snow transport is a complex physical phenomenon that must take into account the dynamics of two different fluids: snow on the one hand, and airflow on the other. Air can be modeled as a continuous fluid using the standard parameters of density $\rho(kg.m^{-3})$ and speed $U(m.s^{-1})$, to which other parameters such as pressure $p(Pa)$, temperature $\theta(K)$ or relative humidity $RH(\%)$ may be added to achieve a complete description. The various forms of ice crystal present in snowflakes have been studied since (Kepler, 1611), and described and documented photographically in modern times by investigators such as (Bentley and Humphreys, 1931) and (Nakaya, 1954). Snow cannot be seen as a continuum but presents itself in the form of flakes or even smaller particles (broken dendrites).

Each fluid may exert an influence on the other. The snow flakes are easily displaced by the airflow due to their low density and relatively large surface area, and tend to attain a stable terminal velocity relative to the air. But snow also increases slightly the density of fluid within each individual volume by its presence, so at first approximation we should consider the influence of either fluid on the other. However, considerations on the relative masses of each fluid permit us to consider only the influence of the air fluid phase on the snow and neglect the opposite effect, as shall be set out more fully in Chapter 5. It is for this reason that in this chapter we will consider air as the main fluid to be modeled, and snow as a secondary fluid whose motion is linked to that of air.

2.1 The Navier-Stokes equations

The well-known Navier-Stokes equations describe the movement of a continuous fluid through the application of three laws of conservation: conservation of mass, conservation of momentum and conservation of energy.

2.1.1 Conservation of mass

The principle of conservation of mass is known since the 18th century. It has been generally attributed to the chemist Antoine Lavoisier (Ebbing and Gammon, 2010) and described in his *Traité Elémentaire de Chimie* of 1789, though some controversy exists about a possible prior discovery by Mikhail Lomonosov (1711–1765) (Pomper, 1962). Simply put, this principle states that within a isolated system there is no net creation or destruction of mass, but that the system mass must stay constant over time. If we extend the concept of an isolated system to exclude not only transfers of mass but also of energy, the law remains valid even under the light of Albert Einstein's Theory of Relativity.

In Fluid Mechanics, it is more customary to consider not an isolated system, but a system that interacts with its environment. The principle of conservation of mass then given, in the absence of mass-energy conversions, as:

> "Within a given volume, the amount of mass accumulating inside the volume in a period of time is equal to the net difference between incoming and outgoing fluxes."

If we consider an elementary volume of the fluid within a static reference, also known as a "control volume" (Figure 2.1), we can calculate the mass of fluid within this volume at any one time as

$$dm = \int_V \rho \cdot dV \tag{2.1}$$

However, the control volume is at all times both gaining and losing mass as fluid passes through its surfaces; this variation of mass may on the one hand be derived from the equation above

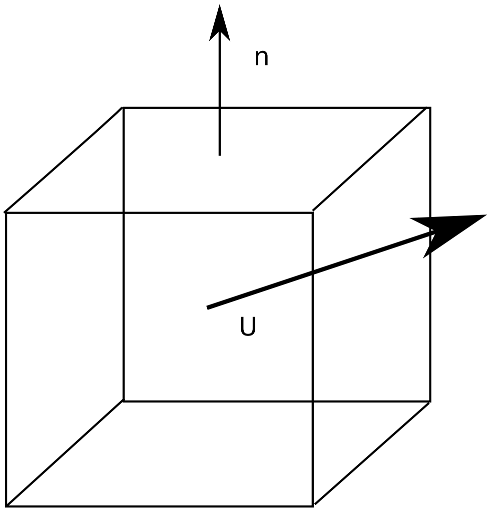


Figure 2.1: Flow through a control volume.

$$\frac{dm}{dt} = \frac{d}{dt}\int_V \rho \cdot dV \tag{2.2}$$

but it may also be found by integrating outwards flows on the surface of the volume

$$\frac{dm}{dt} = -\int_S \rho\vec{u} \cdot \vec{n} \cdot dS \tag{2.3}$$

Equaling both equations, we find the continuity equation

$$\frac{d}{dt}\int_V \rho \cdot dV = -\int_S \rho\vec{u} \cdot \vec{n} \cdot dS \tag{2.4}$$

Using Gauss's theorem, the integral over the surface can be written as

$$\int_S \rho\vec{u} \cdot \vec{n} \cdot dS = \int_V \nabla \cdot (\rho\vec{u}) \cdot dV \tag{2.5}$$

giving us

$$\frac{d}{dt}\int_V \rho \cdot dV = -\int_V \nabla \cdot (\rho\vec{u}) \cdot dV \tag{2.6}$$

But this equation must be verified for any arbitrary choice of control volume V in the fluid. For this reason, we can do away with the integration on both sides and retain that

$$\frac{d}{dt}(\rho) = -\nabla \cdot (\rho\vec{u}) \tag{2.7}$$

This can then be reformulated in the classical form of the continuity equation with all terms concerning density and movement on the left-hand side:

$$\frac{d}{dt}(\rho) + \nabla \cdot (\rho\vec{u}) = 0 \tag{2.8}$$

2.1.2 Conservation of momentum

The principle of conservation of momentum is a consequence of Isaac Newton's three Laws of motion, as set out in *Philosophiae Naturalis Principia Mathematica* (1687). The Second Law reads:

> "Mutationem motus proportionalem esse vi motrici impressae, et fieri secundum lineam rectam qua vis illa imprimitur."

This may be translated as:

> "The changes in movement are proportional to forces applied [to the object considered], and [such changes] are made in a straight line along the direction on which the forces are applied."

This is rendered in more modern mathematical terms by the equation:

$$\sum \vec{F} = m \cdot \vec{a} \tag{2.9}$$

However, acceleration $\vec{a}$ is defined as the derivative of speed U in relation to time:

$$\vec{a} = \frac{d\vec{u}}{dt} \tag{2.10}$$

Under the supposition the system considered has constant mass, $\frac{dm}{dt} = 0$ and so:

$$m \cdot \vec{a} = m \cdot \frac{d\vec{u}}{dt} = \frac{d(m \cdot \vec{u})}{dt} \tag{2.11}$$

The product of mass and velocity $m \cdot \vec{u}$ appears, and is called momentum. Newton's Second Law may be rewritten as:

$$\sum \vec{F} = \frac{d(m \cdot \vec{u})}{dt} \tag{2.12}$$

In an isolated system, we have $\sum \vec{F} = 0$, giving us the principle of conservation of momentum:

$$\frac{d(m \cdot \vec{u})}{dt} = 0 \tag{2.13}$$

In the same way as in the previous section, the momentum of the mass of fluid contained within control volume V may be calculated as:

$$\vec{M} = \int_V \rho\vec{u} \cdot dV \tag{2.14}$$

For momentum also, the amount of fluid momentum may vary in time. The variation may be calculated as a time differential:

$$\frac{d\vec{M}}{dt} = \frac{d}{dt} \int_V \rho\vec{u} \cdot dV \tag{2.15}$$

Alternatively, the variation may also be considered as a balance between momentum conveyed by flows entering and exiting the control volume. However, in this case we must also take into account both volume $\vec{f_V}$ and surface $\vec{f_S}$ forces exerted on the fluid:

$$\frac{d\vec{M}}{dt} = -\int_S \vec{u} \cdot (\rho\vec{u} \cdot \vec{n}) dS + \int_V \vec{f_V} \cdot dV + \int_S \vec{f_S} \cdot dS \tag{2.16}$$

Equaling both equations, we obtain:

$$\frac{d}{dt} \int_V \rho\vec{u} \cdot dV = -\int_S \vec{u} \cdot (\rho\vec{u} \cdot \vec{n}) dS + \int_V \vec{f_V} \cdot dV + \int_S \vec{f_S} \cdot dS \tag{2.17}$$

Surface forces $\vec{f_S}$ are forces within three-dimensional space, that are applied to each point of the control volume's external surface. Their standard notation depends on the introduction of the mathematical concept of **tensor**. This geometrical object may be seen as an extension of the two-dimensional matrix: a one-dimensional vector is equal to a first-order tensor, while a matrix is a second-order tensor. Tensors of orders three and four are usual in Fluid Dynamics, Electromagnetism, Relativity and other fields.

Tensors admit the usual matrix operations such as addition and products. The usual matrix product is known as the *inner product* of two tensors and noted $A.B$. As is the case for matrices, the order of the resulting tensor will be the sum of operand orders, minus 2:

$$order(A \cdot B) = order(A) + order(B) - 2 \tag{2.18}$$

There also exists a specific tensorial *outer product*, noted $A \otimes B$. In this tensorial product, each specific element of each operand is combined with the totality of elements of the other operand to form a tensor; the final product may thus be seen as a tensor of tensors. Its order is the product of operand orders:

$$order(A \otimes B) = order(A) \cdot order(B) \tag{2.19}$$

Surface forces are then written as:

$$\vec{f_S} = \vec{n} \cdot \sigma \tag{2.20}$$

Here σ is the stress tensor $\sigma = [\sigma_{i,j}]$, a second-order tensor. This mathematical representation combines in a single notation two different types of force: pressure that is spherically symmetric in all directions, and other forces that are non-symmetric. σ is then decomposed into a spherical component and a non-spherical using:

$$\sigma = -p.I + \tau \tag{2.21}$$

With p the thermodynamic pressure, I the spherical identity tensor, and $\tau = [\tau_{i,j}]$ the viscous stress tensor. Replacing these into the integral of surface forces, we obtain:

$$\int_S \vec{f_S} \cdot dS = \int_S (n \cdot \sigma).dS = \int_S -p.I \cdot \vec{n} \cdot dS + \int_S n \cdot \tau \cdot dS \tag{2.22}$$

This gives the integral expression for conservation of momentum:

$$\frac{d}{dt} \int_V \rho \vec{u} \cdot dV = -\int_S \vec{u} \cdot (\rho \vec{u} \cdot \vec{n}) dS + \int_V \vec{f_V} \cdot dV + \int_S -pI \cdot \vec{n} \cdot dS + \int_S \vec{n} \cdot \tau \cdot dS \tag{2.23}$$

Using Gauss's theorem, all surface integrals can now be transformed into integrals over volume:

$$\frac{d}{dt}\int_V \rho\vec{u}\cdot dV = -\int_V \triangle(\rho\vec{u}\otimes\vec{u})dV + \int_V \vec{f_V}\cdot dV + \int_V -\nabla p\cdot dV + \int_V \triangle(\tau)\cdot dV \tag{2.24}$$

Once more, this integral equation must hold for any arbitrary choice of control volume. It holds that for each point:

$$\frac{d}{dt}(\rho\vec{u}) = -\triangle(\rho\vec{u}\otimes\vec{u}) + \vec{f_V} - \nabla p + \triangle(\tau) \tag{2.25}$$

In the equation's canonical form:

$$\frac{d}{dt}(\rho\vec{u}) + \triangle(\rho\vec{u}\otimes\vec{u}) = -\nabla p + \triangle(\tau) + \vec{f_V} \tag{2.26}$$

In this equation, the first term on the left hand side represents density variation over time, and is often called the **unsteady term**. The second represents convection within the fluid, and is called the **convection term**. On the right hand side, we find the **pressure term**, the **stress term** and finally the term representing volume forces and which in any practical applications is reduced to a **gravitational term**.

2.1.3 Conservation of energy

The application of the first law of thermodynamics on the control volume gives us the following equation of conservation of energy:

$$de = p.dv + \theta.ds \tag{2.27}$$

Here e is the specific internal energy per unit of mass, p pressure combined with viscous stresses in the case of a viscous fluid (Drikakis and Rider, 2005), θ temperature and s specific entropy of the control volume.

This equation reflects how the variations of internal energy o the control volume depend on the one hand on variations of potential energy due to volume, and on the other on entropy. If needed, an internal heat source $\tilde{q}$ may be added to describe chemical reactions occurring within the fluid, or a heat flux vector $\vec{f}$ to describe the transfer of heat by conduction.

2.1.4 Application to air flow and simplifications

The equation of conservation of mass or continuity equation 2.8 derived in section 2.1.1 is:

$$\frac{d}{dt}(\rho) + \nabla(\rho\vec{u}) = 0 \tag{2.28}$$

This equation permits no further simplification, unless the fluid is considered to be ***isotropic***. Such a fluid has the same properties in every direction. Specifically, under this assumption ρ no longer depends on direction, and the first equation may be written as:

$$\frac{d}{dt}(\rho) + \rho\nabla\vec{u} = 0 \tag{2.29}$$

Fluid speeds to be considered will be low, in any case much lower than the speed of sound within air:

$$|\vec{u}| \ll c_{air} \tag{2.30}$$

This so justifies the use of the supposition that the fluid is ***incompressible***. Under this assumption, volume does not vary according to pressure: density ρ no longer depends on pressure. Since at steady-state pressure p is stable over time, so is density, giving $\frac{d}{dt}\rho = 0$ and transforming the equation into the simplified (Eulerian) form of the conservation of mass:

$$\nabla\vec{u} = 0 \tag{2.31}$$

As for the conservation of momentum equation 2.26 derived in section 2.1.2:

$$\frac{d}{dt}(\rho\vec{u}) + \triangle(\rho\vec{u}\otimes\vec{u}) = -\nabla p + \triangle(\tau) + \vec{f_V} \tag{2.32}$$

This equation can benefit from a first simplification by considering the fluid to be ***Newtonian***. Under this supposition, each element of viscous stress tensor τ is considered to be proportional to the rate of deformation, such that:

$$\tau = 2\mu D + \lambda\delta_{i,j}\triangle\vec{u} \tag{2.33}$$

Here, in the first term a viscosity coefficient μ is introduced along with the tensor of deformation D given by:

$$D = \frac{1}{2}[\nabla\vec{u} + (\nabla\vec{u})^T] \tag{2.34}$$

In the second term bulk viscosity coefficient λ and the Kronecker delta function. In this second term, it is usual to use the approximation:

$$\lambda \approx -\frac{2}{3}\mu \tag{2.35}$$

This gives:

$$\tau = 2\mu D - \frac{2}{3}\mu\delta_{i,j}\triangle\vec{u} \tag{2.36}$$

The viscosity of a fluid corresponds to the internal friction that tends to oppose gradual deformations. These cause the appearance of shear stresses within parts of the fluid moving at different relative velocities. All real fluids have some degree of viscosity, except for super-fluids that are by definition friction-less. However, fluid viscosity varies considerably: for example, the viscosity of an ideal gas may be calculated using Sutherland's Formula (Smits and Dussauge, 2006) giving $18.6 \cdot 10^{-6} Pa.s$ at 27 degrees C , while that of water is $8.94 \cdot 10^{-4} Pa.s$ at 25 degrees C (Linstrom and Mallard, eds.). This difference in viscosity leads to the simplification of the equation of conservation of momentum, when applied to gases, by considering the fluid to be ***inviscid***, also called an ***Eulerian*** fluid.

Applying this supposition to the air medium during snow transport is equivalent to considering inertial forces to be much higher than the internal stresses, allowing us to consider the stress tensor τ to be null. However, when applying such reasoning to problems such as modeling snowfall, the airflow is in contact with the ground surface. For this reason, boundary layer effects exist that must be taken into account to represent the transmission of stresses from the surface up into the fluid.

Finally, variations of density ρ in the fluid are linked to temperature θ, and may be the cause of fluid motion. However, if temperature is considered to be stable within the volume of fluid, density variation will not be large and one may treat density as constant in the unsteady and convection terms, and variable only within the gravitational term. This is the ***Boussinesq approximation*** (Ferziger and N. Perić, 2002)[1]. It is specially appropriate when the only volume force considered is gravitation, and the resulting simplified equation for momentum is given as:

$$\rho\frac{d}{dt}(\vec{u}) + \rho\triangle(\vec{u}\otimes\vec{u}) = -\nabla p + \triangle(2\mu D - \frac{2}{3}\mu\delta_{i,j}\triangle\vec{u}) - \rho \cdot g \tag{2.37}$$

As for the conservation of energy:

$$de = p.dv + \theta.ds \tag{2.38}$$

[1] page 15

Although this equation has been included here for the sake of completeness, it shall not in fact be needed within the scope of the present work. On the one hand the fluid will be considered as incompressible, nullifying the term $p \cdot dv$. On the other hand, the standard temperature gradient $0.0065 K.m^{-1}$ described in the US Atmosphere Model (Talay, 1975; NASA, 1976) for the Troposphere shows us that no great differences in temperature need to be considered within the lower layers of air in contact with the surface. Using these two assumptions, in fact internal specific energy is considered to be constant for each control volume.

2.2 Primary and secondary snow transport

Primary snow transport occurs when snow is initially deposited through the falling snow process. If wind speeds are low during initial deposition, snow accumulates in mainly vertical shapes (Figure 2.2).

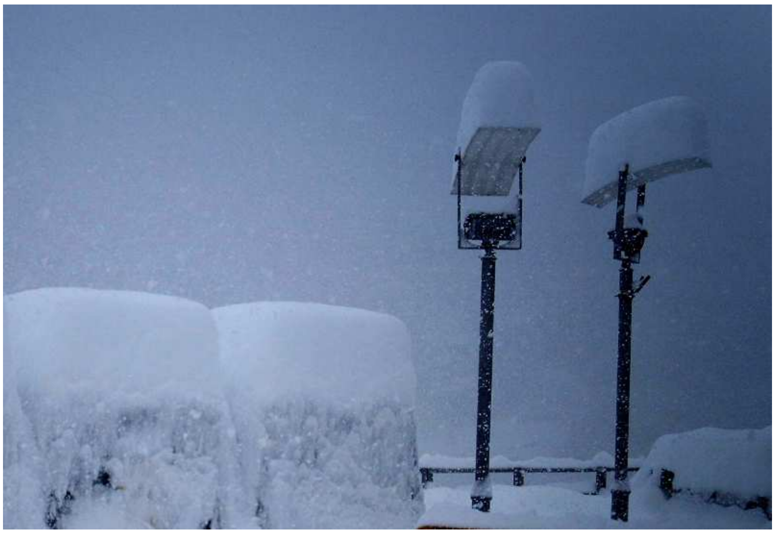


Figure 2.2: Undisturbed primary snow deposition on containers and streetlights. Ordino, Andorra..

Secondary transport, on the other hand, concerns the erosion by wind of snow already deposited in some places, and re-deposition in others. Through this mechanism, the snow layer once formed may be sculpted into new forms though wind action and local accumulations (snow-drifts) formed (Figure 2.3).

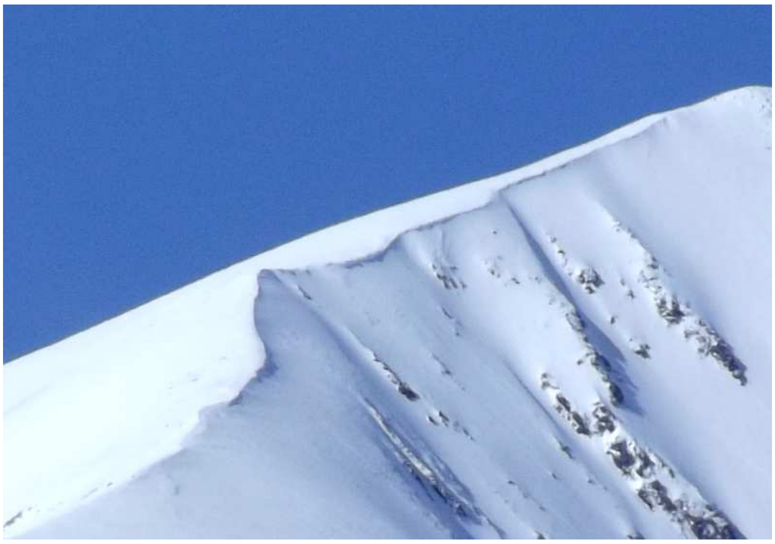


Figure 2.3: Snowdrifts formed by wind on a ridge. Pic de Salòria, Alt Urgell, Catalonia.

2.2.1 Primary snow transport

Snowflakes may be seen as aggregates of ice crystals. As with raindrops, ice crystals are formed around a condensation nucleus, and the resulting crystal will in turn coalesce with other crystals, growing with the adjunction of water captured from nearby droplets, to form final aggregates of varying shapes: needles, plane hexagonal crystals, plane and spatial dendritic crystals (the "traditional" conception of a snowflake), or more complex amalgamates of solid crystal with liquid water such as graupels. This process had already been partially studied, among others, by René Descartes in the 17th century.

At the beginning of the 20th century, (Schmidt, 1909) had already estimated terminal velocity of raindrops in still air, measurements which became more precise during the 1940's (Spilhaus, 1948; Gunn and Kinzer, 1949). However, the dynamics of snowfall measurements are more involved than for rain due to two factors:

1. Snowflakes present a larger surface area for a given weight than raindrops.

2. Individual snowflakes show large variations of form, increasing or decreasing the surface area/weight within a large range (Nakaya and Terada, 1935).

For these reasons, although the basic mechanisms are known and studies of snowflakes terminal velocities have been published since the middle of the

20th century (Langleben, 1954), more recent studies (Hanesch, 1999; Schefold *et al.*, 2002) have taken into consideration snowflake axial ratios and shape to present more precise forms of the power law:

$$v_{term} = k.D^{\alpha} \tag{2.39}$$

Here v_{term} is the snowflake terminal velocity, D the diameter, k and α two constants that depend mostly on the degree of rimming of the flake (number of frozen cloud droplets on its surface). In relation to this equation, it is interesting to note that since snowflake shapes and diameters show a large variation, so must terminal velocities. This corresponds well to empirical observations during which one can see, for example, how dry snow falls slowly on a cold night, contrary to late spring wet snow that falls in a fashion similar to raindrops. For the computer modeler, the implications are that airmass humidity and temperature are of importance to determine flake terminal velocity and thus snowflake behavior within the air flow.

2.2.2 Secondary snow transport

Secondary snow transport mechanisms have received much attention since Masao Takeuchi's seminal work on simple snow transport (Takeuchi, 1980). This investigator used a frozen river as a flat surface over which wind flow blew up snow from one riverbank. Using series of snow-traps situated at varying heights above ground-level, he was able to evaluate the distances over which blowing snow may be transported through three different mechanisms (Figure 2.4). In the first place, snow particulate matter that has been carried high enough ($\approx 3m$ above the ground) is carried along by the wind in a state of ***turbulent suspension***. The vertical movement will eventually decay and particles will fall back to the ground.

In the second place, there exists a form of transport within close proximity ($\approx 1 - 2cm$) to the surface under which snow particles progress through a rolling motion. This is known as ***creeping***.

Finally, ***saltation*** consists in particles that are torn off the surface by wind action, arise to intermediate heights ($\approx 1m$ above the ground), and fall only to bounce back and continue their transport. This is the form of secondary snow transport that accounts for most of the volume transported.

Other investigators have built upon Takeuchi's work in order to study snow transport altered by the presence of obstacles. For example, (Uematsu *et al.*, 1991) considers a regular obstacle built up of two transportable building modules, and the formation of accumulation zones, ridges as well as completely uncovered areas in the snow layer around it. More recent work

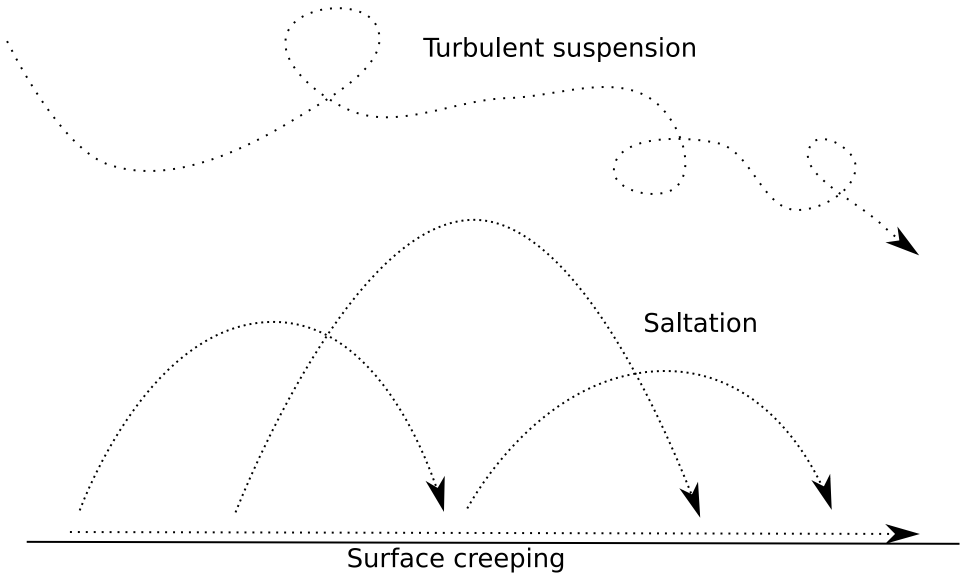


Figure 2.4: Various secondary snow transport mechanisms.

has scaled up to consider the interaction of snow and wind in relatively large-scale ($\approx 100m$) completely build-up urban scenarios, such as (Tominaga *et al.*, 2010).

On the other hand, computer-based models have been introduced to help research the formation of snow-drifts arising from secondary transport. The Prairie Blowing Snow Model (PBSM) was developed in order to model transport in flat environments (Pomeroy *et al.*, 1993), and applied *inter alia* to the study of the formation of snowdrifts along communication lines in Kansas (USA). In another development, the commercial CFD code FLOW-3D written in the Fortran programming language was adapted to model snow drift formation around building steps (Sundsbø, 1998) and cubic obstacles (Beyers *et al.*, 2004). This has been applied to several practical problems concerning snow build-up around the buildings in scientific establishments in the Arctic.

A further development was SnowTran-3D, destined to model fluxes on topographically variable terrain (Liston and Sturm, 1998) and into which forcing by meteorological data was introduced (Greene, 1999). Finally, the current generation of models include ALPINE3D (Lehning *et al.*, 2006), based on a radiation balance model. These more recent models are often designed to consider to take into account snow pack transformation (e.g. through sublimation) and interaction with natural or man-made influences (e.g. vegetation cover). Interest has also been shown in covering a period of time of several months, allowing the evolution of the snow cover in a natural scene to be modeled over an entire winter season.

2.3 Chapter conclusions

In this chapter, the Navier-Stokes equations governing the movement of continuous flows have been derived from the three principles of conservation of mass, momentum and energy:

- Conservation of mass:

$$\frac{d}{dt}(\rho) + \nabla \cdot (\rho\vec{u}) = 0 \tag{2.40}$$

- Conservation of momentum:

$$\rho\frac{d}{dt}(\vec{u}) + \rho\triangle(\vec{u}\otimes\vec{u}) = -\nabla p + \triangle(2\mu D - \frac{2}{3}\mu\delta_{i,j}\triangle\vec{u}) - \rho\cdot g \tag{2.41}$$

- Conservation of energy:

$$de = p\cdot dv + \theta\cdot ds \tag{2.42}$$

The application of these equations to air flow and various suppositions allowing the simplification of the mathematical expressions have been presented. The mechanisms of primary snow transport and initial deposition have been described, as well as secondary transport through turbulent suspension, saltation and surface creeping.

In the next chapter, the physical domain under study shall be prepared for mathematical modeling through volume discretization and its optimization. This shall permit establishing the connection between the physical quantities to be modeled (air density and flow speed), and the shape of the domain.

Chapter 3

Optimizing domain discretization

A computer model of a fluid dynamics system must set limits on the physical volume which is modeled. In most cases, a regular volume is preferred in order to simplify description; a cylinder can represent the interior of a segment of pipe in a hydraulic application, or a rectangular volume may represent a volume of air surrounding an aircraft. In some cases, the geometry of the problem set is such that several individual regular shapes must be combined to represent the physical domain, for example in the case of the now classical T-junction pipe (Paritosh R. Vasava, 2007). In yet further cases, it is difficult to represent a complex physical volume using elementary shapes due to its irregularity, as in the case to the complex orography discussed in this work and in previous work (A. Ward and J. Jorba, 2011).

The first task in planning a numerical model of a particular physical domain is to decompose it up into discreet elements (*cells*). These will form a mesh of individual planar elements (in 2D), or volumes (in 3D). It is on these cells that in this chapter the equations of continuity described previously will be discretized, a system of linear equations will be derived and then solved. Specific attention will be given to physical domains consisting of the air volume above mountain terrain, constructed as a rectangular volume with the terrain as its lower boundary. In the context of modeling spatial domains situated over mountain terrain, domain decomposition has to contend with complex orographies: sharp ridges, serrations and individual peaks. All give rise to situations in which the mesh cells take on irregular shapes, specifically those in the mesh layer closest to the terrain.

This chapter is specifically concerned with **Goal 1** of the thesis.

3.1 Mesh types and formation

When a 2D mesh is constructed, planer cells will usually consist of forms such as triangles, quadrangles (Giles, Mike, and Robert Haimes, 1990). In some cases, hexagons may also be used either directly e.g. (W. Kelly and B. Gigas, 2003), or as a mesh structure or as a support for further mesh construction (Xinghua Liang and Yongjie Zhang, 2011).

The cells in mathematical terms form a tessellation of the surface under consideration. They may form either a structured or unstructured mesh, depending on the regularity of their placement.

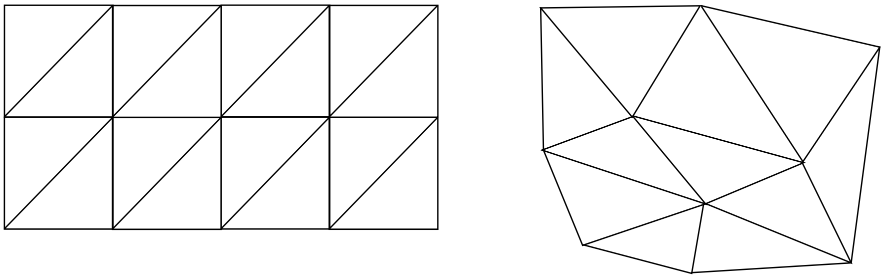


Figure 3.1: Structured and unstructured 2D meshes formed by triangular cells.

In structured meshes (Figure 3.1), placing each cell with the same spatial relationship to preceding and following cells along the two axis of a reference allows -by the way of a well-thought-out numeration of cells- obtaining a system of equations in the form of a banded matrix (Pulliam, 1986). This type of sparse matrix may then be inverted and the system of equations solved using iterative methods.

$$\begin{bmatrix} A & B & C & & & 0 \\ B & \ddots & \ddots & \ddots & & \\ C & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & C \\ & & \ddots & \ddots & \ddots & B \\ 0 & & & C & B & A \end{bmatrix}$$

In the 3D case, the usual forms for individual cells are either tetrahedra (convex polyhedra with four vertices, and four triangular faces), or hexahedra (convex polyhedra with eight vertices and six faces). These cells form a tessellation of the 3D volume occupied by the fluids in a CFD model. In the same way as above, a regular structured mesh will give rise to an ordered system of equations with non-zero matrix elements placed in bands near the main diagonal.

Each matrix line describes a mesh cell and its relationship with neighboring cells. However, in the 3D case each mesh cell has more neighbors than in 2D, thus rendering matrices with more bands than in a planar mesh.

There has been some discussion about the merits of each form of spatial decomposition: tetrahedral versus hexahedral. Since mechanically-generated Digital Elevation Models (DEMs) are given in the form of raster images, conversion to hexahedral cells is a straightforward process. Conversion of a square height matrix into tetrahedra, while possible (Albertelli and Crawfis, 1997), is an involved process that may generate tetrahedra with square or obtuse angles in one vertex, while other vertices will have sharp angles. On the other hand, a square mesh will make an ideal candidate for a 2D terrain grid, with a hexahedral extension in 3D.

An additional argument is that leaving the mesh in hexahedral form gives rise to less individual cells, and thus reduces the number of equations generated when solving the fluid dynamics problem (Weingarten, 1994).

That being said, it should also be noted that some algorithms such as particle streamline tracing (Kenwright and Lane, 1996) only work with tetrahedral spatial decompositions. This particular technique has applications in detecting recirculation areas (vortices) within the fluid (Kenwright, 1998; S. Bryson *et al.*, 1999) such as those in Figure 3.2.

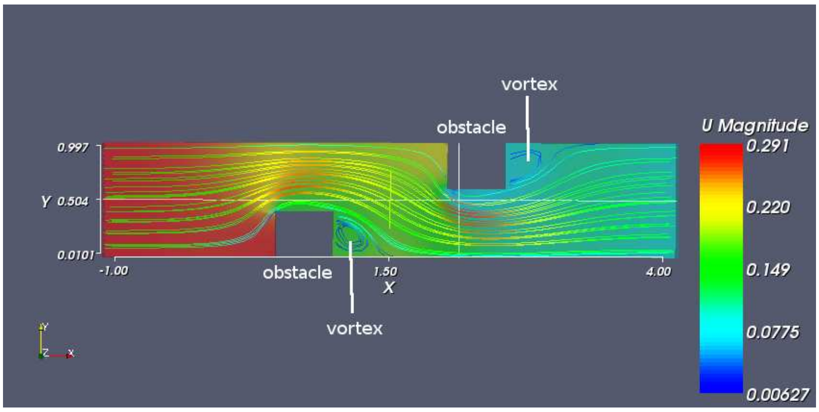


Figure 3.2: Streamlines and vortices in a model of a weir supplying a waterwheel. Adapted from (Alan Ward, 2008).

Suffice it to say that both positions for tetrahedra and for hexahedra have their proponents.

When modeling spatial domains situated over mountain terrain using hexahedra, deformed volumes appear especially in the layer in contact with the terrain, that must follow its shape. When the volume of air above glacier cirque is modeled (Figure 3.3), irregular hexahedra are notable in the vicinity of the horizontal ridges above glacier cirques, and wherever smaller ridges lead downwards into the valleys below.

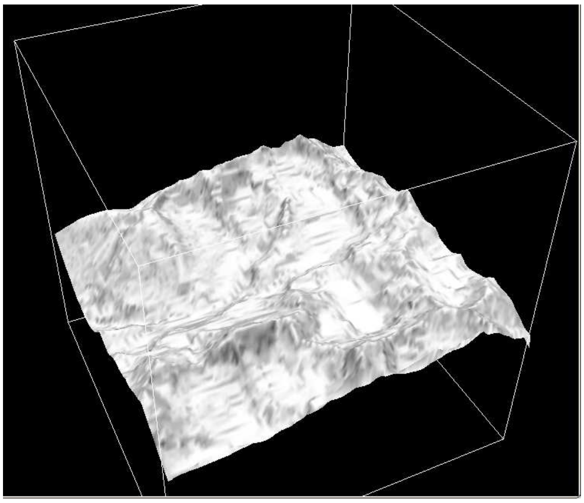


Figure 3.3: 3D representation of a Digital Elevation Model of a glacier cirque.

It has been pointed out among others by (Ferziger and N. Perić, 2002) [1] that, even though in some situations it is impossible to have a grid that is completely orthogonal at cell junctions, it is still important to make such junctions as regular as possible. As this author states, when performing numerical calculations "parts of the errors made at opposite cell faces cancel partially if cell faces are parallel, and completely when opposite cell faces are parallel and of equal area." An important aim of mesh generation should thus be to create individual hexahedral cells with as regular a geometry as possible and angles approaching orthogonality.

3.2 The need for a measure of mesh quality

A criterion of the quality of individual mesh elements has been proposed in (Delaunay, 1934) for planar meshes, and long been used as the "Delaunay criterion". Given concisely, it states that an individual quadrangular cell is of "sufficient quality" when none of its vertices is situated within the circle circumscribing the triangle made up of the other three points. When a quadrangular mesh is split into triangular elements, the same criterion

[1] pp. 219 - 220

may be satisfied by avoiding placing any vertex within the circumcircle of any triangle not containing the vertex. This idea has been applied to the meshing of 2D areas, for example in (Lawson, 1977; Watson, 1981).

When this idea is applied to regular hexahedral 3D meshes as in (Baker, 1989; George *et al.*, 1991; Weatherill and Hassan, 1994), the criterion becomes (Figure 3.4):

> No mesh vertex should be placed within the circumsphere of a mesh cell to which the vertex does not pertain

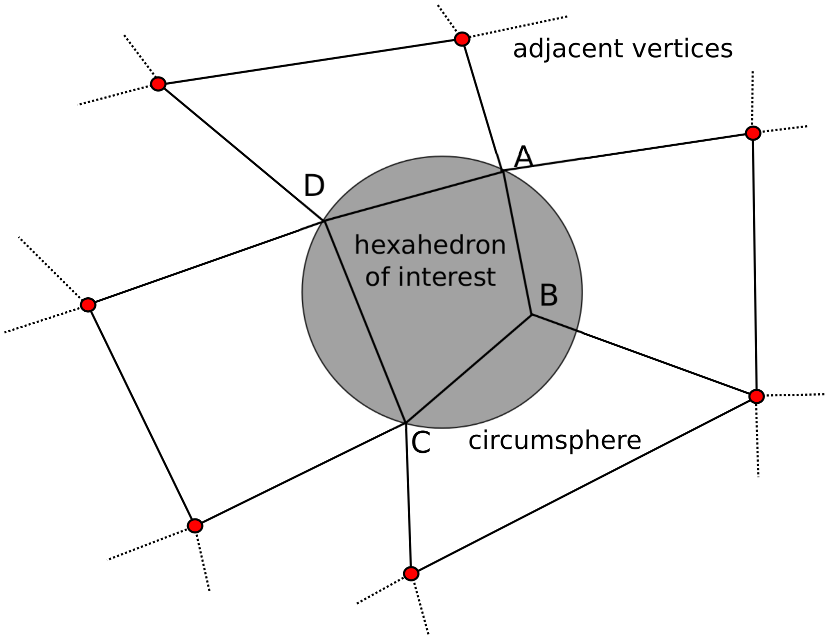


Figure 3.4: Cut through a hexahedral mesh cell along one facet. A to D: vertices of the hexahedron of interest forming a facet. Shaded area: projection of circumsphere. Red circles: adjacent mesh vertices.

From the standpoint of the computer modeler, iterative algorithms may be used to either construct distinct meshes with successively increasing quality (mesh refinement), or to set out from an initial mesh which is then progressively modified to achieve higher quality (mesh smoothing). In either case, it is necessary to obtain a measure of mesh quality in order to follow the progression of the algorithm that has been used, and as a means of implementing a condition for termination. This measure of quality must be applicable both to individual mesh cells or regions, and to the complete mesh as a whole (e.g. by addition of individual mesh qualities).

Four aspects of a well-formed quality measurement technique are set out in (Field, 2000):

1. An ability to detect all degenerate elements. An element is considered degenerate when at least two of its vertices have been merged, leading to a zero-length edge between them. A degenerate element will no longer retain all the properties of a hexahedron: number of faces, number of edges, etc.

2. Non-dimensionality. The measurement technique should give the same results, regardless of the units used to measure physical mesh properties (meters, inches, degrees, radians etc.). Quality should thus be independent of mesh scale.

3. Boundedness. The values of quality should be bounded. In practice, measures of quality using a technique with an open-ended scale such as $]-\infty;+\infty[$ may be converted to a bounded measure through the use of an appropriate bijective function. However, in this case, it may be noted that extreme values will never be achieved.

4. Normalization. This property is in fact a consequence of the previous, since any bounded interval $[a;b]$ may be transposed into $[0;1]$ without losing the possibility of comparing individual values.

Many existing measurement techniques have been proposed. The earliest was by calculating the aspect ratio of areas or volumes, by (Parthasarathy and Kodiyalam, 1991; Lee and Lo, 1994) and many others. This technique is useful and not too costly in computational terms for plane triangular elements, since the number of individual distances within the triangle is small. However, when applied to hexahedral 3D elements, a total of 28 distances must be calculated for each mesh cell: 12 along edges, 12 along face diagonals and 4 volume diagonals.

For this reason, other schemes such as the Condition Number objective function (Knupp, 2001) or methods based on bilinear functions (Robinson, 1987) have been drawn up. However, these methods are targeted mostly at problems in 2D, and have later on been modified for application in 3D. But the calculation needed for their evaluation are much more computationally expensive in 3D.

This is why a different approach for this problem is proposed. Angle-based optimization for triangles (Freitag and Ollivier-Gooch, 1995), (Zhou and Shimada, 2000) has been known for many years, and used for quality measurements mostly by searching for small angles. On the other hand,

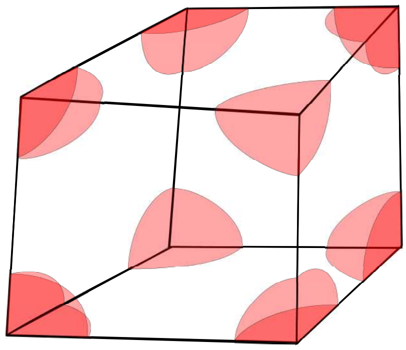


Figure 3.5: Solid angles in a hexahedral mesh cell.

little attention have been given to the use of measures based on solid angles in 3D, with the most notable exception being (Frietag and Ollivier-Gooch, 1997). This approach may be more efficient for optimization smoothing in 3D meshes for the following reason: when a single vertex's position is modified, quality measures for all 8 adjacent hexahedra must be recalculated. If the measure is based on aspect, at least 26 distances must be calculated (between the displaced vertex and all adjacent vertices). If a measure based on the Jacobian matrix is used, this matrix must be newly treated for all adjacent hexahedra. However, if a measure based on solid angles is used, only 8 solid angles must be recalculated which can be done by vectorial means and may be efficiently implemented in a computer program (Figure 3.5).

Freitag's works are based on the detection and avoidance of small angles. A possible approach would be to compare the minimal angle in a hexahedron with the $\frac{\pi}{2}$ value for a regular hexahedron's solid angle. However, this would not allow one to distinguish between hexahedra with just one vertex slightly displaced from its optimal position (one small angle reduced, all other angles have a similar value) and very deformed hexahedra (various different angular values) (Figure 3.6). For this reason, it is preferable to use the following measure for quality, in which both minimum and maximum solid angles participate in the determination of quality:

$$q = \begin{cases} \frac{\max(\alpha_i)-\min(\alpha_i)}{\max(\alpha_i)} & \text{if all } \alpha_i \text{ are defined} \\ 1 & \text{otherwise} \end{cases} \tag{3.1}$$

Here (α_i) are the interior solid angles of an individual hexahedron. They will be defined when the corresponding vertex of the hexahedron has not

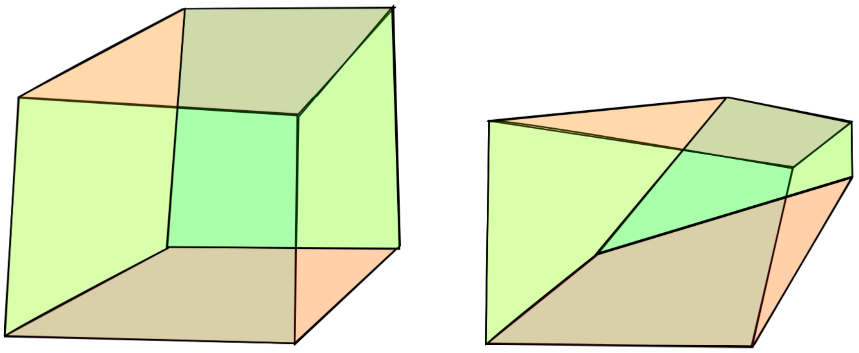


Figure 3.6: Almost regular and deformed hexahedra.

been merged with any adjacent vertex, leading to a zero-length edge.

It should be noted that maximum quality (i.e. very regular meshes) is identified by lowest values of the measurement function: the function may be taken as a value of deformation, to be minimized by smoothing algorithms.

This measure of mesh quality for each individual hexahedron satisfies the four conditions given by Field:

1. Degenerate elements - in which any edge or face is null - have measure value $= 1$.

   Non-degenerate elements, but with large deformations ("slivers"), have values approaching 1. On the other hand, nearly regular elements have measure values approaching 0.

2. Non-dimensionality is achieved by the use of angles. Similar-shaped hexahedra will have the same angles, and thus give the same quality value.

3. This measure is trivially contained within interval $[0, 1]$, giving both boundedness and normalization.

In order to evaluate the entire mesh, it is proposed to simply average this quantity over all hexahedra in the mesh. This technique is flexible enough to be applicable to part of a mesh if desired. This is a desirable characteristic, since calculation of solid angles requires more effort by a computer than that of distances. When only a subset of vertices are displaced in a given iteration, only adjacent hexahedra quality measurements need be recalculated.

Avoiding having to recalculate the quality of the entire mesh is thus a means of shortening algorithm execution times.

3.3 Optimizing mesh quality

To achieve meshes of high quality, authors such as (Field, 2000) proceed from a qualitative standpoint and recommend avoiding leaving sharp angles in individual cells. One technique is by relaxing individual cells with small, nearly degenerate face areas by moving vertices while maintaining mesh relationships; another is through splitting cells into a number of smaller cells with more regular shapes, e.g. in (Blacker, 2000). Other techniques include mesh refinement, in which individual cells are broken up into smaller cells with more suitable geometries. This is an area of on-going progress by authors such as (Gaffney, Hassan and Salas, 1987) or, more recently, (Staten *et al.*, 2009). However, these authors tend to study applications made from the standpoint of engineering problems concerned with man-made objects such as aerial vehicles (Aftosmis *et al.*, 1998). These contain multiple individual components (e.g. fuselage, wings ...) that often present smooth surfaces. Natural orography is less geometrically complex since there are no components to be assembled, but on the other hand the individual components of man-made vehicles lack the sharp, irregular variations present in mountain terrain. It is for this reason that classical mesh-creation techniques such as quad- and oct-trees (Yerry and Shepard, 1983, 1984; Shepard and Georges, 1991), the advanced-front approach (Lohner *et al.*, 1988; Lo, 1991), paving and plastering (Canann, 1991; Blacker and Myers, 1993) and whisker-weaving (Tautges *et al.*, 1996) are pertinent for what are essentially simple man-made shapes associated by spatial relationships, but respond less well to meshing the air volumes above mountain terrain.

In the case of meshing air volumes above mountain terrain, the particularities of the initial data model may be used to create a structured mesh, that is then successively refined using the quality measurement as a criterion of convergence. However, it must be ensured that the Delaunay criterion is respected in the final mesh, which can be maintained if the following suppositions are verified:

1. The criterion is respected for the initial mesh.

2. The position of a single vertex is modified at each iteration of the smoothing algorithm.

3. Ensuing mesh quality is re-calculated: if vertex displacement has made overall mesh quality better, the new position of the vertex is maintained. Otherwise, a different vertex is chosen.

3.3.1 Creating an initial mesh

The 3D mesh used as a basis for calculations is a mathematical description of the volume of air situated above the orographical feature to be considered.

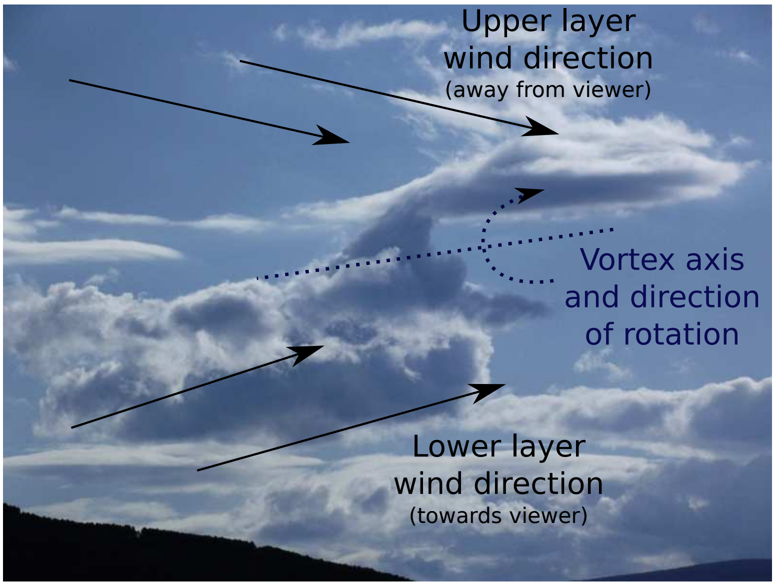


Figure 3.7: Cloud formation showing a large vertical structure. A vortex with horizontal axis is observed, due to the interaction of two horizontal layers with differing wind directions.

The volume being meshed must have sufficient vertical free space to represent correctly fluid structures formed (turbulent vortexes) at all scales of formation. Observation gives an indication of the largest scale structures that may be expected (Figure 3.7).

As pointed out in (A. Ward and J. Jorba, 2011), Digital Elevation Model data is generally available in the form of a square grid with regular horizontal separation l along both X and Y axis. Each grid data item consists in a vertical coordinate along the Z axis, whose value corresponds to the terrain altitude at that horizontal location. The DEM thus defines a the shape of the terrain, as a 2D object mapped into 3D space.

Altitudes above sea level are noted $h_{i,j}$. The lower boundary of the mesh is the terrain itself. Let $h_{bottom} = \min(h_{i,j,k})$. Since the volume of interest

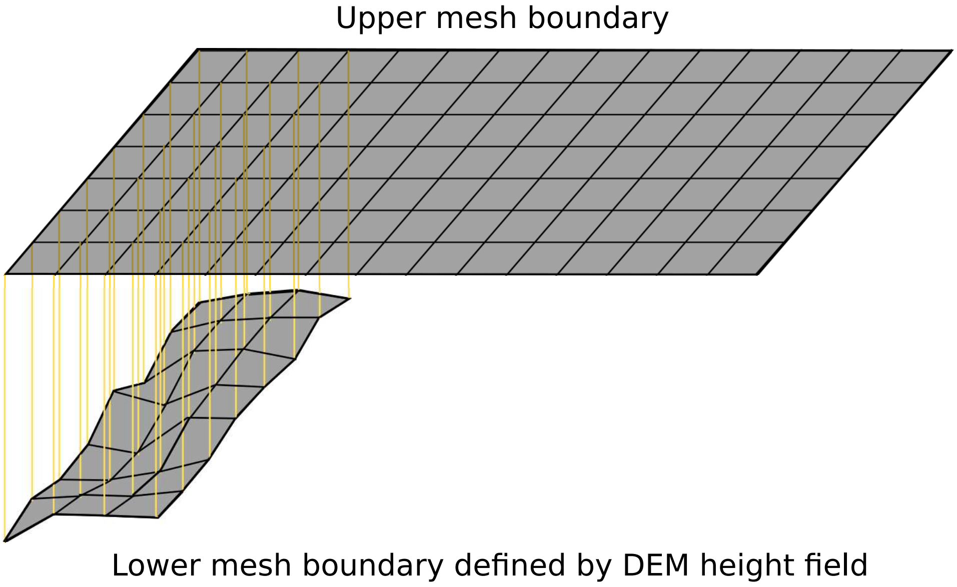


Figure 3.8: The 3D mesh defined at its lower bound by a DEM description of terrain heights, and at the upper by a fixed value.

may not always start at sea level, h_{bottom} may correspond to an altitude of e.g. 800 m.

A flat upper boundary h_{upper} is chosen for the mesh in such a manner as to leave sufficient vertical space above the highest points in the terrain (Figure 3.8.

Then, $(n-1)$ horizontal mesh layers are formed by vertical sweeping. Since the mesh is square, mesh points may be given a coordinate system with (i, j) horizontal indexes, e.g. i in the East-West direction and j North-South, and k a vertical index [2]. Mesh points have coordinates $x_{i,j,k}$ If the mesh is chosen with original mesh points (before optimization) regularly placed on an orthogonal grid with horizontal separation l along both X- and Y-axes, mesh point coordinates are simplified giving:

$$\left| \begin{array}{l} x_{i,j,k} = x_{i,j} = l.i \\ y_{i,j,k} = y_{i,j} = l.j \\ z_{i,j,k} = h_{i,j} + (h_{upper} - h_{i,j}).\frac{k}{n} \end{array} \right| \tag{3.2}$$

[2] Although there is no specific requirement for the mesh to be aligned along the cardinal directions, this disposition is the easiest when the underlying DEM data is given with such a disposition.

- X-axis horizontal coordinate x depends only on the first horizontal index i.
- Y-axis horizontal coordinate y depends only on the second horizontal index i.
- Vertical coordinate z depends only on the vertical index (grid layer index) k.

This initial mesh can be built to respect the Delaunay criterion. In Figure 3.9, a worst-case scenario is shown, in which the circumsphere of a hexahedron is displaced laterally in such a manner that its center is located on one of the lateral faces. If the horizontal distance l separating vertical mesh lines is smaller than circumsphere radius r, mesh points on adjacent vertical lines at x_{i-1} might be included inside the sphere.

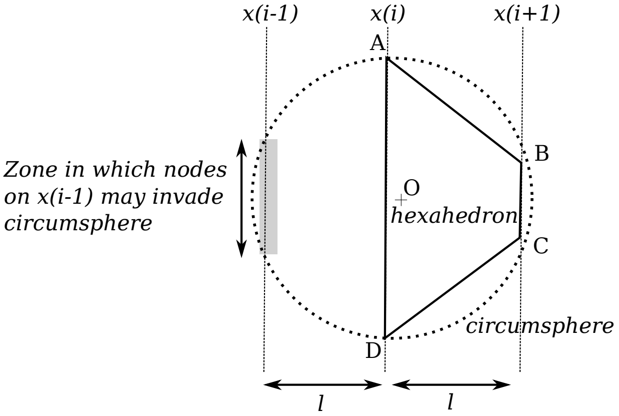


Figure 3.9: Worst-case circumsphere around a hexahedron during initial mesh creation. 2D projection.

To ensure this cannot happen and Delaunay's criterion is respected during initial mesh generation, it must be ensured that the distance between point O, center of this hexahedron's circumsphere, and the vertical mesh grid line at x_{i-1} is less than r, radius of the circumsphere.

A worst-case scenario is achieved when O is as far along the negative X axis as possible, i.e. when points A and D are at extreme separation, and B and C are practically united on AD's median axis (Figure 3.10).

Posing $\delta h_{max} = \max(h_{i,j,k+1} - h_{i,j,k})$, in the worst-case situation we have AD $= \delta h_{max}$ and obtain the system:

$$\begin{cases} e + r = l \\ r^2 = \left(\frac{\delta h_{max}}{2}\right)^2 + e^2 \end{cases} \tag{3.3}$$

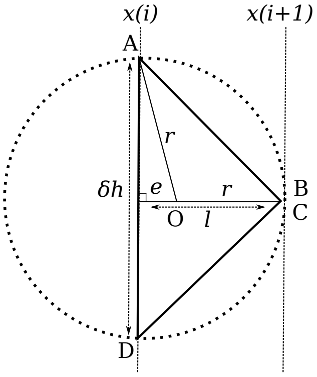


Figure 3.10: Close-up view of worst-case circumsphere.

e is eliminated to give:

$$r^2 = \left(\frac{\delta h_{max}}{2}\right)^2 + r^2 + l^2 - 2rl \tag{3.4}$$

$$2rl = \left(\frac{\delta h_{max}}{2}\right)^2 + l^2 \tag{3.5}$$

However:

$$r < l \tag{3.6}$$

Therefore:

$$\left(\frac{\delta h_{max}}{2}\right)^2 + l^2 = 2rl < 2l^2 \tag{3.7}$$

$$\left(\frac{\delta h_{max}}{2}\right)^2 < l^2 \tag{3.8}$$

And finally, the condition is that:

$$l > \frac{\delta h_{max}}{2} \tag{3.9}$$

On the other hand, using n the number of horizontal layers, and h_{upper} and h_{lower} the respective upper and lower bounds of the mesh in altitude:

$$\delta h_{max} >= \frac{h_{upper} - h_{lower}}{n} \tag{3.10}$$

Thus a sufficient condition can be:

$$l > \frac{h_{upper} - h_{lower}}{2n} \tag{3.11}$$

Which, by rearrangement, gives a final sufficient condition that ensures that the initial mesh respects the criterion:

> The initial hexahedral mesh generated shall respect the Delaunay criterion, no vertex will find itself within the circumcircle of any triangle not containing that vertex, and individual hexahedra geometry will be as regular as possible thus facilitating numerical calculation precision if a sufficient number n of horizontal layers is used, with:
>
> $$n > \frac{h_{upper} - h_{lower}}{2l} \tag{3.12}$$

3.3.2 Refining the mesh

Searching for a minimum of the potential function can be performed by any appropriate method. In this chapter, some of the more common algorithms for this purpose have been compared: Steepest Descent Hill-Climbing (SDHC, (Haskell B. Curry, 1944)), Conjugate-Gradient (CG, (Hestenes and Stiefel, 1952; Fletcher and Powell, 1963) and many others), Simulated Annealing (SA, (Kirkpatrick *et al.*, 1983)) and Genetic Algorithms (GA, (John H. Holland, 1975)).

Other techniques, such as Laplacian smoothing, were considered but discarded because of the specific nature of the terrain surface: sharp convexities around mountain peaks and ridges make the use of Laplacian smoothing difficult, since it "*can result in distorted or even inverted elements near concavities in the model*" (Cannan *et al.*, 1998) without the adjunction of other methods. Initial experimentation showed that methods such as constrained Laplacian smoothing (Parthasarathy and Kodiyalam, 1991) does not propagate quickly from the level of the terrain upwards into the interior of the volume.

Whatever the algorithm, for each mesh cell certain of the hexahedron's vertices shall define the circumsphere and be located on its border. Others shall be entirely within the sphere. No general suppositions can be made as to the positions of the maximum and minimum interior angles $\max(\alpha_i)$ and $\min(\alpha_i)$.

If the smoothing algorithm alters the position of one of the vertices within the circumsphere, the radius and center of the sphere will not in itself be modified - at least, until the vertex reaches the border of the sphere. We can thus be sure no vertices belonging to other hexahedra will be absorbed into this circumsphere, and the Delaunay criterion is still respected.

On the other hand, if the smoothing algorithm alters the position of one of the vertices on the border of the circumsphere, this action may well alter the shape of the sphere itself. The hexahedron may be in one of several situations:

- If the vertex whose position is altered is that corresponding to minimum solid angle $\min(\alpha_i)$, the algorithm will tend to decrease the quality measurement towards 0, or in other terms *increase* the value of $\min(\alpha_i)$. This "pushes" the vertex towards the center of the circumsphere, and the new circumsphere shall in any case be included within the shape of the older one. In this way it is assured that external vertices may be absorbed within this contracting circumsphere.

- If the vertex altered corresponds to maximum solid angle $\max(\alpha_i)$, the situation is reversed: the algorithm's decreasing the quality measurement towards 0 implies *decreasing* the value of the solid angle, or "pulling" the vertex further away from the center of the existing sphere. This may conceivably stretch the circumsphere outwards to absorb vertices belonging to adjoining hexahedra.

- If the vertex the position of which is altered corresponds neither to $\min(\alpha_i)$ nor $\max(\alpha_i)$ but is adjoining the vertex corresponding to these solid angles, the situation is more complex and may not be decided by reasoning alone.

From the above discussion, a supplementary sufficient condition for each iteration of the smoothing algorithm can be derived:

> Any smoothing algorithm will maintain a hexahedral mesh that originally respects the Delaunay criterion within this state if it only alters the position of vertices within the circumsphere of each hexahedron, or those on the circumsphere that correspond to the minimum solid angle thereof.

Smoothing algorithms have been implemented in such a way as to ensure this condition is met; otherwise, the vertex displacement has been taken back an the vertex replaced in its original position.

3.4 Experimental results

The smoothing algorithms chosen have been applied to a square horizontal domain around and above the ski slopes at Arcalís (Andorra), 4416 x 4416 m in size (Figure 3.11). Terrain altitudes range from 1400 to 3000 m above sea level, so $h_{bottom} = 1400$ m while the top of the mesh was taken at $h_{upper} = 4000$ m giving a minimum vertical displacement of $h_{upper} - max(h_{i,j}) = 1000$ m to ensure large fluid structures are allowed to develop within the computer model.

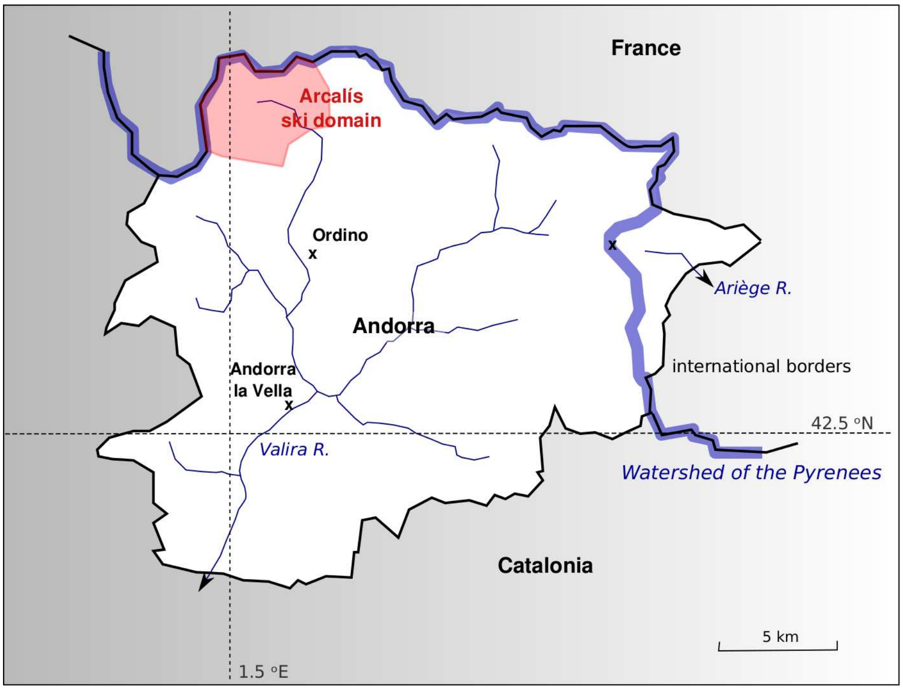


Figure 3.11: Geographical situation of the Arcalís ski slopes, Parrish of Ordino, Principality of Andorra.

Geographical formations include several high altitude glacier cirques, with associated ridges that form the watershed of the Pyrenean range. The upper part of a glacier-formed valley starts out in a southward direction from the south-eastern quadrant of the domain. These formations give rise to both high values of slope (up to 75° in places) and rates of slope variation (from 15° to 75° within a 100 m horizontal distance).

A digital elevation model has been created from data made available online by the Sigma server (Sigma map server, 2009), based on data originally drawn up by the Andorran Government. Meshes of increasing sizes, with 10x10x6, 20x20x12, 40x40x24 and 80x80x48 cells, were built and refined using the methods set out in the section above, continuing iteration in each case until the progress of quality became inferior to 10^{-6} per iteration.

Mesh smoothing was performed in serial and parallel fashion on various computational 64-bit test-beds, with Central Processing Units (CPU) ranging from single-core single-thread Mobile AMD Sempron model 3500+ to nodes with dual quad-core HT Intel Xeon model E5440. Parallel processing was enabled using a shared-memory paradigm and lightweight POSIX threads (IEEE Threads, 1992). Code was executed on one dual quad-core Xeon mode E5440 (at 2.53 GHz) computation node in order to avoid accessing remote memory across an external network fabric. Using only local memory with no local or remote disk access, the computational overhead associated with implementing a message-passing scheme such as MPI was avoided, which was appropriate since each processing thread required access to the complete data set.

With each technique, mesh quality was evaluated at each iteration of the each smoothing algorithm as set out in the previous section: the quality of each individual mesh hexahedron was evaluated and the results added over the complete mesh thus giving a dimensionless measure of mesh quality.

After 25 iterations of each smoothing algorithm, the results given in Table 3.1 were obtained. The different methods of smoothing were applied on an identical mesh sized 10x10x6. Initial mesh quality was 81.11 for the complete mesh in all cases, with a 0.135 average hexahedron quality.

The simplest algorithm used, Steepest Descent Hill-Climbing (SDHC), is more computationally expensive than its simple nature would lead one to expect. The reason is that the gradient

$$\frac{\delta q}{\delta p_i}, i \in \{1, 2, 3\} \tag{3.13}$$

must be calculated for each position of each node n_i. It also shows an inevitable tendency to overshoot optimal positions for each node. Though methods may be used to correct this – both a polynomial evaluation of the gradient, and trial-and-error displacements of the node were considered – , they also require several additional evaluations of q for each node and iteration. This is also the case for third-order variants of Newton's method such as those discussed in (Babajee and Dauhoo, 2006).

Derivatives of SDHC such as Conjugate Gradient (CG) methods also display the same drawback, with additional requirements in terms of memory space. For these methods, intermediate positions must be maintained for each and every node in the mesh. This is not optimal when the parallel implementation of this algorithm is run on a single node, and the complete data set (mesh node positions) must fit within a single RAM space.

Table 3.1: Results of application of different methods of smoothing.

Method	Time/ iteration (s)	Final quality	Gain in quality(s)	Time/gain in quality
Steepest Descent (SDHC)	9.15	76.27	4.84	49.17
Random search	4.13	74.23	6.88	16.32
Simulated Annealing (SA)	4.07	66.25	14.86	7.44
Genetic Algorithm (GA)	21.77	81.11	$< 10^{-4}$	$> 6.5 \cdot 10^{6}$

This contrasts with the relatively good results of a random search, which consists in considering a random displacement for each node within adjacent hexahedra. Quality change is calculated and the displacement is validated only if quality progresses. One of the reasons for the good results of this method in terms of computation time is that it is only necessary to evaluate the change of quality of adjacent hexahedra, not of the entire mesh.

Simulated Annealing (SA) was also evaluated, using an exponential temperature function that started at an initial value corresponding to average cell width:

$$T_{iter} = T_0 \cdot exp^{-\frac{iter}{\tau}} \tag{3.14}$$

The values tabulated above correspond to a value of $\tau = 10$. Though some authors such as (Alrefaei and Diabat, 2009) suggest the use of constant temperature SA for multi-objective optimization, it has not given good results for our specific problem.

Both pure Genetic Algorithms (GA) and Idealized GA (IGA, (Mitchell *et al.*, 1993)) were not initially seen as an option, due to the large amount of memory space needed to represent individual population chromosomes: each chromosome in the population would need to represent the position of all nodes in the mesh. Experimentation confirmed this fact. At the same time, large numbers ($\approx$ 1000) of new chromosomes are generated through cross-breeding and mutation at each iteration and their fitness (i.e. quality) must be calculated. This slows down considerably the computation of each iteration. On the other hand, while mean population quality does increase, the quality of optimum chromosomes does in general not progress satisfactorily.

Since the Simulated Annealing (SA) technique consistently gave better progression per unit of time, it was chosen as a basis to evaluate the quality of the final meshes produced. The following discussion shall be based on the results this technique. Other techniques gave generally similar quality albeit at a higher cost.

Visual inspection showed that mesh quality of the initial meshes was sub-optimum mostly in the region of the watershed and other sharp peaks. The geological nature of the terrain does not lend itself to sharply concave structures, and so little work was needed at valley bottoms and inside cirque bowls. Near these regions, hexahedra quality was in general good, with measures of quality for individual hexahedra < 0.2 (so nearly cubical in form). However, in some cases the presence of nearby peaks reduced mesh quality at intermediate altitudes above relatively smooth terrain (Figure 3.12).

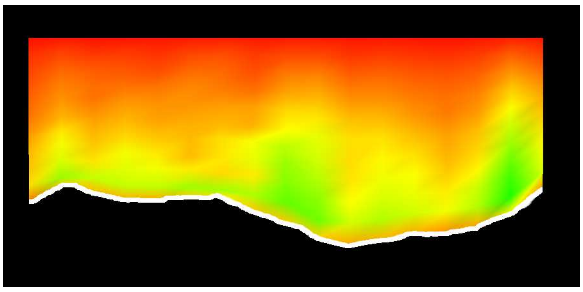


Figure 3.12: Sample cut through a smoothed mesh. Mesh deformation is color-coded from red (hexahedron quality $= 0$, indicating no deformation) to green (hexahedron $quality \approx 0.5$). Completely deformed hexahedra (quality measurement $= 1$) do not appear in the test case.

Computation time per individual iteration of smoothing was found to scale linearly to the number of mesh nodes. This is coherent with the fact that work performed on each node is independent of mesh overall size. Closer inspection using program profiling showed that up to 99.8% of CPU time used in pure algorithm execution was actually spent evaluating hexahedral quality.

On all mesh scales studied, the amount of quality gained at each iteration was measured. In Figure 3.13, the horizontal axis is the number of iterations of the mesh smoothing algorithm using Simulated Annealing, while the vertical axis is the total of the quality function applied to all hexahedra in the mesh: lower values indicate more regular mesh elements. The evolution of the total mesh quality correlates well with a linear function. A decreasing gain would have been considered reasonable, since very deformed hexahedra may be optimized more than more regular ones. The gain in mesh quality can thus propagate in larger amounts at the beginning of the process, while as the algorithm progresses, each individual hexahedron's quality is increased in lesser amounts at each iteration. However, this is not what is observed in practice.

Quality evolution was correlated with various functions. Noting δq the amount of quality gained over the complete mesh and x the number of algorithm iterations, best fitting was obtained for a law of the form $\delta q \propto x^{-1.4}$ with a Pearson's correlation coefficient value of $r^2 = 0.52$, and for a linear law $\delta q \approx q_0 - a.x$ with $r^2 = 0.99$. It is for the time being unclear why quality evolution should correlate best with a linear function.

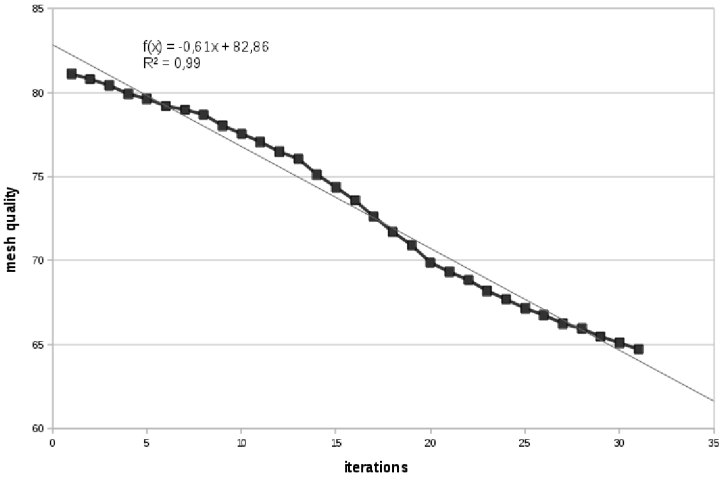


Figure 3.13: Mesh quality evolution per iteration step (dots). Linear trend function (line).

As the number of mesh divisions is increased in size (successively: 10x10x6, 20x20x12, 40x40x24 and 80x80x48 cells), the increasing number of hexahedra gives the mesh better resolution of terrain irregularities. Peaks and creases that are not distinguishable in a 10x10 horizontal mesh appear clearly at 80x80. The unfortunate effect on mesh quality is that additional hexahedra are created either near regular sections of terrain, or near sharp creases. Those near creases are created deformed in a larger extent, and thus of less quality. Thus, while increasing the number of mesh divisions gives a better representation of the reality of the terrain, in actual fact average hexahedral quality diminishes to some extent.

However, in all cases the process of smoothing allows producing a mesh of decreasing deformation measurement, and thus increasing quality, than originally obtained. Over all tests performed, the deformation measured decreased from 0.253 to 0.197 . It should be stressed that this is an average value that includes not only much deformed hexahedra close to the terrain, but also very regular hexahedra at high altitudes.

Analysis of hexahedral quality in function of altitude shows that hexahedra at lower and upper bounds of the domain are not greatly affected by the process of smoothing and that most progress is achieved at intermedi-

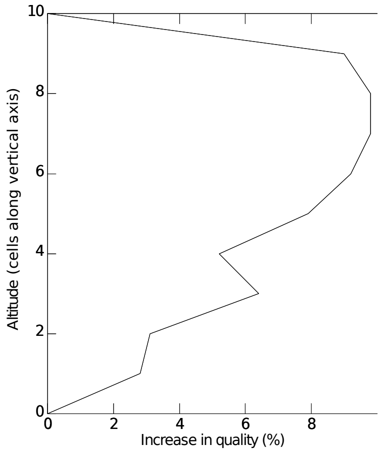


Figure 3.14: Average increase in quality at different heights above the terrain.

ate levels (Figure 3.14). This is consistent, since boundary mesh nodes are not moved, and satisfactory from a physical standpoint, since these are the volumes that shall hold the brunt of large turbulent structures - those with most kinetic energy.

3.5 Chapter conclusions

In this chapter, the physical domain under study has been prepared for mathematical modeling through volume discretization. A measure of mesh quality has been proposed. Due to the characteristics of the complex orography studied in this work, various methods of optimizing the resulting mesh have been compared. Simulated Annealing is proposed as the most promising algorithm to optimize such meshes, though results are not ideal. It has been noted that most optimization can be performed within the middle layers of the mesh, while high and low altitude sections near the mesh boundary have fixed forms that allow little optimization.

The process of establishing the domain discretization permits establishing the connection between the physical quantities to be modeled (air density and flow speed), and the shape of the domain. This shall be used in the next chapter, in which computational means of solving the Navier-Stokes equations for continuous flows are discussed, specifically how existing CFD tool-kits can be adapted to solve the systems considered in this work.

Chapter 4

Solving the Navier-Stokes equations

In the previous chapter, the physical domain has been discretized using a mesh. The peculiarity of mountain terrain is the convoluted shape of the lower boundary of the volume to be meshed. This induces individual cell hexahedral elements to assume shapes with low-quality as measured using the technique proposed. Several techniques for optimizing the mesh structure have been proposed.

Now the mesh structure has been optimized, in this chapter the application of computational techniques to this class of fluid mechanics problem is presented. OpenFOAM (OpenFOAM Foundation, 2011) solvers shall be presented in order to implementing solutions to the Navier-Stokes equations for conservation of mass and momentum. The choice of OpenFOAM software and their structure shall be discussed.

Pursuant to **Goal 2** of this thesis, mesh decomposition strategies are studied with a view to enabling constructing computer models of the area under consideration at successively smaller scales: from regional to local. The parallel application of the solvers to the problem of modeling airflow over complex terrain shall then be planned taking into account the particularities of mountain terrain.

4.1 Computational methods used in Fluid Mechanics

4.1.1 The choice of computational method

Experimental techniques to study problems in Fluid Mechanics have been used for many years. The use of wind tunnels to model airflow around aircraft and other objects in motion relative to wind has been documented since Francis H. Wenham's work in the 1870's (Russel Naughton, 1999). Modern aircraft such as the Airbus S.A.S. model A380 use such physical experimentation techniques alongside computational models to certify aircraft characteristics before building (Airbus S.A.S., 2014).

However, large objects must be reduced in scale due to physical wind tunnel constraints, making the observation of wind flow details more difficult as the size of the domain to be studied increases. At the same time, when modeling airflow above mountain terrain, experiments that take place within the laboratory may have difficulties representing parameters that vary both spatially and temporally during the experiment, such as terrain roughness. To take an example, when modeling snowfall, the falling snow will tend to fill surface irregularities, thus potentially altering air flow characteristics within the surface layer. In these cases, computational models of the fluid system associated with increasing memory capacities of modern machines both allow precise simulation of air flow details, and permit more flexibility of physical parameters that may be altered at will during the simulation process. These modifications made be made either in general, or in a localized fashion (for example, depending on altitude).

Building a computer model for a fluid mechanics problem in performed several stages (Ferziger and N. Perić, 2002):

1. A mathematical model is drawn up. In some cases, all three Navier-Stokes equations must be solved, such as when thermodynamical interaction and or exchange between system components is to be foreseen. In others, certain simplifications may be applied, as discussed previously in Chapter 2.

2. A discretization method is chosen. This is necessary to handle the difference between a physical problem that occurs within a continuous spatial and temporal referential, and a computer representation that must be described in terms of numerical quantities.

   Such methods may be broadly classified as finite element (FE) methods, finite volume (FV) and finite difference (FD).

In the finite volume class used in this study, the fluid values at discrete points within the domain are taken into account, and the small volume of fluid surrounding each point. The Navier-Stokes equations are written as surface integrals using Gauss' divergence theorem as seen in Section 2.1.1. The resulting system is represented in matrix form and solved using computational techniques as described in Chapter 3.

For comparison, the finite element class of methods the fluid domain is separated into individual sub-domains, or elements, represented by a 2D- or 3D-mesh. The Navier-Stokes equations are written for each element, and variational methods used to minimize differences between values at the facets between elements.

As for finite difference methods, finite difference equations are set up over the mesh. A function representing the solution is supposed to be "well-behaved" (i.e. differentiable several times) over the domain. A first-, second- or third degree spatial derivative is then approximated for each mesh point using discrete fluid values and the Taylor Series expansion. Equations for all points are combined into a matrix and solved.

Finite volume methods are known to be intrinsically conservative, maintaining total fluid mass and other constants over the mesh during calculations. For this reason, they may be preferred to other techniques when the computational model must represent not only a state of the system at a single point in time, but also an evolution over a certain period.

Once the computational package is set up, actual fluid model simulation may take place for a single time point, or over a period. Finally, simulation results must be extracted from the model and post-processing performed such as 3D viewing:

3. Perform computer fluid dynamics model simulation routine.

4. Results post-processing.

The complete computer simulation cycle may thus be represented in the manner set out in Figure 4.1.

4.1.2 Open- and closed-source software

When choosing software for research purposes in general, it may be advantageous to take into account the open or closed nature of each program.

- **Open-source** software projects publish their source code, which is then readily available to other users. From a computer science's point of view, this means the program may be verified as to correctness, and

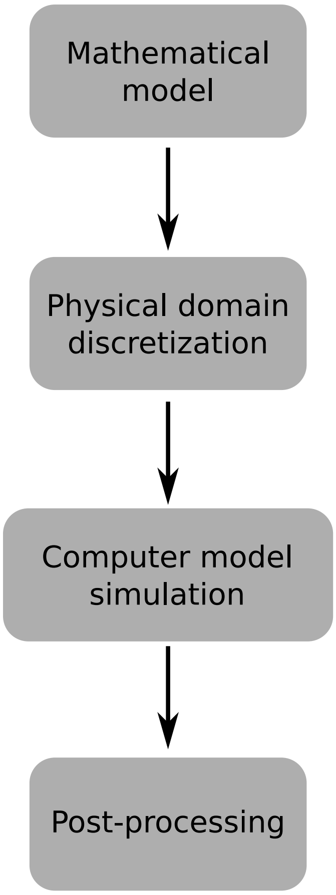


Figure 4.1: Computer fluid dynamics model schema.

also easily modified, for example to run on other -perhaps previously unavailable- hardware platforms.

It must be stressed that although most open-source software is released free of charge, this is totally compatible with the fact that some open-source applications may be developed for a fee, should there be persons or institutions willing to pay for this service.

- In **closed-source** software projects, the source code of the final product is known only to its original developers, and remains carefully hidden from public view. This is a natural tendency in the field of commercial programming, where the code has economic value and may be considered an intangible asset of its owners.

  However, it also makes users dependent on the owners' good will when changes in the program need to made to adapt it to specific uses, or to different hardware platforms. It also makes independent assessment of the correction of the code next to impossible, both from a programming standpoint and from the point of view of fitness for a certain purpose.

Within closed-source software, a further distinction may be made between commercial software and freeware. In this latter category fall applications that remain with their source code non-accessible, but that may be used free of charge for all purposes, or for specific and limited purposes such as education.

From the standpoint of a researcher in other fields, the open-source nature of a software package is also beneficial sincs it enables the investigator to:

1. Easily adapt the existing software tools to the specific nature of the problem considered, since the source code of each part of the CFD package is made available.

2. Verify correctness of key elements. In essence, exposing the code base to public view permits continuous peer-review and criticism, thus granting knowledge of possible shortcomings in the code and making corrections possible (as well as probable is the software is much used).

4.1.3 Classification of software packages

Very many software packages have been constructed for the purpose of performing computer fluid dynamics calculations. Several criteria may be used for their classification.

In the first place, some software packages have been built for a specific purpose, while others are aimed at a more general objective. Specific bespoke codes require significant investment both of time and resources to obtain a similar level of precision and performance as more readily available existing codes (Anthony F. Molland *et al.*, 2011) (p. 176).

Among existing off-the-shelf software packages, a further distinction may be made between commercial, freeware and open-source programs. Commercial software such as FLUENT (originally developed by Fluent Inc., now part of the ANSYS portfolio (Ansys FLUENT, 2014)) have been used in scientific applications as well as industrial applications (Figure 4.2). More recent options such as COMSOL (COMSOL Multiphysics, 2015) are cross-platform for Windows, Linux and Apple operating systems. However, the closed nature of this code makes new module development and insertion a problem for investigators with non-standard needs.

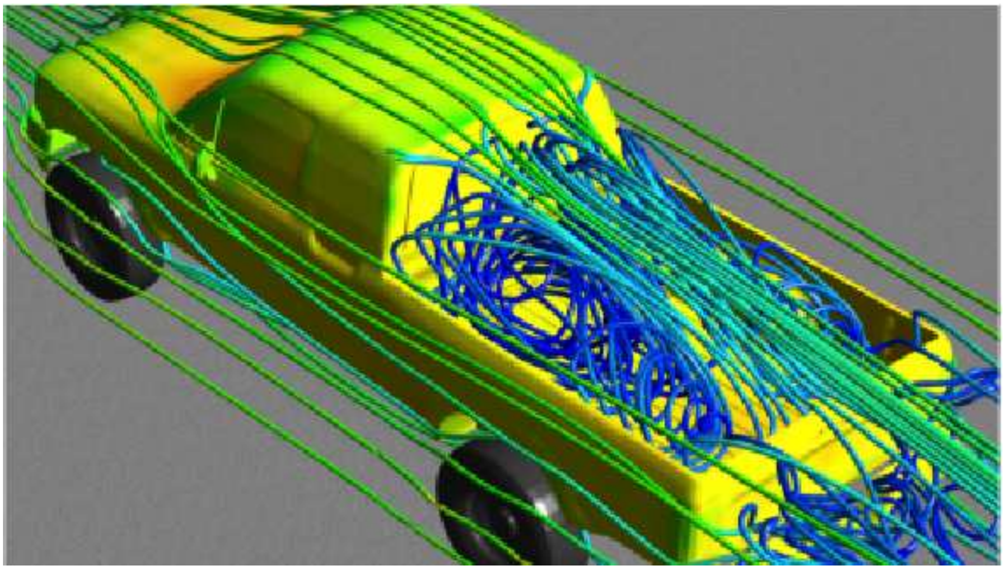


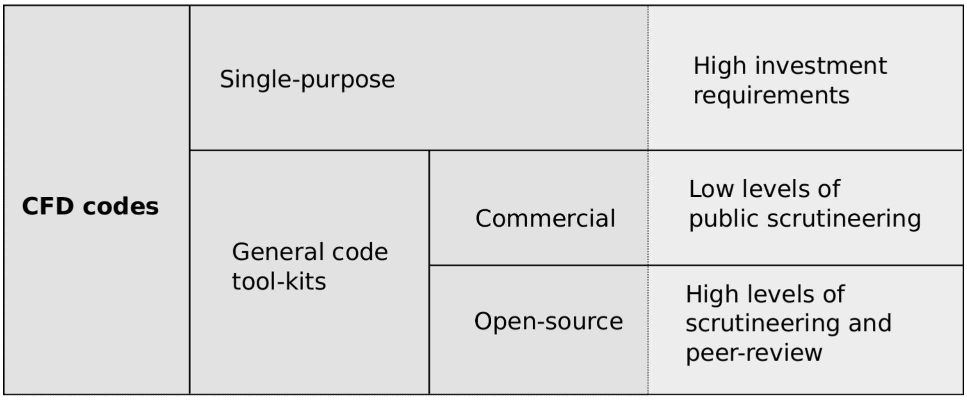
Figure 4.2: Application of FLUENT in an industrial application. Source: (CAE Associates Inc., 2014)

Open-source codes have also become available, such as Code Saturne (Code Saturne, 2015) and OpenFOAM. Although Code Saturne vas developed originally by Électricité de France (EDF), thus in an industrial environment, the source code has later been made open. It is known for its integration into Code Aster, a structural analysis code also used in the first place for the needs of this company.

OpenFOAM originated at the Imperial College (London, England) in the late 1990's. Based on the use of C++ object-oriented code, it differed from existing CFD software such as commercial toolkits that at the time were almost always written in a non object-oriented fashion using the FORTRAN programming language. New techniques developed and presented in several PhD thesis (Hrvoje Jasak, 1996; Onno Ubbink, 1997; David Paul Hill, 1998) acted as foundation for the design of a modular, extensible toolkit that has later been applied to various areas of CFD modeling such as the settling of two-phase flows (Daniel Brennan, 2001) and combustion modeling (Luca Mangani, 2008).

This software toolkit plays a role similar to the GNU/Linux operating system and the Internet encyclopedia Wikipedia, is free and is used by thousands of people worldwide in both academic and industrial settings (Goong Chen *et al.*, 2014). Its open nature and modular structure as a toolkit enable individual researchers to apply the project's code to a specific problem domain with some adaptation.

The situation of various types of CFD codes is summed up in Figure 4.3.



CFD codes	Single-purpose		High investment requirements
	General code tool-kits	Commercial	Low levels of public scrutineering
		Open-source	High levels of scrutineering and peer-review

Figure 4.3: Classification of CFD codes.

Other, smaller open-source projects are also available, such as Coolfluid, Dolfyn, Kratos and many others. The proliferation of such initiatives is indicative of the interest of the CFD community in sharing ideas and methods. However, the small number of developers involved in these projects and changes that do occur in their personal and work situations can make the projects' continued existence a challenge.

4.2 Structure of the OpenFOAM solvers

The OpenFOAM toolkit consists of a set of applications that may be classified as:

- utility programs to prepare the mesh for modeling;
- solver applications that apply the various methods to perform CFD simulation;
- post-processing helper programs.

Performing an OpenFOAM model usually includes using one or more programs from each category. For example, the *blockMesh* utility could first be used to create the mesh from indications given by the investigator. Then the *icoFOAM* transient solver for incompressible, laminar flow of Newtonian fluids could be invoked to resolve the pressure and fluid velocity fields on the mesh. Finally, a helper program such as the *paraFOAM* script used to set up the results to automatically invoke the visualization software Paraview (Amy Henderson, Jim Ahrens and Charles Law, 2004) for on-screen representation.

4.2.1 The choice of an OpenFOAM solver

The solvers available within the OpenFOAM may be classified as:

- Special-purpose solvers that have been constructed over the years to address very specific categories of problem.
- General-purpose solvers, suitable for general solutions for fluid speed $\vec{u}$ and pressure p

The first category comprises solvers such as *potentialWaveFoam* that solves a wave equation, *EHDFoam* to solve for electrostatic fields, *reactingFoam* and *alternateReactingFoam* to solve flows in which chemical reactions such as combustion take place. The open nature makes extending the basic OpenFOAM structure to treat specialized fluids a viable proposition.

As for the first category, such generic solvers may be further classified according to their handling of compressible or incompressible flows, whether a turbulence model is implemented or not and the other assumptions that may be applied to the fluid to simplify the underlying mathematical model. For example, *simplescalarFoam* is a very simple solver that treats a steady-state case, assuming

1. An incompressible fluid.
2. A turbulent flow.
3. Transport of a scalar quantity by the fluid.

The *simplescalarFoam* solver was initially considered for our application, but rejected for several reasons:

- The underlying mathematical model implements a mass diffusion model, that is appropriate to the transport of scalar quantities that move equally in all directions through the underlying fluid. This is not the case of snow particles within the air/snow mix.
- At the time of publication of this solver (January 2009), little documentation was available on on the solver itself.
- Additionally, development of the solver seems to have stalled, with just two versions published (both in year 2009).

At the opposite end of the complexity scale, *buoyantBoussinesqPisoFoam* is a solver that is designed to handle flows with:

- Turbulence, modeled using either the Reynolds-Averaged Navier-Stokes (RANS) or Large Eddy Simulation (LES) schemes.
- Buoyancy, as the effects of the weight of fluid that induces pressure change along the vertical axis and natural convection.
- Incompressible fluids, in which the Boussinesq approximation may be applied.

This solver gives a complete treatment of such fluids, has been under heavy development for some time, and furthermore is well-documented in the OpenFOAM community documentation platform (OpenFOAM Wiki, 2014). However, complexity is introduced into the mathematical model behind this solver by the treatment of the conservation of energy (temperature equation) as well as conservation of mass and momentum.

For this reason, a conservative choice was operated and the *pisoFOAM* solver used throughout experimentation in the context of this study. This transient solver is designed as:

- Solving an incompressible fluid.
- Turbulence may be handled, using several different schemes.

The *pisoFOAM* solver implements the Pressure Implicit with Splitting of Operator (PISO) algorithm (R. I. Issa, 1986), known for its efficiency since very few corrector steps are required to obtain a solution at each time-step (OpenFOAM Wiki, 2014). It is thus considered a good choice for our application, since:

1. Relevant quantities such as fluid speed and pressure are modeled.
2. Calculations that are not necessary in our context are avoided (heat exchange).
3. Algorithm execution is performed in an efficient manner.
4. Transient phenomena may be modeled and studied, such as those induced by increased snow layers in contact with the terrain.

It should be noted that this choice need not be final. The standardized syntax of OpenFOAM cases described in the following sections is such that it is in fact trivial to substitute one solver for another. Thus, if the investigator should require the supplementary features offered by e.g. *buoyantBoussinesqPisoFoam*, substituting this solver for *pisoFOAM* requires adding the supplementary boundary and initial condition files in the case directory, and re-running the computer model. However, the additional computational requirements imply that execution times will be impacted negatively.

4.2.2 Executing a CFD case with OpenFOAM

The general OpenFOAM problem description (or "case") environment consists of a directory in which are placed subdirectories *constant/*, *system/* and several numbered subdirectories (Figure 4.4).

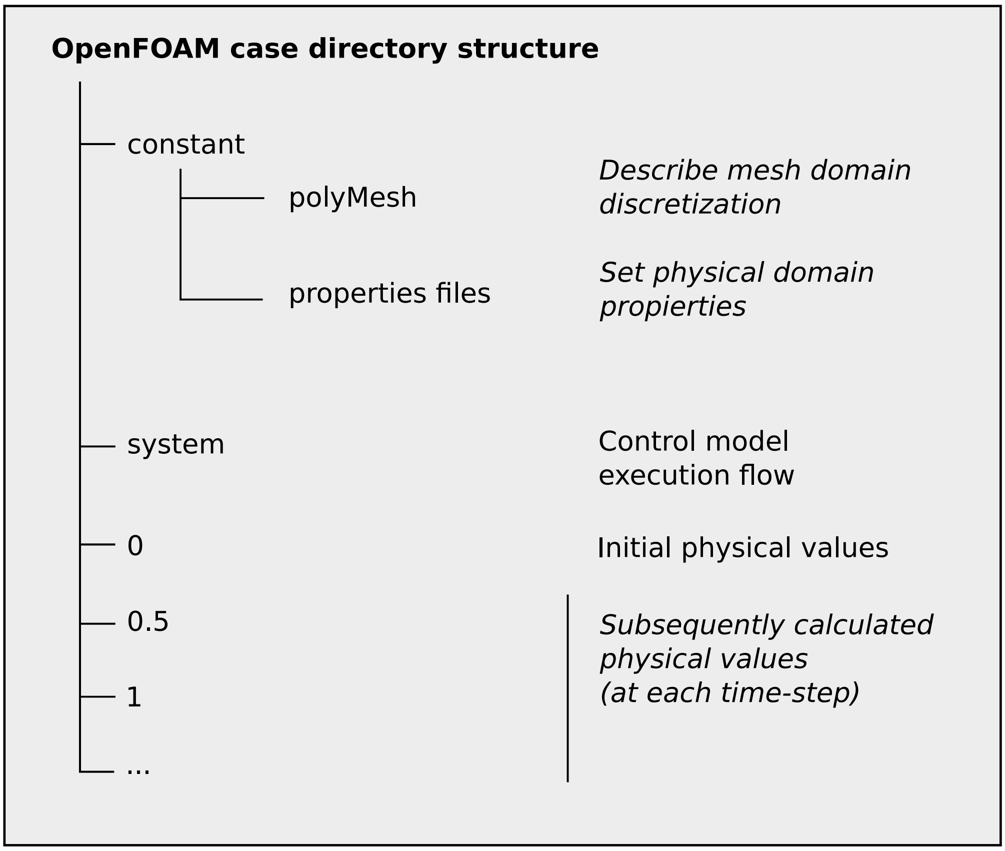

OpenFOAM case directory structure
constant
polyMesh
Describe mesh domain discretization
properties files
Set physical domain propierties
system
Control model execution flow
0
Initial physical values
0.5
1
...
Subsequently calculated physical values (at each time-step)

Figure 4.4: An OpenFOAM case directory structure.

Subdirectory *constant/* containing the description of system constant aspects. These include physical aspects such as mesh geometry in subdirectory *constant/polyMesh/*. Mesh is typically constructed from basic planar and volume elements, which are then meshed individually into smaller elementary volumes, and then adjacent volumes stitched together in a coherent fashion to produce the final mesh for the computer simulation model. The main file is *constant/polyMesh/blockMeshDict*, containing the basic elements. From this, separate files containing the definition of mesh points and faces are generated, as well as other files with meta-information concerning the relationships between elements: *constant/polyMesh/neighbour*, *constant/polyMesh/boundary* and *constant/polyMesh/owner*.

A simple *blockMeshDict* may could define a single basic volume using the following code:

```
convertToMeters 1;

vertices
(
    (-1 -1 -1)
    (1 -1 -1)
    (1 1 -1)
    (-1 1 -1)
    (-1 -1 1)
    (1 -1 1)
    (1 1 1)
    (-1 1 1)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (10 10 10) simpleGrading (1 1 1)
);
```

In this code, eight vertices are defined “vertices” using metric dimensions “convertToMeters”, and then combined together in a single hexahedral volume element “hex (...)”. It is then indicated that this basic element will be meshed using a grid of 10x10x10 individual cells “(10 10 10)”, distributed at equal intervals along all three spatial axes “simpleGrading”.

When meshed using the OpenFOAM *blockMesh* utility, the final mesh produced will contain $11^3 = 1331$ points and $3.11.10.10 = 3300$ distinct volume faces, and the corresponding relationship files will chart their structure relative to each other.

The *constant/* directory also contains files with physical constants to be used for model parametrization. For example, the Reynolds-Averaged Simulation (RAS) (Osborne Reynolds, 1895) consists of separating a physical field into its continuous (time-averaged) and fluctuating components. Applied to the Navier-Stokes equations, this is also known as Reynolds-Averaged Navier-Stokes or RANS turbulence model. To apply this model to the case in OpenFOAM, there will exist a file *constant/turbulenceProperties* containing the option setting:

```
simulationType  RASModel;
```

If this model is used, there must also exist a supplementary file *constant/RASproperties* giving further parameters for implementing RAS. For example, when using the $\kappa - \epsilon$ turbulence model (W. P. Jones and B. E. Launder, 1972), the *constant/RASproperties* file will contain the settings:

```
RASModel        kEpsilon;
turbulence      on;
```

Subdirectory *system/* served to direct model execution. File *system/-controlDict* indicates which solver application to use, and gives parameters relating to model timing such as start and end times, the time-step to be used, and how often model results should be written to disk. For example, when using the *pisoFOAM* solver, an appropriate *system/controlDict* file could contain:

```
application     pisoFoam;

startFrom       startTime;
startTime       0;

stopAt          endTime;
endTime         10;

deltaT          0.005;

writeControl    timeStep;
writeInterval   100;
```

File *system/fvSchemes* within the same directory allows the user to control which specific numerical schemes are used to implement mathematical operations. For example, the following extract instructs all solvers used to calculate this case to use a Gaussian linear scheme as default to calculate the gradient operator, and also specifically when it is to be applied to the pressure p and fluid speed $\vec{u}$ fields:

```
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
    grad(U)         Gauss linear;
}
```

Finally, in file *system/fvSolution* the method for solving systems may be chosen. In the following example, a pre-conditioned version of the iterative Bi-Conjugate Gradient method (PBiCG) is specified, with as preconditionner Diagonal-Incomplete Lower-Upper decomposition. The tolerance maximum limit for method residuals is given as a condition for iteration termination, while relative tolerance is not specified.

```
    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }
```

It can be noted that the OpenFOAM tool-kit is modular software environment, that uses a coherent programming model and a single file format with an identical syntax for all aspects of model description, thus simplifying extensions if necessary.

4.2.3 Specificities of the PISO solver

The *pisoFOAM* solver is available to implement the Pressure Implicit with Splitting of Operator (PISO) algorithm (R. I. Issa, 1986). This algorithm and derived methods (P. J. Oliveira and R. I. Issa, 2001) solve the combined mass- and momentum conservation Navier-Stokes equations efficiently. The algorithm's salient points are as follows (Hrvoje Jasak, 1996)[1]:

- The equation for conservation of momentum cited in section 2.1.4 is:

$$\rho\frac{d}{dt}(\vec{u}) + \rho\triangle(\vec{u}\otimes\vec{u}) = -\nabla p + \triangle(2\mu D - \frac{2}{3}\mu\delta_{i,j}\triangle\vec{u}) - \rho\cdot g \tag{4.1}$$

[1] Page 147 and following pages.

This equation is solved for the three components of fluid speed $\vec{u}$. External forces D and the gravitational effect (buoyancy) term $\rho.g$ are considered constant for this step. As for the gradient of pressure ∇p, data from the previous iteration of the algorithm is used initially.

- Using the new fluid speeds from the previous step, necessary corrections for pressure are calculated and the pressure field is adjusted.

- Conservation of fluxes between mesh elements and the domain border are introduced, allowing a second calculation of the velocity to be formulated and the final pressure field to be solved.

- If needed, other conservative quantities may be calculated.

From a computational standpoint, equations solving for fluid speed are handled in OpenFOAM using the Diagonal-Incomplete LU Bi-Conjugate Gradient method(DILUPBiCG) for asymmetric matrices (H.A. Van Der Vorst, 1992). This iterative method is applied three times in step 1 of the PISO algorithm, one for each spatial component of $\vec{u}$. Results are obtained when the Bi-Conjugate Gradient algorithm's residual is less than a specified value. In OpenFOAM, residuals less than 10^{-5} (as specified in file *system/fvSolution*) are typically obtained for meshes of 1331 cells in 1 or 2 iterations of the method.

On the other hand, solving for pressure is handled using the Diagonal-Incomplete Cholesky Conjugate Gradient method (DICPCG) for symmetric matrices (D.A.H. Jacobs, 1981, 1986). This method must be applied two times, in steps 2 and 3. In each application, a variable number of iterations is required to achieve satisfactory residuals.

Finally, when implementing the PISO algorithm to solve a mathematical model implementing a Reynolds-Averaged Simulation using the $\kappa - \epsilon$ turbulence model, the values of parameter fields κ and ϵ must also be updated, for which the DILUPBiCG method is also used.

An example of a complete iteration of the PISO implementation in OpenFOAM is as follows: fluid speed is solved along all three axis (Ux, Uy and Uz) using a single iteration of DILUPBiCG; then pressure p is solved using two iterations of DICPCG, and finally turbulence quantities ϵ and κ are solved using a single iteration of DILUPBiCG each.

```
Time = 0.14

Courant Number mean: 0.00952868 max: 0.588332
DILUPBiCG:  Solving for Ux, Initial residual = 0.00602228,
Final residual = 2.00996e-07, No Iterations 1
DILUPBiCG:  Solving for Uy, Initial residual = 0.0203194,
Final residual = 8.70494e-07, No Iterations 1
DILUPBiCG:  Solving for Uz, Initial residual = 0.00276547,
Final residual = 2.91961e-07, No Iterations 1
DICPCG:  Solving for p, Initial residual = 0.0551835,
Final residual = 0.00472294, No Iterations 18
time step continuity errors : sum local = 2.2839e-07,
global = 1.88137e-18, cumulative = 7.3806e-18
DICPCG:  Solving for p, Initial residual = 0.0323436,
Final residual = 9.22157e-07, No Iterations 155
time step continuity errors : sum local = 4.21735e-11,
global = 1.35803e-19, cumulative = 7.5164e-18
DILUPBiCG:  Solving for epsilon, Initial residual = 0.00893162,
Final residual = 5.78942e-07, No Iterations 1
DILUPBiCG:  Solving for k, Initial residual = 0.043843,
Final residual = 2.07392e-08, No Iterations 2
ExecutionTime = 53.26 s  ClockTime = 54 s
```

It can be observed that in this example that is typical of *pisoFOAM* execution, solving for fluid speed consumes one single iteration in all 3 applications of the DILUPBiCG solver, and solving for κ and ϵ consumes one each. On the other hand, solving for pressure consumes a total of 156 iterations of the DICPCG solver. Overall computer model efficiency clearly depends on the implementation of this solver, as confirmed in (Uli Göhner, 2010) where replacing existing OpenFOAM solvers (executed within the system's main CPU) were replaced by GPU-based solvers, reporting an over 75% performance gain. This is comparable to more recent offerings, such as SpeedIT FLOW (Vratis, 2015) that reports performance gains of 2.5x to 3.3x using an AMD K6000 GPU, in comparison to two Intel Xeon model E5649 CPUs.

4.3 Parallel strategies for solving the equations

As discussed in Section 3.1, solving the continuity equations numerically on a large mesh implies solving matrix systems with dimensions n x n, with n the number of volume elements in the mesh. For large meshes, the effort required to solve the system is considerable, leading to long execution times when executed on a single computing platform. For this reason, considerable interest has been placed in parallel execution of CFD codes in general, and specifically OpenFOAM.

The OpenFOAM toolkit contains parallel versions of solvers, that can use the OpenMPI implementation (Edgar Gabriel *et al.*, 2004) of the MPI interface (Marc Snir *et al.*, 1996) to communicate processes. The interface to parallel communications routines is situated at low-level within the toolkit. Most high-level routines such as solver code or user programming need not be aware of the presence or absence of a parallel computing environment, which simplifies writing generic code. For example, in source file *applications/-solvers/incompressible/pisoFoam/pisoFoam.C* the pisoFOAM solver defines the pressure corrector equation as follows:

```
fvScalarMatrix pEqn
(
    fvm::laplacian(rUA, p) == fvc::div(phi)
);
```

The equation is later solved using code:

```
pEqn.solve();
```

This is transparent for the final user, eliminating the need for considering different code bases for single-threaded and parallel execution of a model, which may be helpful when specific interests may or may not relate to parallel processing methods for mathematics in computer science.

4.3.1 Executing an OpenFOAM case in a parallel computing environment

Before solving the OpenFOAM case, some preparation has to be undertaken to prepare the case for parallel execution. The first step (OpenFOAM User Guide, 2014) is to decompose the mesh. Mesh decomposition involves separating the complete mesh into different segments, each segment representing a physical sub-domain that will be assigned to separate MPI execution tasks.

Many mesh decomposition schemes have been proposed, ranging from simple regular partitions to more complex algorithms that may be executed in parallel (Wu Poting and Elias N. Houstis, 1996). Recent developments include the detailed analysis of 3D object morphology and structure using fuzzy clustering (Sagi Katz and Ayellet Tal, 2003), curvature clustering (G. Lavoué, F. Dupont and A. Baskurt, 2005) or Reeb graphs (Stefano Berretti, Alberto Del Bimbo and Pietro Pala, 2009) *inter alia*, of use when a pre-existing volume on irregular shape must be meshed such as human or animal forms.

Optimal mesh decomposition should produce sub-domains with two properties (Wu Poting and Elias N. Houstis, 1996):

- balancing the quantity of computation necessary between nodes, producing sub-domains needing similar amounts of calculation time;
- minimizing bisection area between sub-domains, and thus communication between MPI tasks being executed on separate computation nodes.

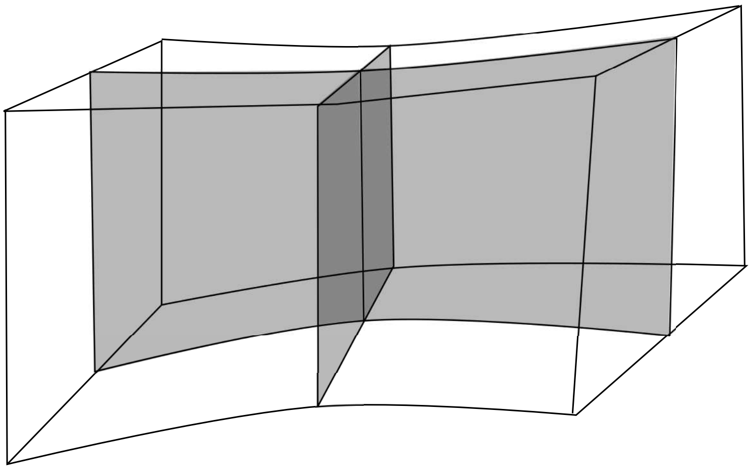


Figure 4.5: Domain simple decomposition into four sub-domains, with partitioning along the X- and Z- (horizontal) axis.

In OpenFOAM, three methods of domain decomposition are available. The first is simple decomposition. In this, the mesh will be partitioned in a regular fashion along each X-, Y- and Z-axis (Figure 4.5). The user indicates the number of domains -for example, “4”- and the partitioning along each axis -such as “(2 1 2)”- in file *system/decomposeParDict*. For example:

```
numberOfSubdomains 4;

method          simple;

simpleCoeffs
{
    n               ( 2 1 2 );
    delta           0.001;
}

hierarchicalCoeffs
{
    n               ( 1 1 1 );
    delta           0.001;
    order           xyz;
}
```

The *decomposeDict* utility is then run, that decomposes the mesh into sub-domains within separate sub-directories *processor0/*, *processor1/*, ...

The second method for domain decomposition is the METIS (Multi-level k-way Partitioning Scheme for Irregular Graphs) family of algorithms (George Karypis and Vipin Kumar, 1998). These multilevel bisection algorithms are implemented in the *metisDecomp* utility, and has been shown to execute quickly. However, it has also been reported as not being effective for fine partitions of the mesh, when the number of individual sub-domains is larger than 1024 (Jesús Antonio Izaguirre Paz, 1997).

```
metisCoeffs
{
    processorWeights ( 1 1 );
}
```

Finally, Scotch domain decomposition is also available (François Pellegrini and Jean Roman, 1996). This implementation of the Dual Recursive Bipartitionning algorithm allows the mapping of any weighted source graph (mesh cells) to be mapped onto any weighted target graph (compute node network topology). As such, it has been conceived to handle the mapping of meshes with complex topologies. This method has been implemented in OpenFOAM in the *scotchDecomp* utility.

Whatever the method chosen for domain decomposition, once this is performed the case execution itself may take place. However, instead of execution the solver (e.g. pisoFOAM) directly, it must be run from within the OpenMPI framework. For example, if the domain is decomposed into four different sectors, case execution could be performed using the command:

```
mpirun --hostfile machines -np 4 pisoFoam -parallel
```

File *machines* describes the MPI parallel processing nodes available on the computation cluster.

4.3.2 Elements contributing to parallel scalability

The scalability of parallel solvers may be defined as the possibility of increasing program execution speed as the number of compute nodes involved increases. The ideal case would be when linear scalability is achieved, when an N-fold increase of computing nodes would render a equally proportioned N-fold increase in execution speed, or correlatively an execution time reduced by N times.

Scalability is in general known to depend on several factors. The well-known Amdahl's Law (Gene M. Amdahl, 1967) separates the process to be executed into two separate components: W_s is the part of computation that must be performed as a single-threaded process, in a sequential manner, and W_p the part that may be performed in parallel. It is then postulated that maximum speedup S that may be expected when N compute nodes are used is

$$S = \frac{W_s + W_p}{W_s + \frac{W_p}{N}} \tag{4.2}$$

Since W_s and W_p are both in arbitrary work units, it can be taken that $W_s + W_p = 1$, thus giving

$$S = \frac{1}{\frac{1}{N} + W_s \cdot (1 - \frac{1}{N})} \tag{4.3}$$

$$S = \frac{N}{1 + W_s \cdot (N - 1)} \tag{4.4}$$

In this form of the expression, it is clear that speedup S will be maximized when the sequential part of the work flow is reduced to zero, at which time the ideal maximum speedup of $S = N$ may be achieved. Since in practice work flows all have a sequential segment, be it only to read the system initial state and to output results, this ideal maximum is not realistic in practice.

Gustafson (John L. Gustafson, 1988) interprets this law in different terms, noting that when increased computation power is available, the problem is generally expanded to make use of the increased facilities.

More recently, the availability of larger numbers of computing nodes in massively parallel computing environments has lead researchers to study the relationship carefully and conclude that splitting a work load into larger values N of individual tasks helps mitigate the effect of the sequential component W_s (Shi Yuan, 1996). However, as pointed out in (Mark D. Hill and Michael R. Marty, 2008), the appearance of multi-core processor chips complicates calculations since various levels of interconnection are used: internal to each chip for inter-core communication, within the motherboard between chips in a dual- or quad- CPU system, and over the network infrastructure between compute nodes when computation is parallelized using a cluster.

Analysis on a 16 CPU (AMD Opteron 280 at 2.4 GHz, dual core) compute test-bed in (Håkan Nilsson, 2007) conclude on an increase in execution speed that is nearly linear when two compute nodes are used, but degrades when the number of physical nodes is increased. This investigator also reports consistently better results when an Infiniband (PCI-X) interconnect is used instead of a Gigabit Ethernet, giving speed-ups of x7.2 using all 16 cores and Gigabit Ethernet, compared to x8.2 using 16 cores and Infiniband.

As for the specific solvers used on equation sets, much effort regarding the efficiency of system resolution with OpenFOAM has been placed on the Conjugate Gradient class of sparse matrix solvers. In (Orlando Rivera, Karl Fürlinger and Dieter Kranzlmüller, 2011), investigators conclude that the BiCG solver scales well over the number of compute nodes when compared to other options such as the Geometric Algebraic Multigrid (GAMG) solver (John Ruge and Klaus Stüben, 1984). Authors indicate that two particular MPI function calls, MPI_Recv and MPI_Allreduce, are not among those most invoked but do contribute to a large extent to performance issues.

Results of (Massimiliano Culpo, 2011) include a comparison of two different MPI implementations (OpenMPI and Intel MPI) and indicate that computational efficiency and execution times do not depend on the MPI implementation (we add: as long as the same interconnects and network libraries are used during comparison). The same author studied the Preconditioned Conjugate Gradient method used in the DICPCG solver, and also concluded that the MPI_Allreduce barrier invoked when calculating matrix scalar products acts as a limit to code scalability. Further analysis of specific points at which the MPI implementation reduces calculation efficiency may be possible using automatic parallel program analyzers, as pointed out in (Josep Jorba Esteve, 2006).

4.4 Application to complex mountain orography. Experimental results

Constructing a computer model of the airflow above a complex mountain orography from a Digital Elevation Model may be done starting from a simple hexahedral global mesh form. The top and lateral faces of the hexahedron assume a planar shape, while the lower face must follow the forms of surface terrain (Figure 4.6).

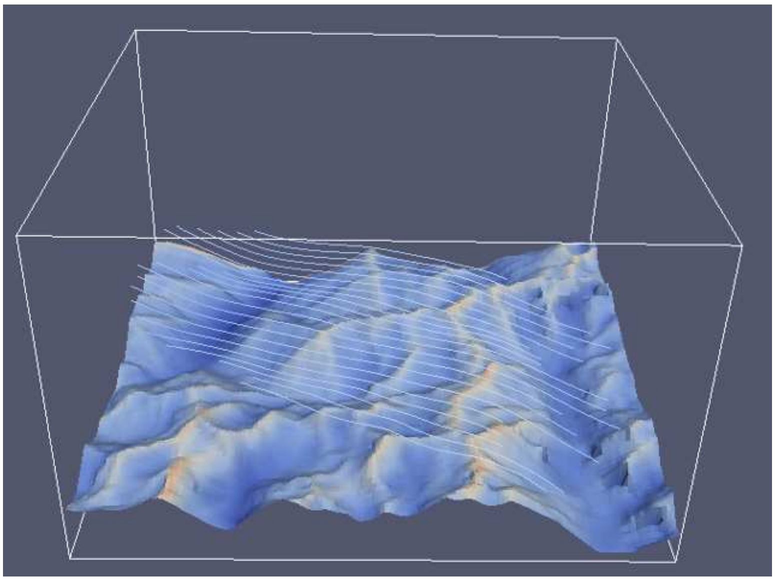


Figure 4.6: Volume of air to be meshed above the Arcalís ski slopes, Principality of Andorra. Observation angle is from due north. Air streamlines are represented in the vicinity of the highest peaks (altitude approx. 2800 m) provoked by a regional wind of $12\text{m}\cdot\text{s}^{-1}$.

From the standpoint of parallel computation, it is necessary to determine an efficient scheme for domain decomposition, taking into account the peculiarities of this specific situation.

4.4.1 Mesh deformation and computational workload

As discussed in Section 3.1, mesh deformation occurs in lower part of the mesh, immediately above the terrain. Authors such as (Weingarten, 1994) note that reducing the number of individual cells is beneficial to computer workload, since the number of equations to be solved is reduced accordingly. However, less importance has been placed on the efficiency of the iterative algorithms -such as the PCG method- that are implemented to actually deal with the equation systems when mesh deformation takes place.

To investigate this behavior, a first experiment was designed in which simple OpenFOAM test case was adapted from the tutorial case *cavity*. The mesh volume consists of a regular cube of 2m in size, meshed into 10 elementary elements along each axis, giving a final mesh of 1000 cells. A uniform flow of constant speed was imposed along all faces of the cube, at an angle to cube faces. Ten iterations of the PISO algorithm were performed, and the total number of iterations of the DICPCG solver for pressure were added up. As noted in Section 4.3.1, this solver performs most iterations during the PISO loop, typically 100 iterations of DICPCG for 1 iteration of DILUPBiCG. The results are given in Table 4.1.

Table 4.1: Total iterations of the DICPCG solver with equal incident flow imposed on all faces.

u_x	u_y	Angle	Total iterations
0.000	1.000	90.000	40077
0.100	0.995	84.261	35002
0.250	0.968	75.519	39918
0.500	0.866	59.999	40363
0.707	0.707	45.000	38886
0.866	0.500	30.001	40404
0.968	0.250	14.481	39931

The total number of iterations vary in less than 13.5% across all incident angles, and no clear pattern emerges that would indicate a relationship between the angle of wind incident direction and mesh cell, and the computational workload to solve the equations.

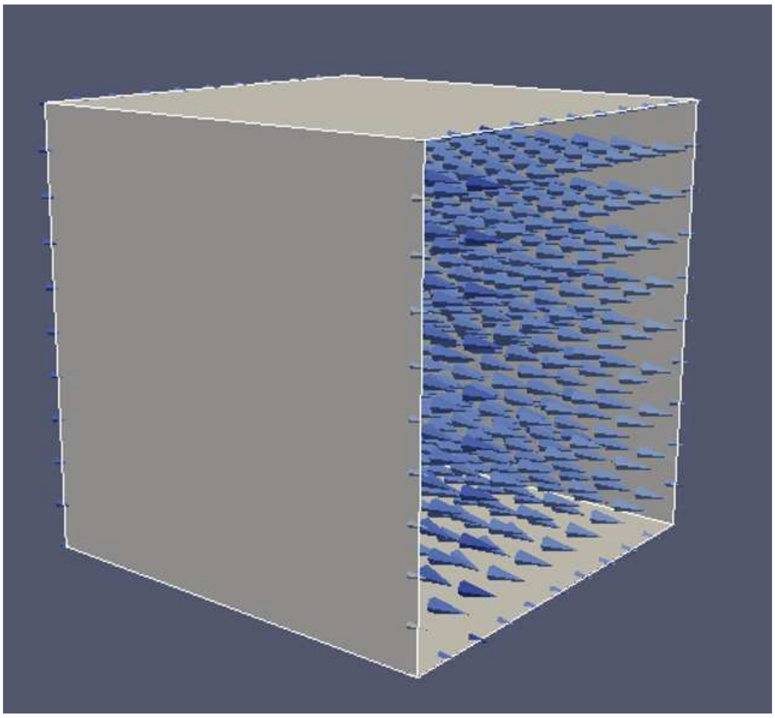


Figure 4.7: Simple cavity with fluid flow imposed on two opposing faces.

In a second experiment, the same cavity geometry was retained. However, in this case the fluid flow across two opposite walls was imposed, while flow along the remaining four walls was set to zero (Figure 4.7).

When the test case was run with the *pisoFOAM* solver, this time execution times depended largely on the angle of incident airflow to the cube (Table 4.2). Incident flows at an angle of less than 20^o to cube geometry had a minimal computation cost of $\approx$ 26000 iterations, while angles of more than 45^o lead to maximal computation costs of $\approx$ 46000 iterations. Between these angles, computation costs progresses regularly.

Table 4.2: Total iterations of the DICPCG solver with fluid flow imposed on two opposing faces.

u_x	u_y	Angle	Total iterations
0.000	1.000	0.000	26004
0.100	0.995	14.481	25798
0.250	0.968	30.001	36526
0.500	0.866	45.000	44941
0.707	0.707	59.999	46030
0.866	0.500	75.519	45918
0.968	0.250	84.261	47943

These results confirm that the computational work with this type of iterative solver is connected not to the direction of flow, but to the gradient of flow speed change. This is coherent with the fact that the PISO method requires many more iterations to solve for pressure; at the same time, higher pressure gradients are directly connected to the rate of change of flow speed by the momentum conservation equation.

4.4.2 Mesh decomposition strategies

When decomposing the volume of air above a terrain, the more pronounced gradients of pressure are to be expected in the vicinity of the terrain. In the top layers, no orographic obstacles exist to block fluid flow and create regions of dynamic pressure increase. For this reason, the lower layers of the mesh increase the computational workload disproportionately. This result has been verified by running a mesh of equal dimensions and number of cells, once with a regular cubic geometry and once with the Arcalís test volume represented in Figure 4.6. Wall-clock execution times for the regular geometry were only $\approx 40\%$ of the times required to model the Arcalís case.

It is thus clear that decomposition strategies in which some processes are assigned to sub-domains at high altitudes while other processes are assigned to lower altitudes must lead to unbalanced workloads.

On the other hand, more advanced decomposition methods such as METIS or Scotch have not been conceived to take into account differences in individual cell workloads. In this specific case, they hold no advantage over simple domain decomposition.

For this reason, simple domain decomposition has been implemented in the final model, using regular vertical slices that reach from top to bottom of the complete mesh volume. However, decomposition may be performed either by partitioning preferentially only along a single axis (e.g. X, Figure 4.8 a), or by partitioning in a balanced fashion along both horizontal axis (X and Z, Figure 4.8 b). If possible, balanced partitioning is preferred since mesh bisection areas (the grayed faces on the diagrams) are reduced, and thus interprocess communication between compute nodes.

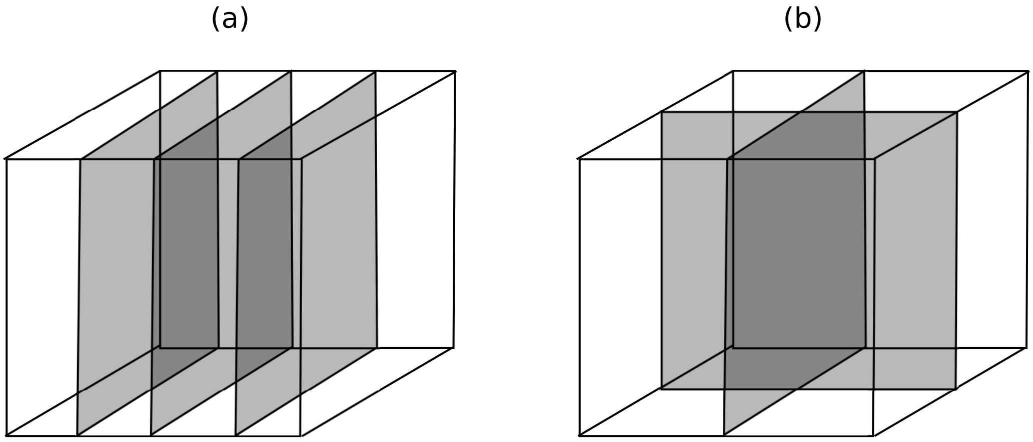


Figure 4.8: Simple domain decomposition by partitioning along a single direction (a) and in a balanced fashion along two different axis (b).

But preferences cannot be immediately established, since workloads depend not only on the ground area covered by each process, but also on the complexity of orography in that area and the pressure gradient produced at ground level. Since it is not easy to foresee the pressure gradient before the circulation model has actually been built, several alternative decomposition strategies may be proposed:

1. If the topography of the site has similar characteristics across the mesh lower face, simply assign sub-domains of equal area to each parallel process. There may be small differences in workload, leading to minor inefficiency during modeling.

2. When the topography is known to contain bands of terrain of similar characteristics, partition the domain across these bands.

3. Start by applying a simple decomposition on a rough mesh with large cell dimensions, then calculate pressure gradient. Use this initial result as a means of evaluating workload and re-partitioning the domain into sub-domains of approximately equal computational cost.

Strategy (1) has been applied to domains such as the Arcalís test case, where decomposition into a small number of regular sub-domains ($n = 4$) ensures all parallel workloads contain a similar amount of sharp ridges and of relatively flat glacier cirque bottoms.

Strategy (2) has been applied to a large-resolution model of the central Pyrenees mountain range (Figure 4.9). The topography of this region consists in first approximation of a relatively flat pre-Pyrenean area to the south and its counterpart to the north, between which the main Pyrenees range stretches from east to west. Thus, a partition consisting in four equal bands running from north to south, each traversing the mountain range and containing parts of (relative) lowlands at each extremity seems to constitute an effective way of decomposing the domain into four sub-domains of equal difficulty from the computing point of view (Figure 4.9).

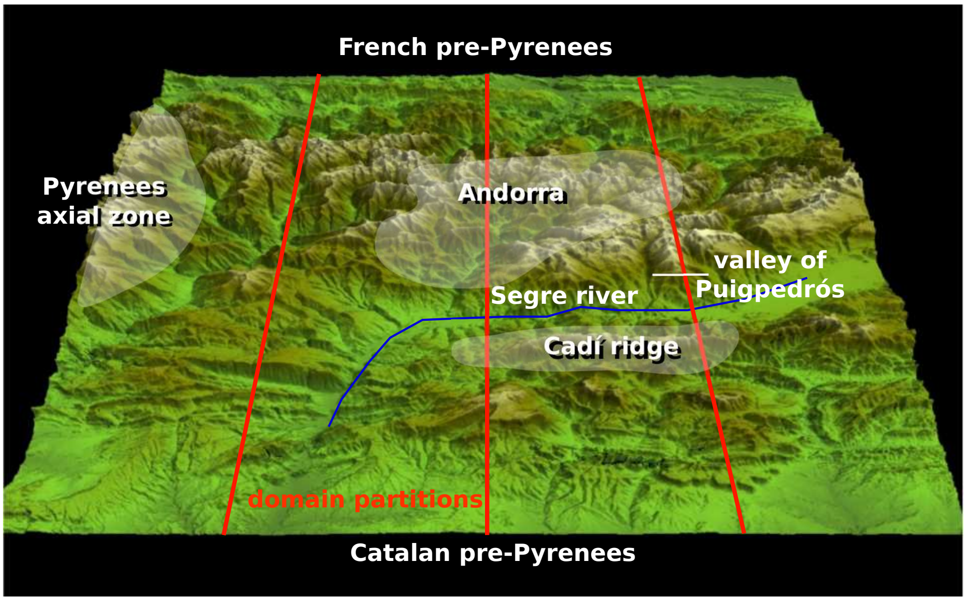


Figure 4.9: Domain decomposition of an area centered on the Principality of Andorra.

However, a closer examination permits observations of several distinctive features. At the western edge of the region under consideration the central, axial zone of the Pyrenees starts in the high Pallars. This is the robust central core of the mountain range, with higher peaks and more difficult terrain with a large vertical development. Should the initial decomposition be maintained, it can be foreseen that the task in charge of the western sub-domain would have a heavier workload on that account.

In a similar fashion, the third sub-domain (from west to east) contains the rock formation known as the Cadí. This sharp ridge is a known obstacle to north-south winds, in the lee of which local depressionary areas are observed, also leading to a local increase of computational workload. These areas of pressure gradient are reproduced in a preliminary rough model of the area.

For this reason, strategy (3) may be particularly effective in such situations: when the site contains irregular areas of smooth surface, when pressure differentials are expected to be low and computation load light, but also other areas of more complex surface characteristics where sharp orographic features lead to locally elevated levels of pressure gradient and computation is expected to be more involved.

Finally, the small river valley from Puigpedrós down to the main Segre river valley running East-West is a specific geological formation approximately 10km long that cuts from north to south into the mountain range. As with many river valleys, regional winds can easily run up and down the valley, while transverse winds are blocked by the mountains to each side (Figure 4.10).

This is why very little pressure gradient is to be found when north-south winds are dominant in the region. On the other hand, east-west winds such as the Llevantada (east wind from the Mediterranean) cause dynamic pressure ridges to form in the vicinity of each side of the valley. Thus, computational workload is locally increased depending on the model's border conditions. In extreme cases, planning a domain decomposition may need to take into account such factors.

Similar decomposition may be performed in other situations in which transverse bands of terrain cross the domain to be studied. An example could be the case of coastal mountain ranges, when the flow of land- or sea-winds is considered (Figure 4.11).

4.4.3 Efficiency of OpenFOAM parallel execution

In this section, efficiency of the parallel computation is defined to be:

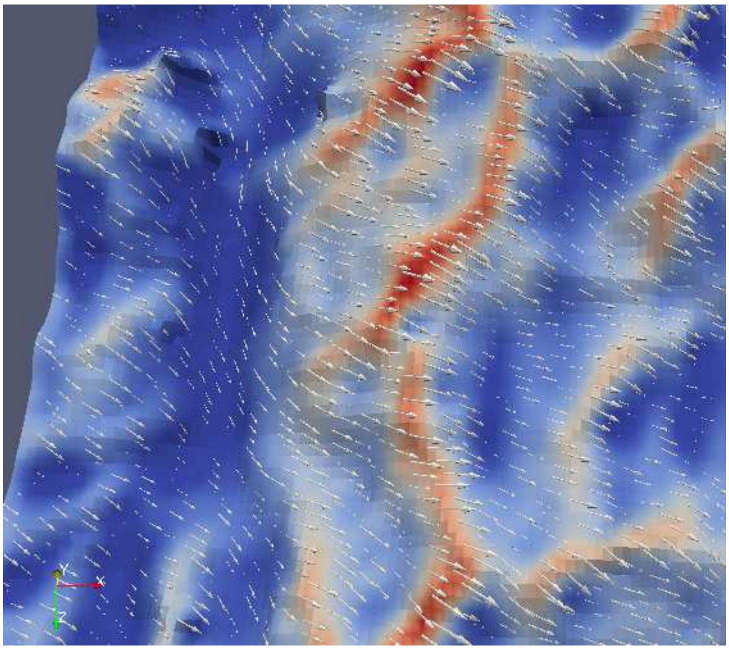


Figure 4.10: Transverse winds across ridges west of Arcalís, Principality of Andorra. River valleys lying transverse to the regional wind direction have little flow at their bottom (in blue), while transverse ridges allow dynamic pressure differential to build up and exhibit strong wind flows (in red).

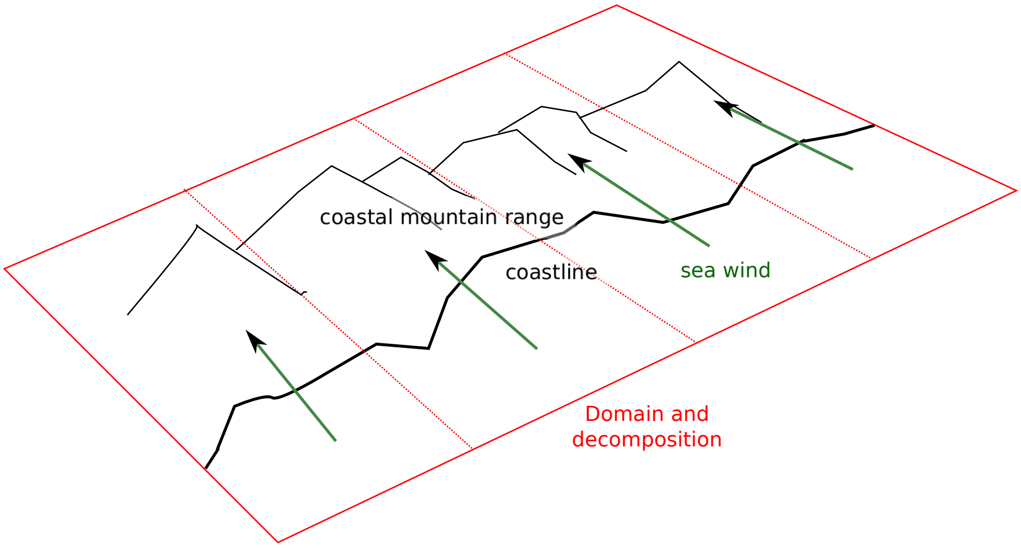


Figure 4.11: Decomposition of a coastline and parallel mountain range.

$$E = \frac{S_r}{S_e} \tag{4.5}$$

S_r denotes the speed-up relative to sequential execution obtained, and S_e expected speed-up when more nodes or CPU cores are used in the task. In an ideal situation, the network infrastructure or messaging library used should not impact expected speed-up, so S_e is taken as directly proportional to the number of CPU cores.

Efficiency may be considered during the two main steps executed during model construction, which both may be run in parallel using the OpenMPI libraries:

1. Meshing and mesh decomposition

2. Model solving

As noted in Section 4.3.2 above, the OpenFOAM solvers are largely dependent on the operation of the subjacent OpenMPI message-passing library. For this reason, attention was given to the network infrastructure.

Model preparation

During model preparation, the *blockMesh* utility is used to prepare the mesh from high-level hexahedra and perform domain decomposition. A typical decomposition case was prepared, using a mesh supported by a 240x240 resolution DEM of the central Pyrenees seen in Figure 4.9. Vertical resolution was set at 100 mesh layers, giving a complete mesh of $5.76 \cdot 10^6$ elementary cells.

Meshing and decomposition was performed on the UOC DCPS Grid cluster, consisting of nodes with Intel Xeon E5440 CPUs or similar, compute nodes with single quad-core processors and the cluster font-end with a dual quad-core. Memory available is 8 GByte/node. Nodes are interconnected with a Gigabit Ethernet network infrastructure. Meshing was parallelized into eight tasks, performed in parallel on a single dual quad-core node. Efficiency was calculated as measured wall- clock time in relation to expected time. Runs were performed three times for each situation, and average execution wall-clock times were taken into account. Deviations from average values were in all cases $< 5\%$. Results are presented in Table 4.3:

It is observed that the use of multiple cores on a single motherboard permits only a two-fold increase in speed, with efficiency degrading as more MPI tasks are executed simultaneously.

Table 4.3: Meshing and decomposing the test case with $5.76 \cdot 10^6$ cells in parallel on a single node. Efficiency is calculated relative to execution of the problem set in a single task on a single node.

Cores used	Wall-clock execution time(s)	Relative speed	Efficiency
1	6475.5	1.0	100%
2	4224.0	1.5	76.7%
4	3406.0	1.9	47.5%
8	3379.0	1.9	24.0%

It is observed that serial operations such as reading and writing to disk are not substantial consumers of wall-clock time during meshing. Input files may be read and output files written within several seconds for this case, to be compared with 2-hour global execution times.

Neither is RAM access a limiting factor, since the complete run was executed from within physical memory.

Finally, network activity was also considered as a source of congestion. However, monitoring network activity both on the client and master nodes during execution shows levels far below saturation of the physical medium.

Running the same meshing process in eight MPI tasks on multiple compute nodes gives the following results (Table 4.4):

Table 4.4: Meshing and decomposing the test case with $5.76 \cdot 10^6$ cells into 8 parallel tasks. Execution on 2, 4 or 8 nodes. Efficiency is calculated relative to execution of the problem set in a single task on a single node.

Nodes used	Wall-clock execution time(s)	Relative speed	Efficiency
2	3606.0	1.8	22.4%
4	2260.0	2.9	35.8%
8	1532.0	4.2	52.8%

In this case, best results are obtained when using a single task (and core) on each of eight nodes. When running four tasks each on two nodes, congestion is apparent as when running multiple tasks on a single node.

Network activity may not be invoked as a cause of said congestion, since the OpenMPI implementation can prefer shared memory access over TCP/IP communication when run on a single node.

Thus congestion is seen to arise both when a single or multiple nodes are used, but only when multiple tasks are executed on a single physical processor. It is thus hypothesized that the limiting factor for *blockMesh* parallel execution is RAM bandwidth from the CPU.

Model execution

During model execution, sequential tasks include:

- Reading in the case definition. This is performed once, on the master node controlling the parallel computation. Typical wall-clock times are in the order of several seconds, comparable to typical case execution times which can amount to several tens of minutes or hours with a mesh of 10^6 individual cells in a calculation involving the PISO algorithm.

- Writing out results, also performed only on the master node. File *system/controlDict* controls the frequency of data dump, so the user can parameter disk access patterns. With moderate amounts of writing (e.g. 50 times per run), disk write usage is also in the order of several seconds over the complete model execution time.

Parameter W_s in Amdahl's law is thus reduced. However, as pointed out in the referenced works, global efficiency when using N to perform calculations is still lower than the ideal speedup value of N.

Similar tests were performed for model execution using the *pisoFOAM* solver in parallel, with similar results. However, in this case network activity was also seen as a limiting factor during calculations.

To investigate the influence of network underlying protocols, both IPv4 and the newer IPv6 TCP/IP stacks were used. The test case was decomposed into eight sub-domains, and the *pisoFOAM* executed both using a *machines* file with IPv4 static address, and IPv6 link-local addresses. Results are similar, though it may be interesting to note the IPv6 maintains a small but noticeable advantage in speed over all tests (Table 4.5).

Table 4.5: Solving the test case with $5.76 \cdot 10^6$ cells in parallel on multiple nodes.

Nodes used	Wall-clock execution time(s), IPv4	Wall-clock execution time(s), IPv6	Gain
2 nodes, 4 tasks/node	138.425	134.853	2.58%
4 nodes, 2 tasks/node	127.112	123.760	2.64%
8 nodes, 1 task/node	113.798	111.149	2.33%

4.5 Chapter conclusions

In this chapter, the modular structure of the OpenFOAM toolkit has been discussed, as refers both to the common file structure used throughout the toolkit and the coherent programming in the C++ language. The application of the PISO solver was presented, applying the Conjugate Gradient family of solvers to find solutions for the speed and pressure fields on the fluid.

Execution of the OpenFOAM meshing and solver utility programs can be performed in parallel. However, this implies the decomposition of the domain to assign different regions to each parallel task. Workloads must be balanced between processes, while at the same time bisection area between sub-domains must be minimized to reduce inter-process and inter-node communication.

In the case of modeling air volumes above mountain regions, simple (regular) domain decomposition may be applied, although attention must be given to regions on the terrain that present local variations of pressure. In these, supplementary iterations of the Conjugate Gradient solvers may be expected, reducing solver speed and introducing unbalance between parallel execution tasks. Locally different topographical structures and interaction with regional winds make the use of a preliminary coarse model appropriate to map out the expected workload over each region of the area under study, and allow the final solver execution to be planned to better balance task workloads.

Finally, congestion is described when meshing or solving is performed in several parallel MPI tasks on a single multi-core CPU. Other potential causes being eliminated, the concurrent memory access between the CPU and RAM is proposed as a cause of congestion not permitting achieving linear scaling when meshing and solvers are executed in a parallel computing environment. It may be observed in this respect that the data-set handled in the test cases weighed in excess of 200 MBytes in RAM, while Level-2 CPU cache sizes on the Intel Xeon E5440 are only 2 x 6 MBytes as reported by the manufacturer (Intel, 2008), and seen as 6 MBytes by each process of the operating system.

As for the solver stage, however, some network saturation is observed which may in a limited manner be mitigated by the use of the IPv6 version of the TCP/IP stack.

In the next chapter, the relationship between airflow and snow particles will be examined.

Chapter 5

Coupling snowfall with transport fluid motion

In the previous chapter, the application of the OpenFOAM computer toolkit mathematical solvers was discussed to solve the Navier-Stokes mass and momentum conservation equations for fluid transport of a volume of air over mountainous orography. Initially, only a fluid with a single component in a single physical phase -such as air- was considered.

In the present chapter, the problem of coupling the flow of air with the transport of snowflakes will be discussed. The problem of the degree of coupling between both fluids in the system will be presented and discussed for snowfall. The effects of border layer models on snow deposition modeling will be commented. Finally the use of parallel computing techniques to calculate individual snow parcel trajectories will be proposed. Bespoke programming by the author is used to add snow transport and deposition to the OpenFOAM air circulation model.

This chapter is in relationship with **Goal 3** of the thesis.

Some of the material in this chapter has been based on previous work in (A. Ward and J. Jorba, 2014).

5.1 Mixed and multiphase flows

In fluid mechanics, certain flows may contain not one but two or more different physical substances, giving us a ***mixed fluid*** or ***multiple component fluid***. Some of these have a tendency to separate naturally, giving rise to regions of the volume under study with different physical properties. An example could be a mixture of water and oil: the water component will tend to separate from the oil and form a clearly differentiated layer at the bottom of a settling vessel. A computer model of this type of mixed fluid must take into account this separation effect. On the other hand, some mixed fluids contain two components that tend to maintain a state of mixture that will be difficult to separate. For example, water and alcohol mixed together form a single fluid composed of diluted alcohol.

In some fluids, one or several of the components are also in different physical phases, in which case the resulting medium is called a ***multiphase flow***. In some systems, fluid components may change phase during physical observation or computer modeling, leading the computer model constructor to introduce a conservation of energy equation into the system to take into account the exchange of phase-change energy. For example, in industrial power production systems that use the couple water/steam as a thermal transport fluid, the circuit relies on fluid phase change and in order to exchange energy from the hot source to the cold. Modeling such a system must therefore take phase change into account, which subject has been extensively studied in engineering [1].

In Table 5.1 several examples of real-world flows and their classification are presented.

Table 5.1: Classification of flows: examples.

	Single component fluid	**Two-component fluid**
Single phase	water flowing through a pipe: there exists just a single fluid component, the most simple system	mixed water and oil in a settling tank: the two components co-exist in a separated state
Two phases	slurry (water, snow and ice) melting in Spring: a solid phase (ice and snow) co-exists with a fluid state (water)	water released by a breaking dam traveling through air: there is some admixture of the water phase, traveling through the gas phase (air)

When both multiple fluids and multiple phases are present, the fluid system dynamics may be influenced by interaction with the boundaries of

[1] For an extensive review of the literature, see (C.T. Crowe *et al.*, 2005), p. 13-27.

the volume considered. In Figure 5.1, the water released by a breaking dam is further perturbed by the presence of an obstacle that deviates it into a high trajectory.

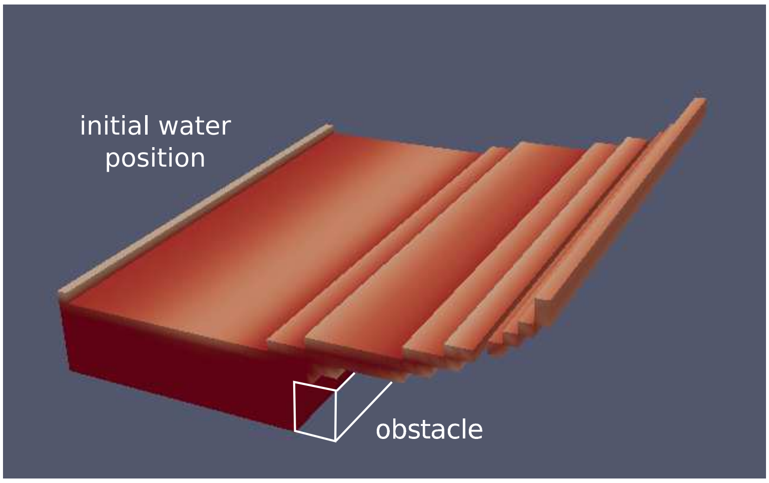


Figure 5.1: Water released by a breaking dam. Adapted from OpenFOAM tutorial “damBreak” case.

Additionally, in physical systems such as during interaction between dissolved gas and environing rock in coupled thermal-hydrologic-mechanical processes (J. Rutqvist *et al.*, 2002), chemical coupling effects must be taken into account and modeled as well.

5.2 Degree of coupling of a mixed fluid

Modeling a mixed fluid system may be performed in various ways, taking into account or not the degree of ***coupling*** between the different fluid components. Coupling may be defined as the exchange of various forms of energy between the components.

1. Mass or kinetic coupling takes place when the motion of one fluid is influenced by that of the others.

2. Thermal coupling is when heat exchange takes place between warm and cold fluid components. Thermal coupling may also take place between a fluid component and the system interface, such as between a CPU heat dissipater and the air-flow around it.

When modeling the mixed fluid, in first approximation, the fluid components may be supposed not to exercise any effect on each other's characteristics of density, speed or momentum. This is known as a system with ***no coupling***.

Though few examples exist of real-world mixed fluids in which no coupling can reasonably be hypothesized, systems exists in which one type of coupling can be neglected while another cannot. For example -(C.T. Crowe *et al.*, 2005), p. 37- the injection of water droplets into a hot air stream will cause the droplets to evaporate. The relative mass of water carried along by the air flow remains the same: there is little modification of the trajectories of each fluid, and mass coupling is low. However, thermal energy transfer takes places from the air to the water, so thermal coupling between the two fluids is large.

For a more complete description of the system modeled, the effects of each fluid on the other may be taken into account, giving a ***two-way coupling*** between fluid components in a two-component fluid, or ***N-way coupling*** in a multiple component mixed fluid.

This is the preferred method when modeling solid-liquid systems, such as floating ships or interaction between sea waves and stationary shore installations (Niels G. Jacobsen *et al.*, 2012). For this type of system, treating both materials as a single fluid and introducing fictitious stresses to model the boundary, while possible, leads to lack of precision in the final solution when fluid-fluid interfaces appear in the computational domain (Tso-Ren Wu *et al.*, 2011).

A third and final alternative supposition would be to consider only the effects of one fluid on the other fluids, neglecting the reciprocal effects. This would be an ***one-way coupling***, that must be justified by very different physical characteristics of the fluid components.

One-way coupling is the basis of the Lagrangian-Eulerian modeling method (Thomas Hughes *et al.*, 1981), for mixed flows in which one component is dilute or discretized in respect to the other. In the terminology of this point of view, the component with major presence is called the ***carrier fluid*** while the minor component is seen as ***discrete particles***.

5.3 Determining the degree of coupling required in a mixed fluid

The flow studied when modeling snowfall has several particularities that must be used so as to delimit clearly which domain will be modeled, as for physical, temporal and thermal aspects. Once defined, domain limits will allow choosing methods to simplify calculations during modeling.

In the first place, the flow consists of two ***different component substances***, air and snow crystals. It is thus a example of a mixed fluid, in which each component has a specific behavior: in general terms, air will flow through the volume modeled and come out on the other side, while the snow component will in some measure be deposited upon the ground surface. If the volume under study is wide and wind speed is not high, it may be supposed that a large proportion of the snow that enters the system from the top boundary will end up deposited on the lower boundary without exiting the model.

On the other hand, when the domain has reduced horizontal dimensions and wind speed is high, blowing snow may enter the domain not from the top boundary but from the side, and exit on the other side without deposition. In such a configuration, the method chosen to model snow movement must take into account a volume larger than the ground area of interest. In Figure 5.2, snow particle movement was modeled around a tree trunk under a mode of blowing snow. The ground area of interest is represented by the green square, with dimensions 4x4 m. The volume of air modeled was extended to each side by a further 2 m to enable all snowflake trajectories ending up within the green area to be modeled and traced.

The two fluid components are in ***different phases***: gaseous for air and solid for snowflakes. It may thus be called a multiphase flow. It is clear that the air component has very low liquefaction temperature (begins at 81.6 K and is complete at 81.6 K under standard pressure conditions, (Daniel Schroeder, 2000))and may safely be assumed to remain in gaseous form during the snowfall process.

However, this is not the case with the snow component. Snowflakes may encounter different conditions during their fall (see section 2.2.1 and (C. Donald Ahrens, 2011)):

1. Passing through a layer of air with temperature above 0^o C may melt the ice crystals partially or entirely. This situation is encountered specially at the end of the winter season.

2. Passing through a layer with low humidity content may allow some sublimation (direct transformation from solid to gaseous phase) to take place.

3. Passing through a layer cold air with sufficient high humidity content may allow supplementary droplets to condense on the flakes' surface, increasing their size and weight (Bergeron process).

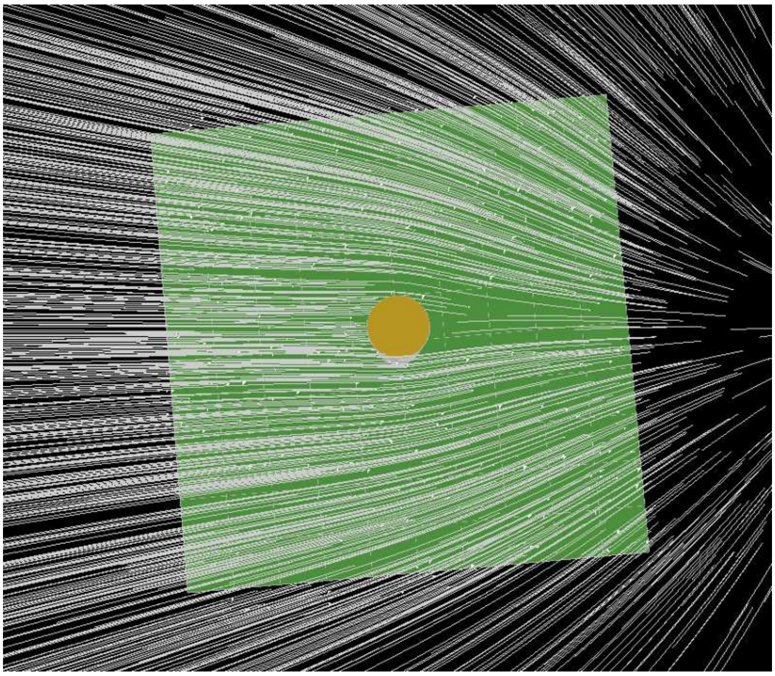


Figure 5.2: Model of snowflakes falling in the area immediately around a tree trunk. Snow crystals with dendrite forms at low temperature and no secondary transport are hypothesized. An application of (Alan Ward, 2009).

4. Finally, once deposited on the ground surface, snow may undergo further phase transformation including melting, sublimation and re-freezing.

In the case of a single fluid but with two different phases we would need to consider the phenomenon of particles changing between one phase and the other, exchanging phase-change energy. The corresponding thermal coupling must be modeled. In the case of snowfall at ambient temperatures, the water component also can potentially exhibit this behavior. When modeling snowfall with snow depositions on the ground, melting snow will receive thermal energy from the other components of the system (ground, air) cooling them at the same time. In this case also, energy exchange with the medium (system boundary, and the air fluid component) must also be modeled.

However, from the computational standpoint, modeling heat exchange requires taking into account and solving the third Navier-Stokes equation, for conservation of energy of the system as a whole. This can be a significant undertaking due to its non-linear formulation. For this reason, it is preferred in models such as SNOWPACK (Perry Bartelt and Michael Lehning, 2002), ALPINE3D (Lehning *et al.*, 2006) and their applications (Gernot

Michlmayr*et al.*, 2008; Florian Kobierska *et al.*, 2011) to separate the two steps of modeling snowfall on one hand, and static fallen snow (snowbank) transformation on the other. Falling snow must be modeled in all three spatial dimensions, while snowbank transformation may be studied using a single-dimension model along bank depth.

In this work, it has been chosen to implement only snowfall during which no phase change occurs, in a similar fashion to that described in (Sundsbø, 1998; F. Naaim-Bouvet *et al.*, 2002; Zhou Xuan-yi and Gu Ming, 2006) and others. This is possible by postulating similar sub-zero temperatures and air water content within the volume to be modeled. In other terms, only the air layer closest to ground level may be accurately modeled, leaving transformational processes that occur in higher layers of the atmosphere to further studies. In this fashion, both fluid components may be supposed to remain in the same phases throughout the actual snowfall event, and the system mathematical representation is simplified by handling only mash coupling and not thermal. Likewise, low-lying regions with temperatures above water boiling point will receive precipitation in the form of rain, and will also be excluded from the present study. Our region of interest is thus limited to a narrow zone between the lower bound of prevailing cloud formations and ground orography, as seen in Figure 5.3.

It can also be noted that snowfall is a mixed fluid with components of ***different relative densities***. Current theory in the field suggests that one-way coupling is suitable when the carrier fluid is not disturbed by the presence of particles (André Kaufmann, 2004). This rejoins the point made in (S. Elghobashi, 1994; Aurélia Vallier, 2010) that the ratio of densities is important in deciding weather to implement one-way (fluid ⇒ particle) or two-way (fluid ⇔ particle) coupling.

As reported in (André Kaufmann, 2004), the ***volume fraction of particles $\mathit{\Phi}_p$***, can be defined as the proportion of particles by mass within a given volume. With (m_p, m_c) the mass per elementary volume and (ρ_p, ρ_c) the relative densities of particles and carrier fluid respectively (Figure 5.4):

$$\mathit{\Phi}_p = \frac{m_p}{m_p + m_c} = \frac{\rho_p}{\rho_p + \rho_c} \tag{5.1}$$

Using $\mathit{\Phi}_p$, various types of particle-laden flows can be defined:

1. ***dense suspension*** for $\mathit{\Phi}_p > 10^{-3}$, in which not only is fluid ⇔ particle coupling taken into account, but also interaction between particles themselves; the authors call this four-way coupling.

2. ***dilute suspension*** for $10^{-6} < \mathit{\Phi}_p < 10^{-3}$, in which only fluid ⇔ particle two-way coupling is considered. In this type of mixed fluid, the

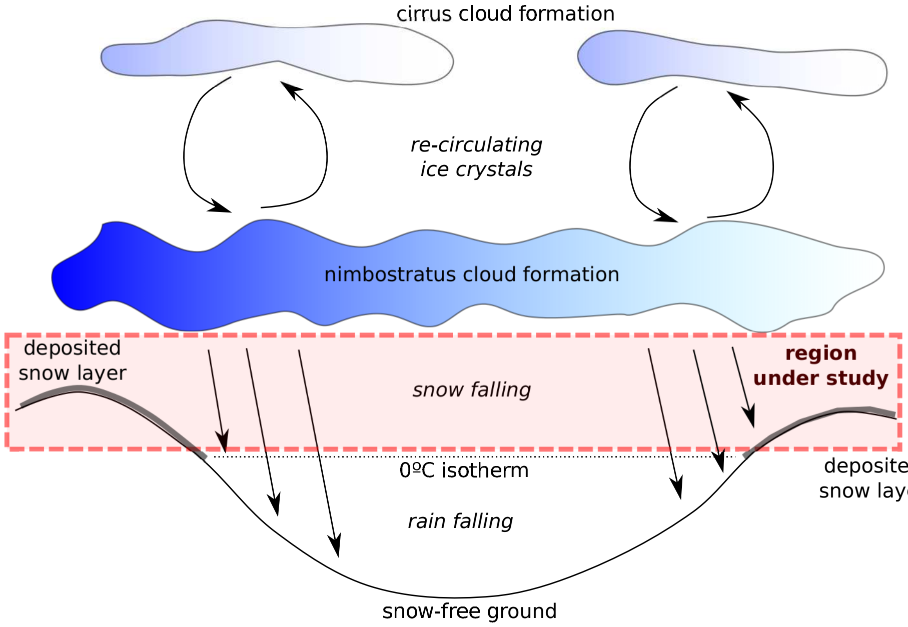


Figure 5.3: Snow fall formation, with phase change to rain over lower terrain. Adapted from (C. Donald Ahrens, 2011), p. 131.

presence of the particles has a measurable influence on the appearance or dissipation of turbulent kinetic energy.

The Stokes number S_t describes the flow of particles within the carrier fluid, and is defined as the ratio between the relaxation time τ of the particles, the carrier fluid speed V and the characteristic dimension of an obstacle obstructing the fluid flow D:

$$S_t = \frac{\tau \cdot V}{D} \tag{5.2}$$

The relaxation time of the particles is the time required for a particle to respond to a change of direction within the carrier fluid. It may be calculated from the particle diameter d, density ρ and the dynamic viscosity μ of the carrier fluid (Don W. Green, 2007):

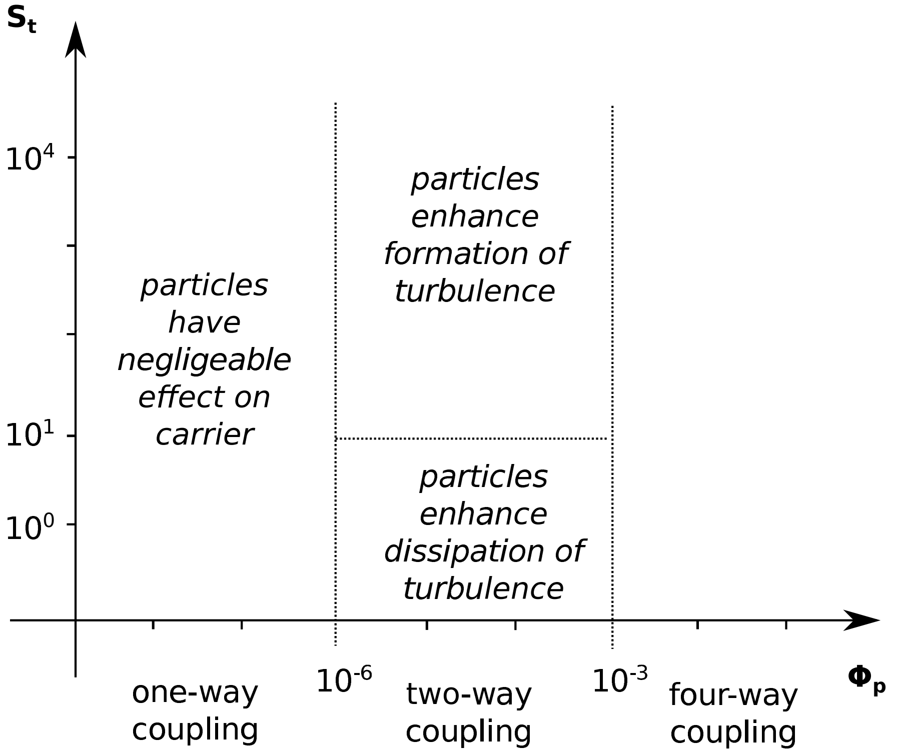
$$\tau = \frac{d^2 \cdot \rho}{18\mu} \tag{5.3}$$

This gives an alternative expression for the Stokes number:

$$S_t = \frac{d^2 \cdot \rho V}{18 \mu D} \tag{5.4}$$

If the Stokes number of the fluid is above a certain limit $S_t > 10^1$, the presence of particles tends to enhance the production of turbulence. On the other hand, if $S_t < 10^1$, their presence enhances the dissipation of turbulent structures.

3. A second form of ***dilute suspension*** for $\Phi_p < 10^{-6}$, in which the presence of particles has negligible effect on the carrier fluid and one-way coupling fluid ⇐ particle is considered.



S_t
10^4
10^1
10^0
particles have negligeable effect on carrier
particles enhance formation of turbulence
particles enhance dissipation of turbulence
10^{-6}
10^{-3}
Φ_p
one-way coupling
two-way coupling
four-way coupling

Figure 5.4: Position of the air-fluid mixed fluid regions, with logarithmic horizontal and vertical scales. Adapted from (André Kaufmann, 2004).

However, the investigators using this theory fail to report clearly how the precise values of 10^{-3} and 10^{-6} are calculated to delimit regions along the Φ_p axis, and why the limit between turbulence enhancing and turbulence mitigation is precisely at $S_t = 10$ on the vertical axis. For this reason, it may be supposed that these values are given as merely indicative of the presence of a change in behavior of the mixed fluid, leaving decisions on the applicability of each mode to the computer model constructor.

That being said, it is not yet clear how the different particle parameters affect carrier turbulence modulation, and the appropriate approach for a two-way fluid $\Leftrightarrow$ particle coupling model -trajectory or 2-fluid model- is still under discussion (C. T. Crowe, T. R. Troutt and J. N. Chung, 1996; Suzuki Yuji, Motofumi Ikenoya and Nobuhide Kasagi, 2000).

This effect is contingent on the presence of turbulence within the carrier fluid: in a steady flow with little turbulence will little influence from the particle presence will be noted. A connection may be may between this fact and the relaxation time τ of the particles that appears in the expression for S_t: turbulence implies changes in fluid movement direction, which particles tend to resist when particle relaxation times (and Stokes number) are high.

More recent developments in computer modeling (Hu Zhiwei, Luo Xiaoyu and Luo Kai H., 2002) conclude that particles with low Stokes numbers can follow the carrier fluid flow closely, since these particles respond quickly to the change of fluid motions.

5.4 Determining the degree of coupling in the specific case of snowfall

Snowfall may be considered a mixed fluid with air as carrier fluid, and snowflakes as the dispersed particle component.

During snowfall, the average deposition height h_d of snow on the ground may vary; Catalonia's automatic meteorologic station network reports (Servei Meteorològic de Catalunya, 2013) maximum values of up to 43mm of daily snowfall at Lac Redon, Vall d'Aran (altitude 2247m) or 63mm at Salòria, Pallars Sobirà (altitude 2451m), both during January 2013. Averaging deposition values over the length of a snowstorm, hourly deposition values may be taken as $h_d = 0.01$ m/hour/m^{-3}. Fresh deposited snow density will be taken as $\rho_{depos} = 8$ kg $\cdot$ m^{-3} (J. Nelson, 2008), and snow particle vertical average speeds as $v_T = 0.95$ m $\cdot$ s^{-1} (Andrew J. Heymsfield and Masahiro Kajikawa, 1987). The average snow density ρ_{snow} may be calculated using:

$$\rho_{snow} = \frac{h_d \cdot \rho_{depos}}{3600 \cdot v_T} \tag{5.5}$$

This gives us the estimated value $\rho_{snow} = 23.39 \cdot 10^{-6}$ kg $\cdot$ m^{-3}. It should be noted that considerable variation may be expected, both of hourly deposition and of particle vertical speed.

The average density of the air component is taken as $\rho_{air} = 1.112 \text{ kg} \cdot \text{m}^{-3}$ at 1000 m altitude above sea level (NASA, 1976), giving $\Phi_p = 21.04 \cdot 10^{-6}$.

The calculated value for Φ_p is clearly below the limit of 10^{-3} separating dilute and dense suspensions, indicating the mixed fluid can be considered a dilute suspension. However, it may be noted that using lower values of h_d would lead to placing Φ_p beneath the limit of 10^{-6} separating operating conditions under which the presence of particles has negligible influence on carrier fluid.

It is also interesting to consider the Stokes number of the flow, as described in the previous section. In the case of snowfall, snowflake diameters can be taken in the range $d \in [0.3 \cdot 10^{-3} - 9 \cdot 10^{-3}]\, m$ (Andrew J. Heymsfield and Masahiro Kajikawa, 1987), with an average value of $d = 4 \cdot 10^{-3}$ m. On the other hand, snowflake density may be taken as $\rho = 0.1 \text{ kg} \cdot \text{m}^{-3}$, fluid characteristic speed as $V = 1 \text{ m} \cdot \text{s}^{-1}$, and air viscosity $\mu = 17.58 \cdot 10^{-6}$ at 1000 m altitude (NASA, 1976).

Characterizing obstacle diameters is more complex, since we may take into account man-made obstacles and tree branches with $D = 0.10$ m or even less, up to $D = 100$ m or more for obstacles created by the form of the terrain. Applying:

$$S_t = \frac{d^2 \cdot \rho V}{18 \mu D} \tag{5.6}$$

We find $S_t \in [50.6 \cdot 10^{-6} - 0.051]$. A very large difference can be appreciated between extreme range values, corresponding to the large variation in possible flow obstacle sizes. The position of the air-snow fluid can be plotted on a graphic with the various domains (Figure 5.5) for comparison. Although both Φ_p and S_t show large variations of their values, some information can be found relevant to computer model construction.

On the left side of the region of interest, low values of Φ_p, associated with low-density snowfalls and high snowflake terminal velocities (wet, heavy snowflakes at high temperatures), may be modeled using only a one-way coupling. However, higher densities of snowfall or snowflakes with low terminal velocities (dry, light snowflakes at low temperatures) give the higher values of Φ_p towards the right, and in such a case computer model construction must take into account both the influence of the carrier fluid on the particle and vice-versa. This is coherent with casual visual observation, that shows that individual snowflakes may take complex fall patterns when light and dry such as in early winter (late October - November) falls. On the other hand,

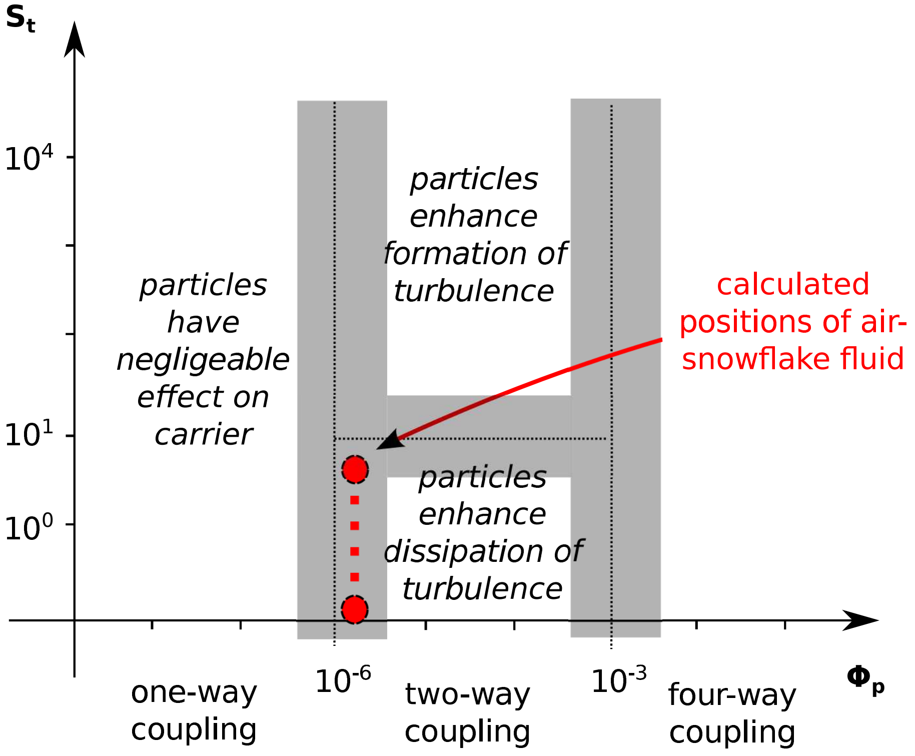

S_t
10^4
10^1
10^0
particles have negligeable effect on carrier
particles enhance formation of turbulence
particles enhance dissipation of turbulence
calculated positions of air-snowflake fluid
10^{-6}
10^{-3}
Φ_p
one-way coupling
two-way coupling
four-way coupling

Figure 5.5: Position of the air-fluid mixed fluid. Imprecision of region boundaries is represented with wide gray borders.

the heavier snowflakes associated with spring falls (late February, March and April) can be seen to fall more vertically, in a similar fashion to raindrops.

In the vertical sense, it may be observed that although the Stokes number has considerable leeway, should the effect of particles on the carrier fluid be taken into account it shall always be with a tendency to dissipate fluid turbulence. However, for the low values of S_t expected ($S_t < 0.051$), individual particles (snowflakes) should be in a quasi-equilibrium status with the fluid and follow carrier turbulent structures closely.

5.5 Computing parcel trajectories in a parallel environment

In computational terms, in an Eulerian calculation the continuous fluid parameters under study are assigned to mesh nodes. The mesh is divided into areas that are assigned to each compute node available. Calculation requires each compute node to have access not only to the values pertaining to the mesh cells assigned to that node, but also to data pertaining to neighboring cells. These may or may not be assigned to the same compute node; in the case of neighboring cells attached to separate compute nodes, the calculation gives rise to some data exchange between nodes - which must go through the network communication layers and is slower than data transport within each compute node.

On the other hand, in a Lagrangian calculation, the variables under study are the values of speed associated with each discrete particle. Particles may be assigned to separate compute nodes. When each particle's movement is not coupled with one another (since $\Phi_p < 10^{-3}$), the particle's trajectory may be supposed independent of other particles, and its calculation likewise. So compute nodes may work in independence of each other, with very little inter-node communication taking places.

From the standpoint of the computer model, implementing one-way or two-way coupling has a notable effect on how the calculation is implemented. One-way coupling implies calculating the parameters of pressure and speed for the carrier fluid first, and then in a separate step calculating the movement of the discrete particles carried by the fluid (Figure 5.6, top). The first calculation step is Eulerian in nature, while the second uses Lagrangian calculation.

When one-way coupling is implemented, the small-grained Eulerian step will take place as usual, but then the parallel Lagrangian step may be performed in an efficient way from the point of view of the use of communication layer.

On the other hand, implementing a two-way coupling is more involved, since the Eulerian and Lagrangian calculations are coupled together (Figure 5.6, bottom). The particle position data is also accessed by each compute node during the calculation of the carrier flow parameters, increasing inter-node communication needs and decreasing overall efficiency.

Scaling may also become an issue when the number of snow particles is increased. An embarrassingly parallel purely Lagrangian calculation will scale upwards in a nearly linear fashion with the number of nodes. However, an Eulerian or coupled Eulerian-Lagrangian calculation will attain communication fabric saturation at lower levels, increasing calculation time.

When implementing a singly coupled model, particle trajectories are governed by airflow and the only parameterizable aspects of their behavior are the mass of snow injected into the system at input level, and the speed of fall (terminal velocity). On the other hand, if a two-way coupling is introduced, the mass of snow at input will have an implication on the relative density of

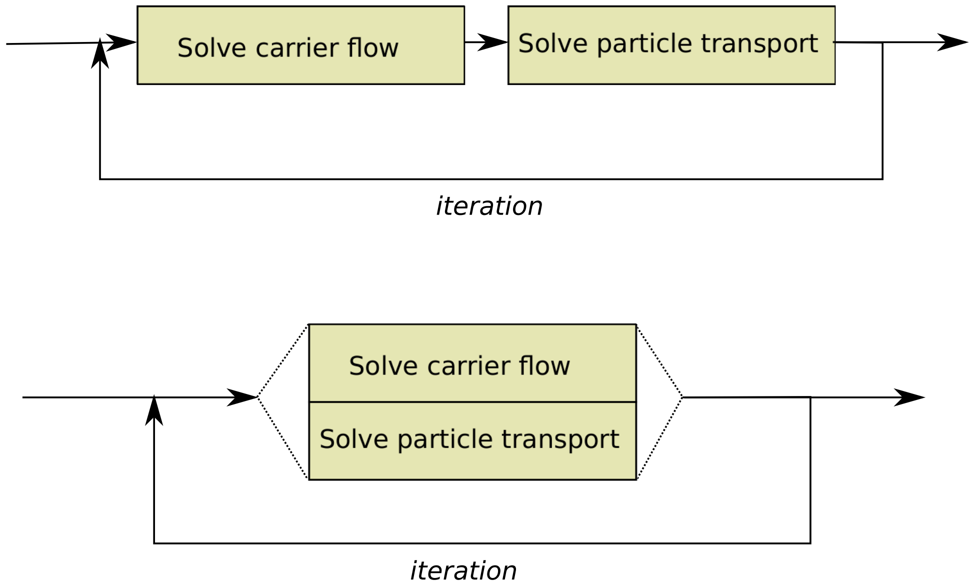


Figure 5.6: Calculation flow of a mixed fluid with one-way coupling (top) and two-way coupling (bottom).

snow per volume unit, and thus also govern the snow-air coupling through the viscous forces that appear within the mixed fluid.

Further parametrization, to take into account snow transformation during fall, implies solving the Navier-Stokes equation for energy conservation as well, adding heat exchange between the two fluid components to the model.

5.6 Experimental results

The “Tunnel de les dos Valires” was built in 2012 to connect the parishes of La Massana and Encamp, Principality of Andorra. On the La Massana side, a complex suspension bridge (“Pont de Lisboa”) was built to connect the tunnel mouth to the existing road infrastructure. The bridge consists of two separate two-lane road beds, one in each direction of traffic.

The bridge crosses the Valira river valley in the East-West direction, and is subjected to north winds during winter that carry snow whenever a suitable front comes in from the Atlantic, blowing orthogonally across the road bed, from left to right in Figure 5.7. Visual observation during the winters of 2012-13 and 2013-14 show that there is a marked tendency for snow banks to form on each side of each road bed.

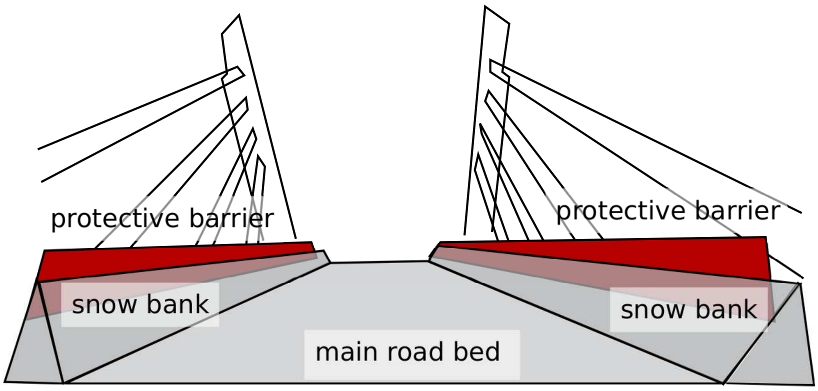


Figure 5.7: North-south perspective of the Pont de Lisboa, La Massana, Principality of Andorra.

A 3D computer model of primary and secondary snow transport was constructed, implementing a mixed Euler-Lagrange model in two steps.

In the first step, the Reynolds-Averaged Navier-Stokes equations for incompressible flow are solved using the CFD toolkit. The $\kappa - \epsilon$ turbulence model (W. P. Jones and B. E. Launder, 1972) is used in conjunction with a simple slip boundary layer condition and solved with the PISO method (R. I. Issa, 1986). This model is in essence identical to that used in our previous work (A. Ward and J. Jorba, 2013).

In the second step, a simple Lagrangian model is constructed to follow the movement of individual snow particles in two stages. In the first stage, low wind conditions typical of heavy snowfall are used to calculate primary snow deposition during the initial snowfall period. In the second, secondary wind transport is modeled in the higher wind conditions typical of a post-fall time period, during which existing surface snow is eroded by wind and re-deposition results in snow-drift formation and compaction.

Snowfall heights of $h_d = 8$ kg/hour/m^{-3} are considered, and snow particle vertical speeds of $v_T = 0.7 \text{ m} \cdot \text{s}^{-1}$ giving an average density of the snow component $\rho_{snow} = 3 \cdot 10^{-3} \text{ kg} \cdot \text{m}^{-3}$. The average density of the air carrier fluid is taken at $\rho_{air} = 1.112 \text{ kg} \cdot \text{m}^{-3}$.

Both road-beds of this double bridge carry suitable safety equipment, including a protective barrier on each side, situated between the road bed and the suspension cable anchorage points. However, this approximately 1 m-high porous barrier acts in winter as a rather effective snow fence, slowing down air flux above the bridge road-bed and allowing areas of low air pressure to form in the lee of the upwind obstacle and across the bed itself. This situation has been modeled using solid obstacles at the bottom boundary of the calculation mesh. Results of the first calculation stage are shown in Figure 5.8, with color-coded relative air pressure (blue is low) and particle streamlines showing carrier fluid motion above the road bed.

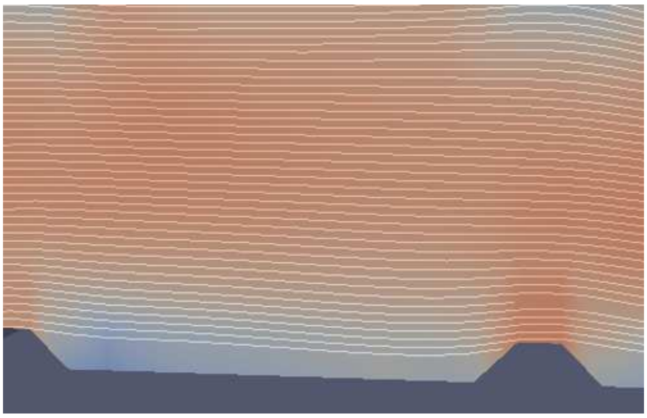


Figure 5.8: Carrier flow circulation model (stage 1) above the Pont de Lisboa, La Massana, Principality of Andorra.

These areas of slow-moving, low-pressure carrier flow observed above the road bed beneath the tops of the lateral snow-banks permit snow particles permits snow particles to fall directly onto the road-bed without any perturbations, and to accumulate and deposit on the bridge surface. Neither can any secondary transport take place since the main flow of air is blocked mainly by the upwind obstacle, though the downwind obstacle also has an influence.

The snow transport stage of the computer model confirms these results. We see an initial deposit of snow (primary transport, in blue in the figure) that conforms to the surface, both on constructive elements and already accumulated snow. The deposit has been modeled in successive time-steps, shown as dashed lines in Figure 5.9 (a).

The mechanical action of snow-plows required to clear the road further develop and compact the snow bank on each side, consolidating the obstacle to the crosswind and allowing further snow deposits to accumulate. This external effect has not been built into the computer model.

The deposited is then reduced by secondary wind transport. As shown by the computer model, erosion affects only snow that is directly exposed to

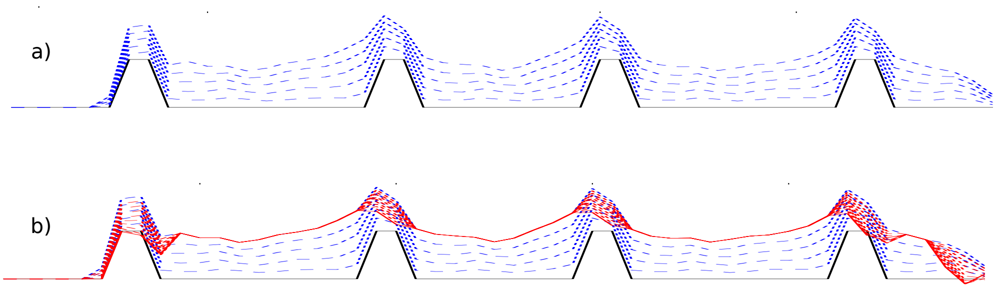


Figure 5.9: Snow particle deposition and secondary transport computer model (stage 2) on the Pont de Lisboa, La Massana, Principality of Andorra. (a) Initial deposition, (b) erosion process.

the carrier flow above the lateral guide-rails, leaving the main body in place occupying the totality of the road-bed. Some secondary transport results in a small increase of the snow deposited on the main road-bed. Successive time-steps of the erosion process are shown in red dashed lines in the figure, with the final snowbank form in a continuous red line in Figure 5.9 (b).

These results are in agreement with visual observation on site. The general mechanism of deposition and partial erosion is in place, resulting in larger deposits of snow are formed on the bridge road-bed than can be found on the neighboring highway. This situation could be alleviated by making the guard rails more open to airflow, which could then sweep at least part of the freshly-deposited snow off the bridge.

Further validation may be undertaken from a more quantitative standpoint. However, this site is on a main road that needs to be open for traffic at all times. Regulations forbid the presence of people on foot on the roadbed itself. For this reason, photographic means may need to be used, to record the height of the snow bank at different times during snowfall.

Automatic digital cameras usually used for non-intrusive wildlife surveillance such as the Acorn 6210MC (Figure 5.10) could be a cost-effective solution to record snow height progression. However, a suitable reference would need to be made available to ensure correct interpretation of the photographic data.

5.7 Chapter conclusions

In this chapter a review of the structure of mixed and multiphase flows has been presented. The concept of degree of coupling necessary to build a computer model of a mixed fluid has been introduced. Methods of determining

Figure 5.10: Acorn 6210MC HD Wildlife Trail Camera.

such a degree have been presented, and applied to the specific case of modeling snowfall. Arguments in favor of a one-way coupling mechanism have been advanced both from the physical and the computational standpoints. A specific example of the complete snowfall model has been constructed and results verified on site.

In the next chapter, similar models will be built, with application to specific problematic conditions that arise through the development of human activities in mountainous areas.

Chapter 6

Case Studies

In the previous chapters, aspects of building a computer model that represents wind circulation above mountainous terrain, and coupling the motion of the fluid with snow transport were treated. In this chapter, several applications are presented of the combined snow transport model. All three may be used in the planning phase for installations related to human activity in mountainous areas, installations that incur expenses both in economic terms, and as an impact upon the natural habitat. From both points of view, gaining knowledge about the effects of the installation on wind circulation and snow deposition before initiating construction will be beneficial towards minimizing costs.

- In the first, *Ski slope planning*, the use of snow cannon is investigated in an environment in which ambient wind flow may pass along the ski slope to be treated -thus helping regular snow deposition- or across the slope with the effects of disturbing deposition and making the artificial snow production less efficient.

- In the second case under study, the snow transport model is used in both its primary and secondary snow transport phases to model *snowdrift formation* on high altitude roads. These are of interest when managing the logistics related to ski centers. A method of road talus profiling is suggested that can help reduce snowdrift formation and the needs for mechanical on-road snow removal.

- Finally, in the third case study a collateral use of the carrier fluid flow part of the transport model has been developed and applied to the *planning of wind turbine installations* in mountain terrain. By combining the fluid circulation model with a model of the wind turbine blade assembly, particularities of wind flow across mountain ridges have been observed and their effects of power production and stress production within the turbine have been modeled, with repercussions on the financial aspects of wind turbine installation and the projected lifespan of the machinery installed.

6.1 Ski slope planning

In modern ski slopes, rising numbers of skiers and the need to ensure adequate snow cover during the complete ski season have induced European domains to invest heavily in installations for making artificial snow. Fears of climate change and reduced snow cover affecting skiing and associate activities (Daniel Scott and Geoff McBoyle, 2007) may also have weighed in the balance. However, current generation snow generation equipment has considerable requirements for operation, both in terms of electrical energy.

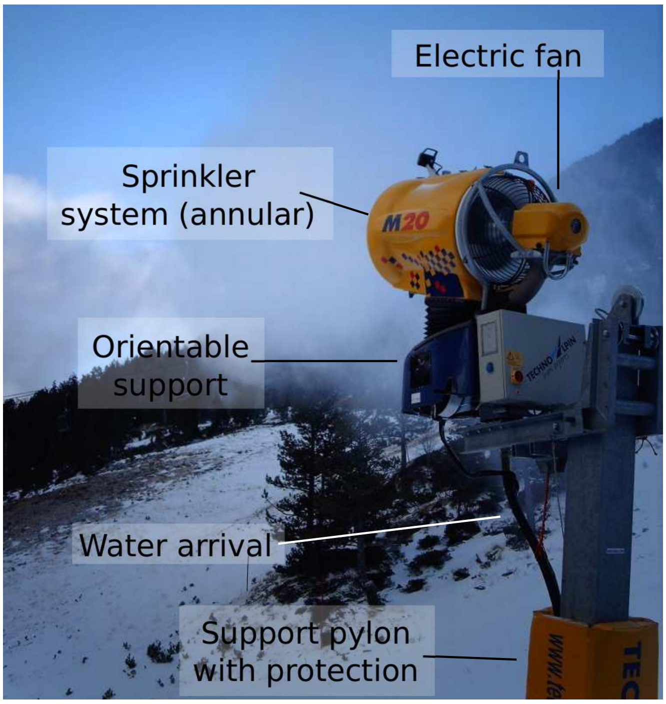


Figure 6.1: Snow gun and its constitutive parts.

Two types of equipment are currently in use. In the snow fan gun (Figure 6.1), an electric motor powers a fan that produces an air current within an orientable cylinder. At the cylinder mouth, a circular injection system sprays pressurized water into the air flow, thus cooling water droplets into small crystals. The complete head assembly is oriented as desired, controlling where the projected snow is to be deposited in the area surrounding the fan gun. The mechanism requires both an arrival of water within temperature and pressure conditions specified by the manufacturer, and an electrical supply.

On the other hand, a simpler means of creating artificial snow is the snow lance. This is simply a long hollow tube, at the tip of which an injection mechanism is placed. The water is mixed with compressed air, thus providing the propulsive force necessary to operate the mechanism. In this case, no external electrical supply is needed.

In both cases, energy expenditure to produce artificial snow includes the cost of pumping water from available sources up to the snow creation mechanism, air compression if required and electricity consumed by fans.

It has been estimated that about $0.6-0.7$ kWh $\cdot$ m^{-3} electrical energy for lances and $1-2$ kWh $\cdot$ m^{-3} for fan guns is necessary, as well as a water supply of about $400-500$ kg $\cdot$ m^{-3} (Jörgen Rogstam and Mattias Dahlberg, 2011). High operating costs are thus an important factor to plan investments for ski stations (Christian Rixen, *et al.*, 2011), specially those situated at lower altitudes where the use of artificial snow may become critical to maintain skiing activity (Catherine M. Pickering and Ralf C. Buckley, 2010).

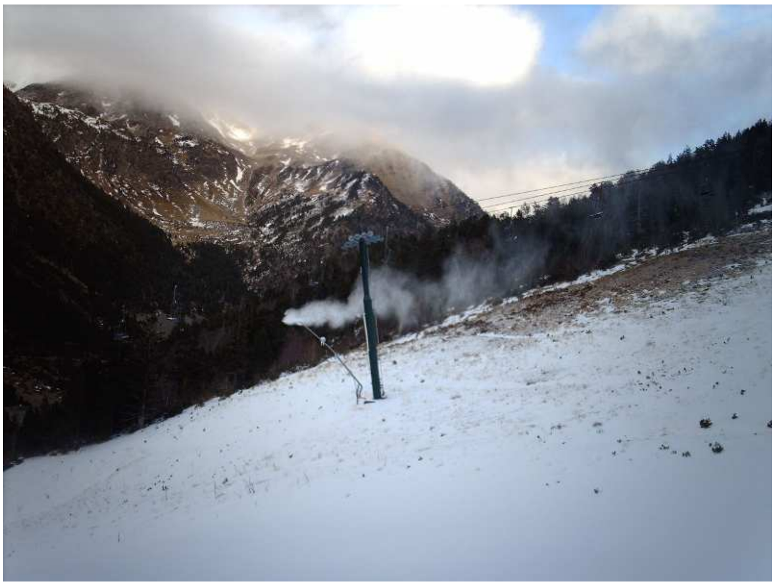


Figure 6.2: Snow lance production being blown completely off-slope at Arcalís, Principality Andorra.

However, the operation of such equipment depends not only on their own characteristics such as ejection speed, but also ambient air flow (Figure 6.2).

In the first case study, airflow is modeled around a snow cannon and a snow lance situated at the Arcalís ski slopes, Principat d'Andorra (latitude 42°38'N, longitude 1°30'E), during testing. The test period initiated at 15:09 local time on November 14, 2013, and ended at 16:09. During this period the test site was photographed from two different angles at 5-minute intervals.

6.1.1 Constructing the computer model

Regional weather information was obtained from automatic weather stations both from (Servei Meteorològic de Catalunya, 2013) and from (Servei de Meteorologia, Govern d'Andorra, 2013). Data from the nearby stations at Arcalís itself (Arcalís SAIH) and the valley leading to the ski slopes (Les Salines) was not used since wind-speed was not available at the time. Similar weather stations situated at the bottom of river valleys were also not taken into consideration since the effects of orography leads consistently to wind readings inferior to regional average. For this reason, data obtained at high-altitude weather stations was privileged, at:

- Port d'Envalira, Andorra: latitude 42°32'N, longitude 1°43'E, altitude 2.510m
- Pic de Salòria, Catalonia: latitude 42°31'N, longitude 1°23'E, altitude 2.451m
- Certascan, Catalonia: latitude 42°43'N, longitude 1°18'E, altitude 2.400m
- Bonaigua, Catalonia: latitude 42°40'N, longitude 0°59'E, altitude 2.260m

During the testing period, the stations at Port d'Envalira and Pic de Salòria reported maximum gust speeds of 11 $\mathrm{m \cdot s^{-1}}$ and 13.3 $\mathrm{m \cdot s^{-1}}$ respectively, both at a heading of 350°. On the other hand, the two more westerly stations of Certascan and Bonaigua reported gust speeds not exceeding 7.6 $\mathrm{m \cdot s^{-1}}$. Wind headings were constant at about 258° for Bonaigua during the preceding hours, while wind heading was at 184° at 15:30 local time for Certascan, though it had varied considerably going from East to West previously.

The synoptic situation that was drawn up from the previous information and local weather bulletins (Figure 6.18) shows general regional wind flow of moderate strength from North to North-West over much of France and Andorra, with a greater western component in the vicinity of the Atlantic Ocean.

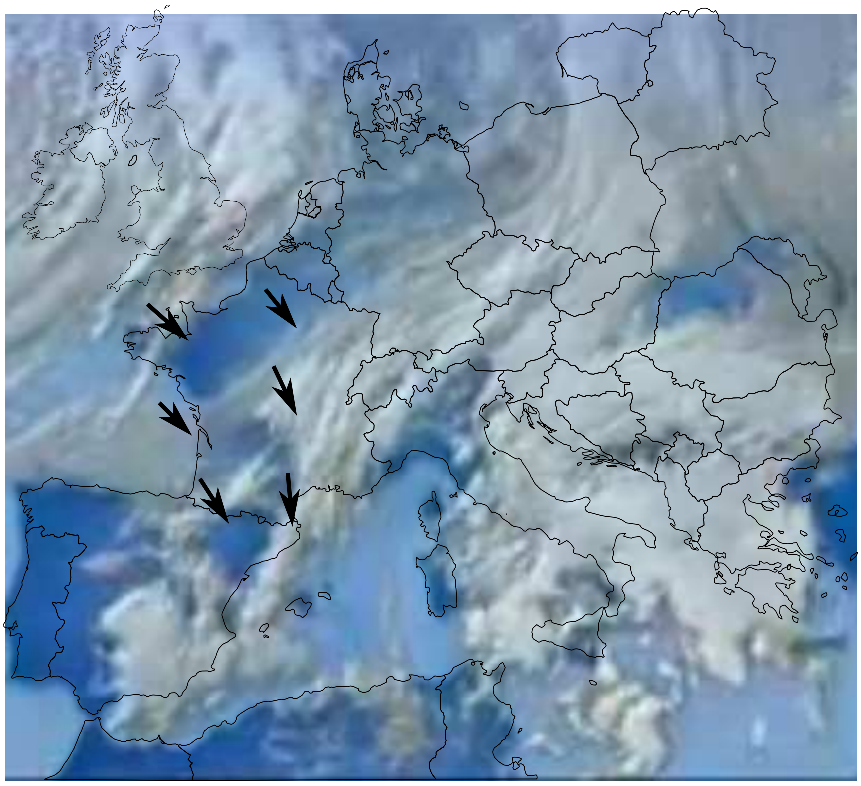


Figure 6.3: Synoptic wind situation for November 14, 2013. Adapted from Météo-France, "Prévisions Météo-France du jeudi 14 novembre 2013", and Marcin Floryan, "A plain SVG map of Europe with countries coded using the 2-letter ISO codes."

This is, in fact, a recurring situation during the beginning of winter season (Pere Esteban *et al.*, 2005).

A constant wind speed of 12 $\text{m} \cdot \text{s}^{-1}$ at a heading of 350° was chosen as input for the computer model.

6.1.2 Computer model execution

Topographical data was taken from the Space Shuttle Radar Mission (T.G. Farr *et al.*, 2007). This data set presents 1 x 1 degree (latitude x longitude) height grids, which at our latitudes give a horizontal resolution of 68 m in the East–West direction, and 93 m North–South. With this Digital Elevation Model data, three successive computer models were prepared.

In the first model, the entire height grid was used, measuring $82.074 \cdot 111.320$ km and covering a large part of the central Pyrenees. This initial grid was meshed using a coarse mesh of 100 x 100 x 100 mesh elements. The Reynolds-Averaged Navier–Stokes equations for incompressible flow were modeled using the well-known standard $\kappa - \epsilon$ turbulence model and simple slip boundary condition was solved numerically with the PISO method, as described in Chapter 4.

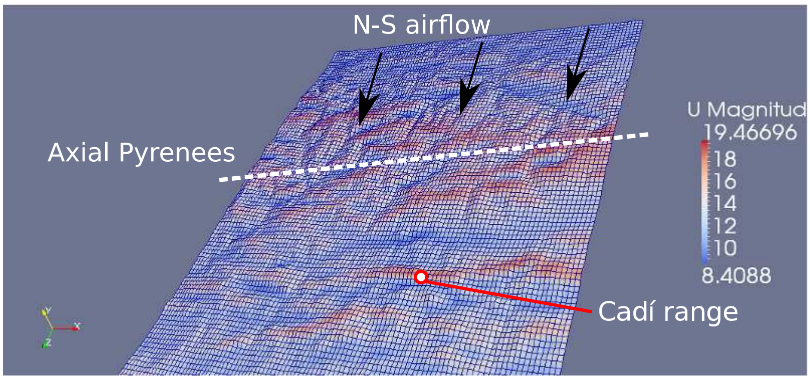


Figure 6.4: Graphical output of the 82.074 x 111.320 km grid.

Model execution was maintained until stability was observed. At this point, were are not concerned with the stability of individual iterations using the PISO scheme, as set out in Section 4.2.3. Instead, the point that needs to be examined more closely is the time necessary for variations in air speed and direction at the boundary of the volume to propagate throughout. As a conservative measure, a minimum end time for the simulation was calculated using the average speed value observed at the boundary $\vec{u}_{ave}$, and the maximal distance transversed through the model $\max d$. A minimal final time step value was then defined as:

$$TF_{min} = \frac{\vec{u}_{ave}}{\max d} \tag{6.1}$$

Using values $\max d = 111320$ m and $\vec{u}_{ave} = 12 \text{ m} \cdot \text{s}^{-1}$, we obtain $TF_{min} \approx 10^4$ s.

It was observed that convergence was significally aided by initializing all values for $\vec{u}$ within the mesh with the average boundary value $\vec{u}_{ave}$.

Once stability was achieved, pressure and fluid velocity values were extracted from the model (Figure 6.4).

In this initial coarse output, it can already be observed that a disparity of wind speeds has developed, with peak speeds of approximately 19 $\text{m} \cdot \text{s}^{-1}$ across the top of ridges running East-West across the wind. On the other hand, lower wind speeds are observed in cross-wind valleys. The barrier effect against North-South regional flow of the Pyrenees mountain range can be seen, as well as that of large topographical formations such as the Cadí mountain range towards the lower third of the domain.

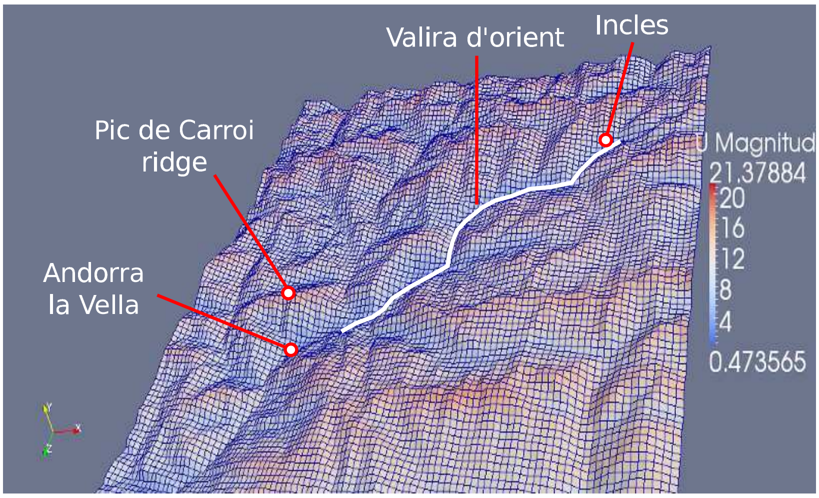


Figure 6.5: Graphical output of the 34.130 x 46.292 km grid.

The second model was prepared with a size of 34.130 x 46.292, centered around the study region. This second model was also meshed with a mesh of 100 x 100 x 100 elements. Boundary conditions were taken from the output of the previous model and model execution performed in the same fashion as before. Pressure and fluid speeds were extracted from the second model (Figure 6.5).

In this model, the main river valley of Valira d'Orient can be seen to bisect the domain, flowing from the Incles valley at the top right down to Sant Julià de Lòria at the bottom left. The sharp Carroi ridge above Andorra la Vella at center left is shown to have high wind speeds in its lee, though the city itself in the valley below it suffers only moderate-to-low ($2-6\ \mathrm{m}\cdot\mathrm{s}^{-1}$) wind-flow.

Finally, a third model was prepared 4 x 3 km in size, centered on the test site in Arcalís. This final model received the output of the second model as boundary conditions, and the model was executed as before.

In Figure 6.6, the lakes of Tristaina glacier cirque can be seen to the top left hand side of each image, showing consistent air flow across the lakes' surface. However, the Tristaina peak (altitude: 2644 m) at to top right blocks the airflow as is reflected in the high fluid speed values in the East-West transverse cut through the model seen in the top image. As a result, air flow in the valley below the peak is reduced to values in the $4-6\ \mathrm{m}\cdot\mathrm{s}^{-1}$

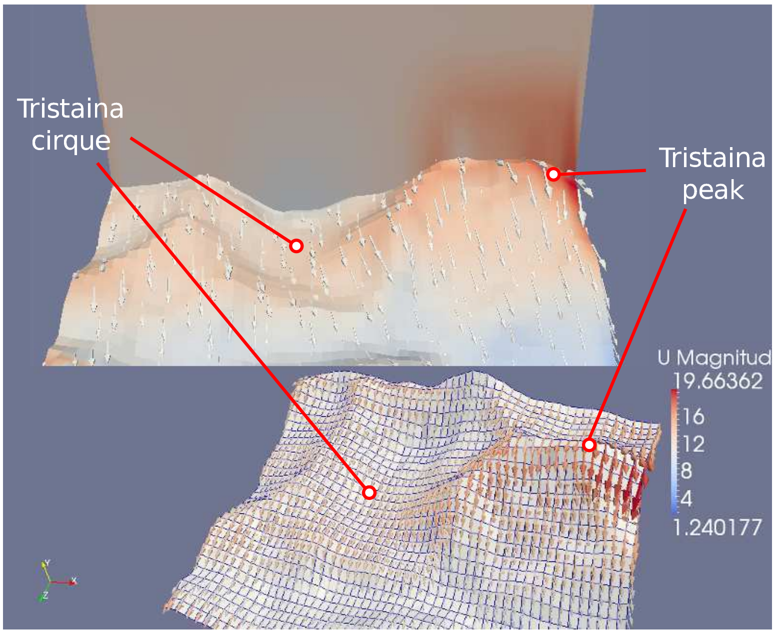


Figure 6.6: Graphical output of the 4 x 3 km grid. View from the South.

range.

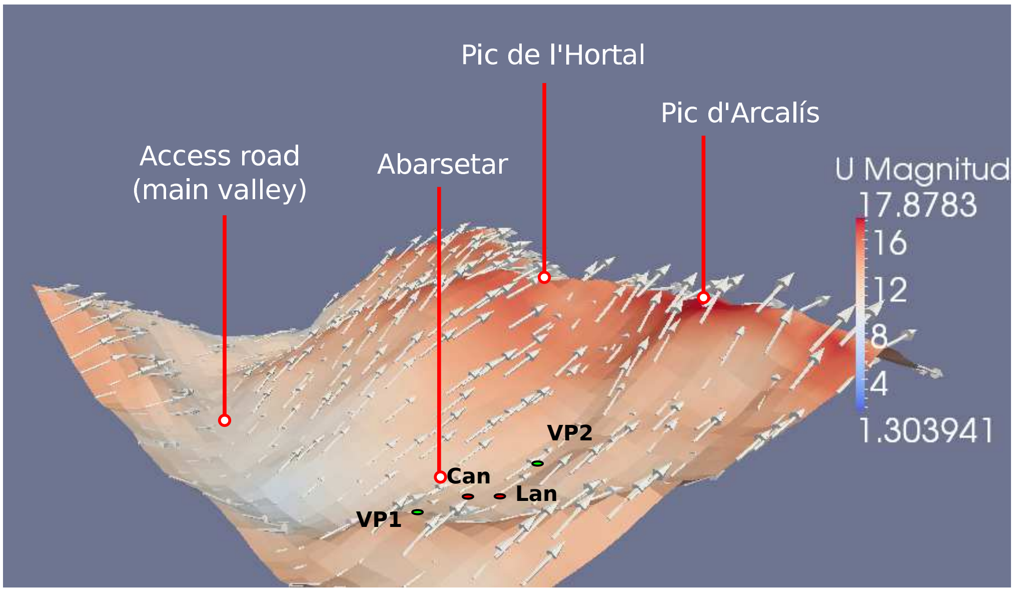


Figure 6.7: Positions of snow cannon "Can" and snow lance "Lan" superimposed on the output of the 4 x 3 km grid. Viewpoints from which photography was taken are situated at "VP1" and "VP2". View from the NW.

The Arcalís station and ski slopes are situated to the South of the Tristaina peak and facing it. The graphical output from the third model was rotated and is shown as seen from a camera position to North-West of the slopes in Figure 6.7.

From this position, it can be seen that the computer model foresees high flow speeds at the ridges of Pic de l'Hortal and Pic d'Arcalís. Although the main river valley and access road are sheltered by Pic de Tristaina and Pic de Font Blanca outside of the area modeled, this is not the case for the ski slopes themselves at Abarsetar. These receive almost direct flow from the 350º regional wind heading, though the flow is somewhat deviated by the ridge on which viewpoint 1 ("VP1") was placed. Wind speeds rising along the valley can be expected at about 12 $\mathrm{m} \cdot \mathrm{s}^{-1}$.

6.1.3 Experimental results

Cameras were placed at view points 1 and 2 (noted "VP1" and "VP2" respectively in Figure 6.7), and photographs taken during operation of a snow cannon "Can" and a snow lance "Lan". The cannon was placed on top of a pylon, while the lance was at normal operating height above ground level.

As is regular procedure in this ski domain, both cannon and lance were pointed downslope, into the oncoming wind flow.

During lulls in the wind (Figure 6.8 (a)), normal operating conditions occurred and the snow cannon was able to deliver its load correctly with deposition occurring within the usual $2-80$ m radius from the cannon (Vijay P. Singh, Pratap Singh and Umesh K. Haritashya, 2011).

However, gusts of wind arose along the slope as predicted by the computer model. As their speed increased, cannon artificial snow production was deviated by more than 90º from its intended direction (Figure 6.8 (b)), eventually dispersing over a large area with no visible impact region on the ground surface (Figure 6.8 (c)). The production by the snow lance was not sufficient to form ground deposits, but could also be observed to be deviated up-slope.

The same scene viewed from point "VP2" also shows how artificial snow production is intended to fall down-slope of the point of production (Figure 6.8 (d)) during the lulls between gusts, while forward motion is completely negated by wind flow (Figure 6.8 (e)) and snow is deposited behind the production apparatus during gusts.

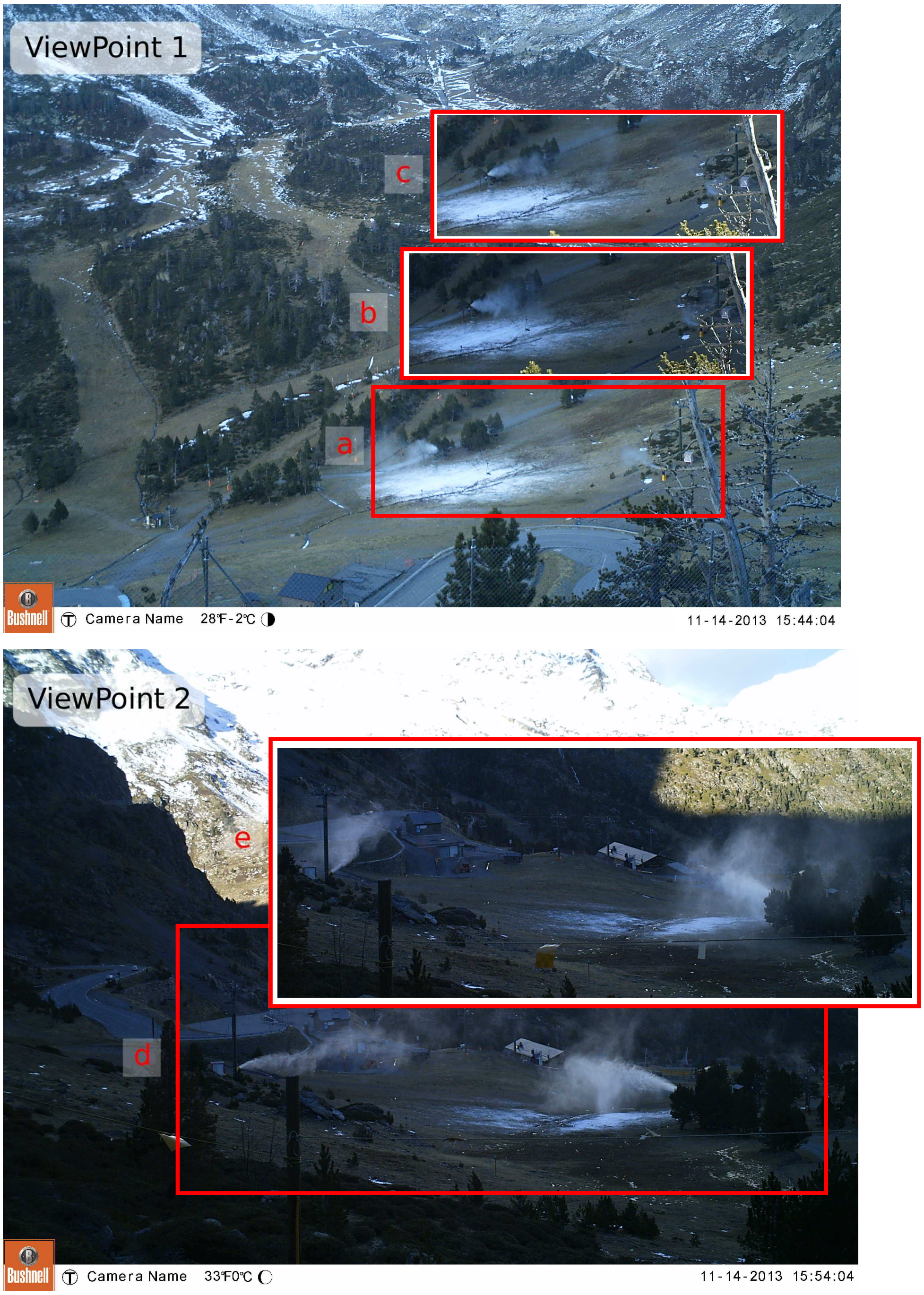


Figure 6.8: Snow cannon and lance observed from Viewpoints 1 and 2. Photo credit: Marc Pons, OBSA, 2013

6.2 High altitude road snowdrift management

Within mountain regions in which the economic activities that take place during winter season -essentially alpine skiing, but also other winter sports such as Nordic skiing, Telemark or snowmobile touring- there arises the problem not only of maintaining ski slopes in a good condition, but also the question of road access to the different points where there activities may take place. Popular memory retains the trace of difficult situations such as that recorded in Andorra on December 8th 1990, when large quantities of tourists became stranded with their vehicles during an unexpectedly intense snow storm. Even during the recent 2013-14 winter, major roads on high ground in the Jerusalem area were closed to traffic during several days (Y. Lappin, 2013) creating inconvenience and potentially dangerous situations for users with little experience of driving over snow-covered roads.

However, experience shows us that not all road segments are equally exposed to snow drift accumulation; local variations in snow transport should be considered when planning the location and capacity of measures to control drifting snow (R.D. Tabler, 1994)[1]. Vehicles tend to lose traction and get stuck in a recurring in the same places, causing traffic build-ups behind them. In a country such as the Principality of Andorra, several spots on the road network are known to exist where the road must be closed to traffic several times each year, such as the access road to the Arcalís ski domain or the main highway to France (e.g. in November 2011, (Canal 324, 2011)). In these situations, the prediction of snowdrift formation on high-altitude roads is of interest for road maintenance tasks and planning of ski resorts.

In this case study, the secondary road from the city of Encamp to Cortals d'Encamp, Principality of Andorra, is taken as an example of high mountain road on which snowdrift formation occurs frequently during winter. The road services both pastoral terrains during summer, and a well-known and much-used funicular terminus station at the top of the valley in winter. The station is complete with a restaurant and connection to the Gran Valira ski domain (Figure 6.10).

A three-dimensional time-dependent computer model of drift formation is presented, that takes into account the effect of natural orographic formations, natural obstacles such as trees, man-made obstacles, the form of the road bed and its adjacent embankments. A 3D air flow model is coupled to a snow transport model and parametrized with time-variable air speed, temperature and snowfall density. The snow transport model takes into account

[1] p. 30

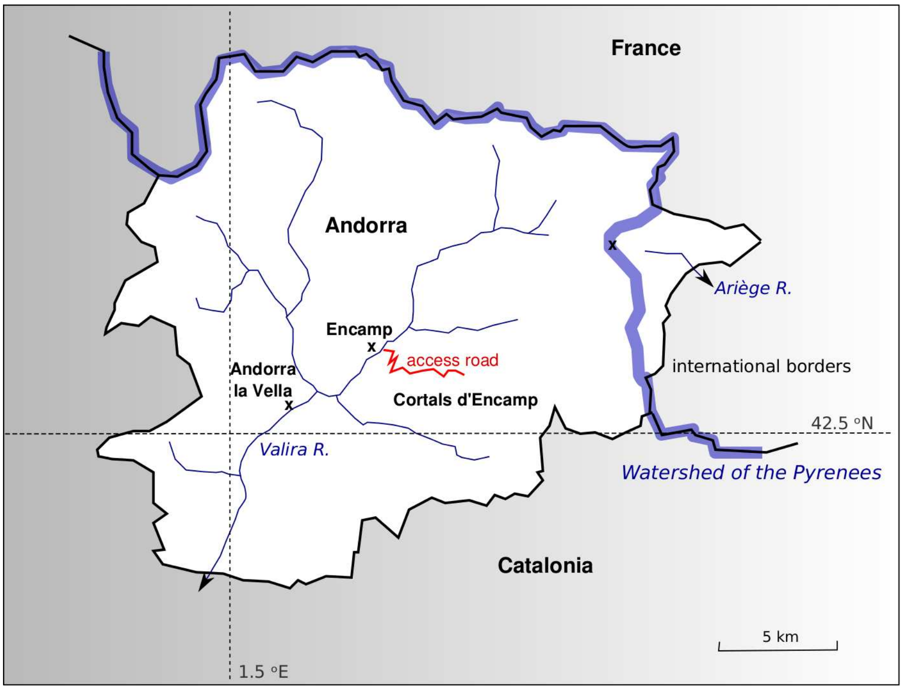


Figure 6.9: Our reference site at Cortals d'Encamp, Principality of Andorra.

both primary and secondary transport. Domain morphology and surface friction alterations due to accumulated snow are tracked and used in successive calculation time steps.

This section has been adapted from previously published material in (A. Ward and J. Jorba, 2014).

6.2.1 Snowdrift formation model

The access road to the Cortals d'Encamp has been built into the mountain side on either side of the Riu dels Cortals valley. The cross-section of the roadbed and surrounding slopes is similar to that of many mountain roads, that are often constructed to one side of a river valley. To the mountain side of the road (Figure 6.10), an abrupt slope arises from the need to cut into the mountain side to make sufficient horizontal space for the road bed. On the river side, the slope downwards to the river bed may be abrupt or gentle according to local terrain configuration.

This is a less-transited road that has its importance both for agricultural activities (in summer) and for tourism (in winter). Being a secondary road, passive means of reducing snow cover are suitable for reducing dependency on the snow-plows that may have to prioritize work on major roads with

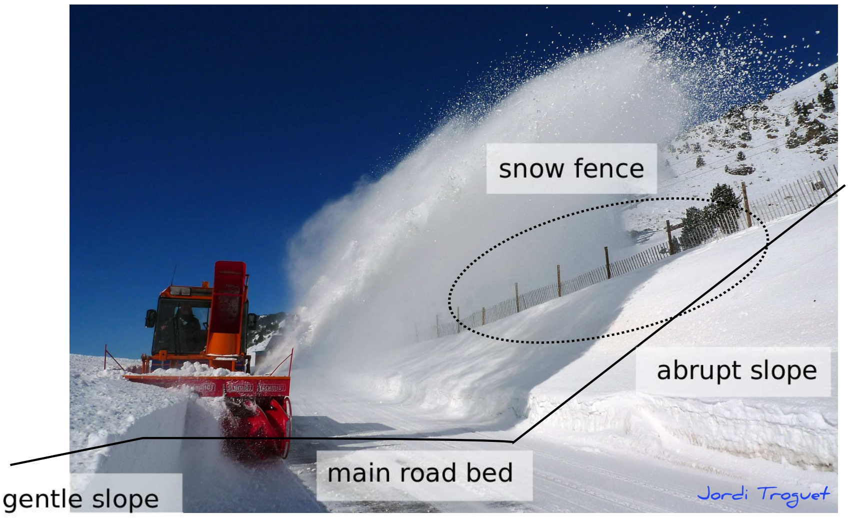


Figure 6.10: Snow bank formation at Cortals d'Encamp, Encamp, Principality Andorra. Photo credit: Jordi Troguet ©2012.

heavy traffic. The presence of segments of snow fence on the slope above the road bed may be noted as a means to reduce drift formation. The fact that snow turbines must be used instead of the more economical blade plows to clear accumulated snow fall is also noticeable, perhaps due to the lower service frequencies than on main roads.

Three different slope profiles have been modeled (Figure 6.11). All have been based on an average slope ratio for the mountain side of 2:1, into which the road-bed has been cut. In all cases, the road bed has typical width of 10 m, and placed in the center of the field at horizontal coordinates [20-30] (all dimensions are in meters).

The model has been initially submitted to a first stage of primary snow transport, during which five partial snowfalls are modeled, each depositing a 20 cm height of fresh snow. After that, ten individual periods of secondary transport have been implemented (dotted red lines), eroding some parts of the initial deposit through eolian action. Wind-driven snow is re-deposited lower down. The final shape of snow cover ends at the position denoted by a continuous red line. Both initial and secondary snow transport takes place under a continuous carrier flow descending the slope, from left to right in the diagrams.

In sub-figure (a), the existing slope has been shown. Snow-fall coming from the top of the slope has deposited a regular layer of snow on the road

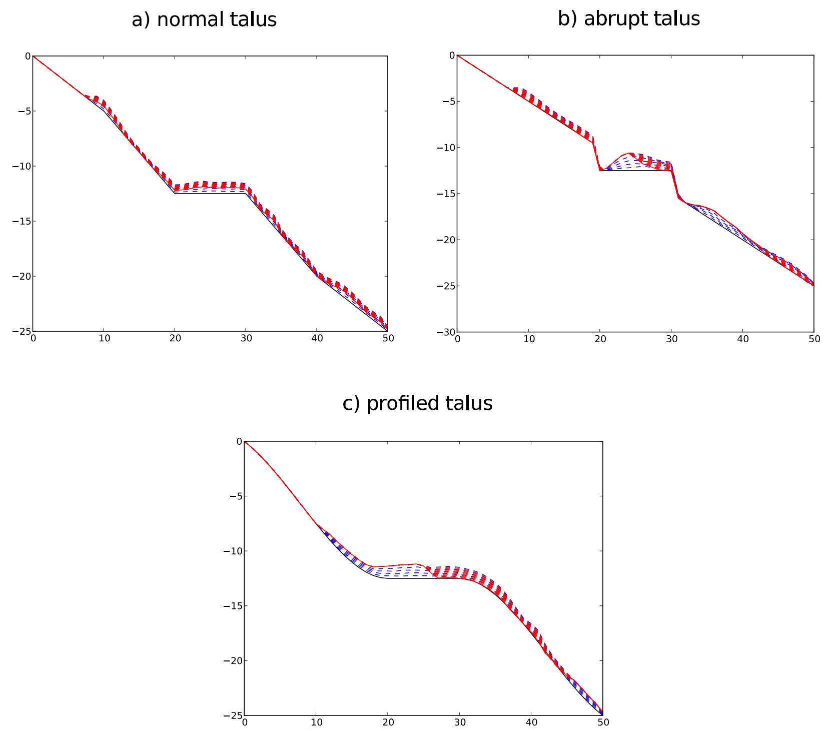


Figure 6.11: Snow deposition and secondary transport with three different slope profiles. Wind direction is from the reader's left.

bed, which has then been eroded by subsequent stronger wind up to half of its depth.

In sub-figure (b), the same situation has been modeled, but with a steeper slope immediately to either side of the road bed. This configuration gives rise to an area of relatively low air flow above and to the left of the road bed. Snow particles have the time to drop and accumulate on the road-bed itself, forming a snow-drift of dimensions up to 3 m. However, the abrupt convex form to the right of the road-bed tends to accelerate airflow in contact with the surface, and subsequent secondary transport is increased eroding the snow-drift and transporting part of its material down-slope and -perhaps- on to the next road curve situated beneath this segment.

Finally, in sub-figure (c), the same road-bed has been placed in between slope sections profiled to create a smooth surface for which the wind to flow upon. Visualization of wind pressure patterns across the road-bed in situation (c) allows us to see in Figure 6.12 that snow deposition in consequent all across the road-bed, due to slower air speed in contact with the surface than in cases (a) and (b). Contrary to expectations, this situation does not permit subsequent wind erosion to evacuate the snow; rather, erosion is confined to the region of the road-bed situated to the right of the figure, leaving a considerable snow-drift occupying the left shoulder and half the road-bed.

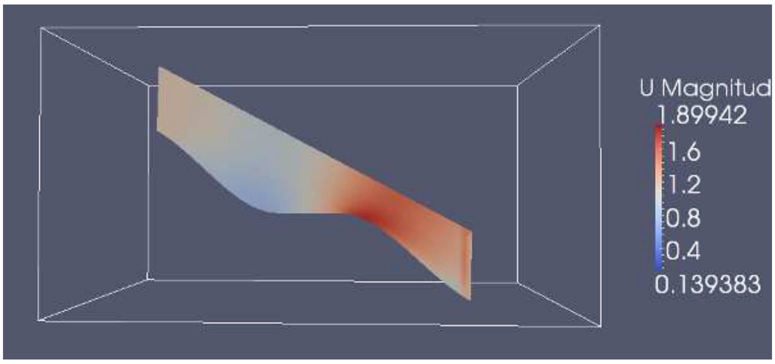


Figure 6.12: Wind speeds calculated by the model in situation (c).

6.2.2 The effect of an immobilized vehicle

When a heavy snowfall is foreseen, orders are often given to keep large vehicles such as trucks and buses off the road. However, forecasts can not always be effective or sufficiently heeded, and such vehicles at times get stuck. Traffic behind them is impeded, and at times also gets stuck, causing difficulties for efficient removal of snow. A further effect that may be taken into consideration is the effect of a large vehicle on air flow around it.

An obstacle of dimensions 3 x 10 x 4 m -corresponding to a single-unit truck or mid-sized bus- was placed on the mountain road studied above. Under the same conditions of initial snow transport and deposition, it was observed that airflow was forced around the front and rear of the immobilized vehicle, and also back up above it. Two areas of slow-moving air are formed up- and down-wind of the vehicle, in blue in Figure 6.13

During initial snowfall, snow-laden air entering the upstream slow-moving area has time to deposit the snow in this region of the road-bed between the vehicle and the slope above. A large snow-drift forms in this space. Once over and behind the obstacle, little further snow deposits are formed on the other side (Figure 6.14). During the secondary transport phase, part of the snow deposited on the vehicle itself may be eroded and transported away from the road, however the drift between the vehicle and the upwind slope is protected by the zone of low wind-speeds and so remains in place.

If there should arise such a situation where a large vehicle should be temporarily left on the open road during a heavy snowfall, it would seem

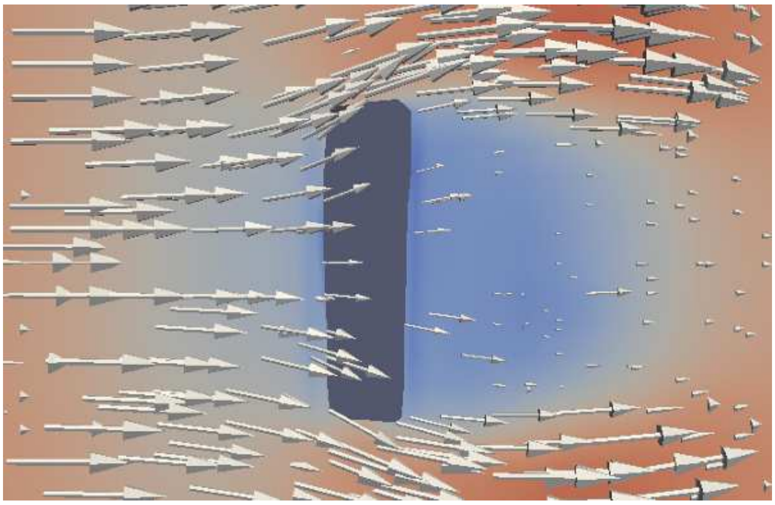


Figure 6.13: Airflow around a large vehicle, immobilized along the road.

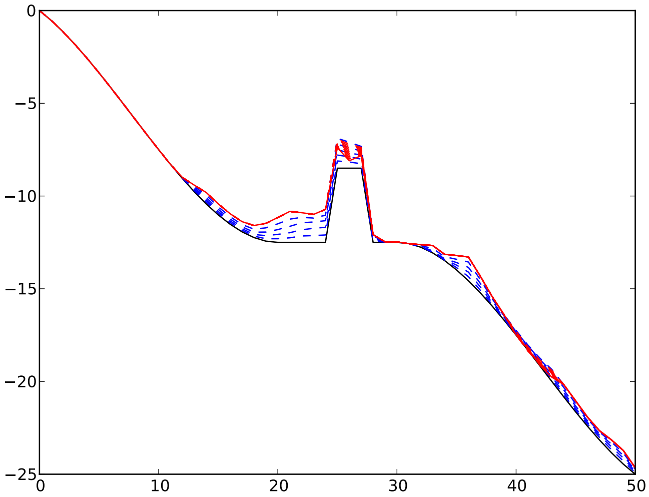


Figure 6.14: Formation of a snow drift between an immobilized vehicle and the slope above.

preferable to place the vehicle as far left as possible, given the circumstances, closer to the mountain slope and upwind of the road-bed. In this place, it could contribute to reducing the load of snow on the road-bed itself. However, it is foreseeable that removal of the vehicle itself would be hampered by the accumulated snow, requiring mechanical removal before initiating vehicle extraction.

6.2.3 Model validation

Model validation is complex since on open roads there exist interactions between single-event snow deposition and:

- Ordinary traffic circulating along the road.
- Active snow management, such as snow-plows.
- Previously deposed snow layers.

For this reason, it seems appropriate to valid the computer model by seeking a road that is not in use during the winter season. A suitable example seems to be the Coll d'Ordino road in Andorra (long. 1.53665, lat. 42.54706). This road has several advantages for the needs of this project:

1. The road is temporarily closed to wheeled traffic during the winter months (October to April/May). Snow is allowed to accumulate, with no plowing.
2. The road is still accessible, by other means such as mountain skis or snowmobile. Traffic with these characteristics have less effect on the deposited snow, while allowing the investigator greater ease of access to the area.
3. Several curves on the road are exposed to the West. They are thus exposed to incident snow-bearing winds (mostly from the NW), and can be expected to reproduce the behavior described by the computer model. At the same time, they are sufficiently exposed to solar radiation for snowfall to melt between major snowfalls.

It is thus proposed that a good location may be found along this road than can then be instrumented using snow pickets and snow levels logged using either camera techniques or personal observation.

6.2.4 Conclusions

We can say that the analysis of carrier flow models show us that slope profiles may have a strong influence both on the initial deposition of snow, and on the subsequent partial erosion and formation of snow-drifts. It also documents the effect of stationary vehicles on the road-bed on snow accumulation and the initial formation of snowbanks.

6.3 Wind turbine implantation

Some factors leading to the creation of negative anthropic influences in mountain regions include densely populated areas such as Grenoble (France), Mexico D.C. (Mexico), or Andorra la Vella (Andorra). In such areas, the presence of large human populations place supplementary loads on the natural habitat, including water consumption on the one hand, and waste production and contamination on the other. In the context of minimizing this influence, some interest has been expressed for the use of clean energies to reduce air contamination in mountain valleys, with applications either for heating requirements (Diari d'Andorra, 2013) or as alternative transport power sources (El Periòdic d'Andorra, 2014). However, harvesting energy with wind turbines presents new challenges when machines such as Horizontal-Axis Wind Turbines (HAWT) are to be installed in mountain terrain.

The study of such installations has a relationship with this thesis, from two points of view.

1. In the first place, when considering HAWT installation at high altitudes, there arose a concern about the possibility of ice formation on the blades, as a combination of direct ice deposition and interaction with snowfall. Ice blocks formed on the blades while stopped or at low speed may come unstuck and be projected to certain distances, thus becoming a hazard for people or structures in their vicinity (Figure 6.15).

   During the construction of the initial HAWT computer model, it became apparent that other phenomena would also be present, reducing HAWT energy production, increasing internal stresses and lower life expectancy. The original concern then became secondary to the mere viability of electrical production through this means.

2. The second relationship was as a means of validation of the first part of the snow transport model, when only the carried fluid (airflow) is considered. During the construction of the circulation model, interesting points were made regarding the formation of the ground layer and the importance of the terrain surface in relationship to slip conditions at the interface between the terrain and airflow.

As discussed in (A. Ward and J. Jorba, 2013) existing computer models of wind farms tend to hypothesize an ideal situation in which the wind turbine is situated on a flat surface and within a fluid structure that can be approximated using the Prandtl log profile law:

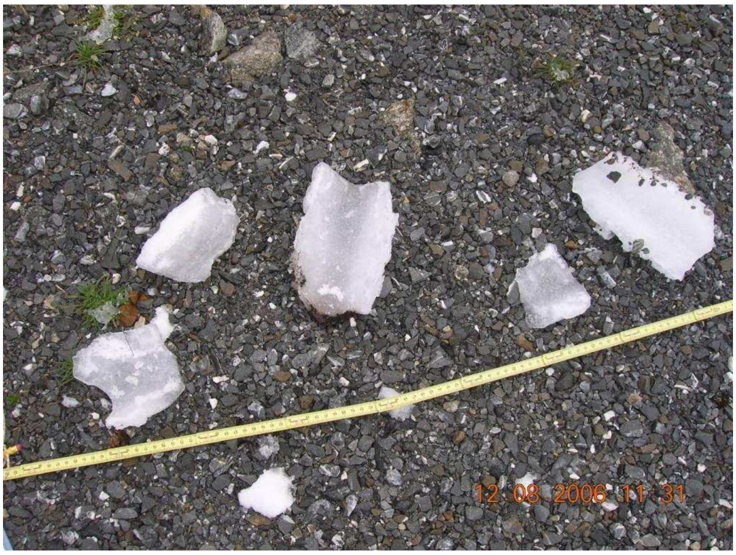


Figure 6.15: Pieces of ice thrown from a HAWT turbine blade. Photo credit: Alpine Test Site Gütsch, Federal Office of Meteorology and Climatology, Switzerland.

$$\frac{v(z)}{v_0} \approx \frac{1}{\kappa} \cdot ln\left(\frac{z}{z_0}\right) \tag{6.2}$$

In reality, however, airflow above complex terrain is more complex, both in front of and specially behind the obstacle to airflow (G.T. Bitsuamlak *et al.*, 2004). For this reason, the computer model developed in this work has been adapted to solve the RANS equations for an incompressible flow above a mountain ridge on which a HAWT turbine was to be installed.

Additionally, a theoretical Blade-Element Model (BEM) of such a turbine was implemented. This is model of a wind turbine blade that predicts yaw and tilt torque on the complete HAWT assembly, as well as overall power output under simulated field conditions. The influence of the HAWT on wind circulation is neglected; the two models are thus one-way coupled. By combining both models it is shown that a wind turbine placed at such a location may receive less dense incoming air with repercussions on power output, as well as other unforeseen effects due to airflow negative vertical incidence such as the appearance of harmonic cyclic torque both in the turbine main shaft and nacelle yaw-control system.

6.3.1 Material and methods

The target is to build a wind circulation computer model for an orographically complicated terrain. Such terrain presents a challenge in that fluid circulation must be modeled for larger mountain valleys with lengths of several km, but also in sufficient detail to reproduce well smaller scales in the vicinity of the HAWT itself.

As an actual case study, the methodology developed has being put into practical use by site engineers at the site of Pic del Maià (2.614 m a.s.l.) (Figure 6.16), for Forces Elèctriques d'Andorra (FEDA, the main producer and distributor of electricity in Andorra) (Forces Elèctriques d'Andorra, 2011) to help plan this future HAWT installation. This site was chosen on the watershed of the Pyrenees mountain range, to benefit from regular North–South cross-range dominant winds, and so represents an extreme case of ridge-mounted HAWT installation.

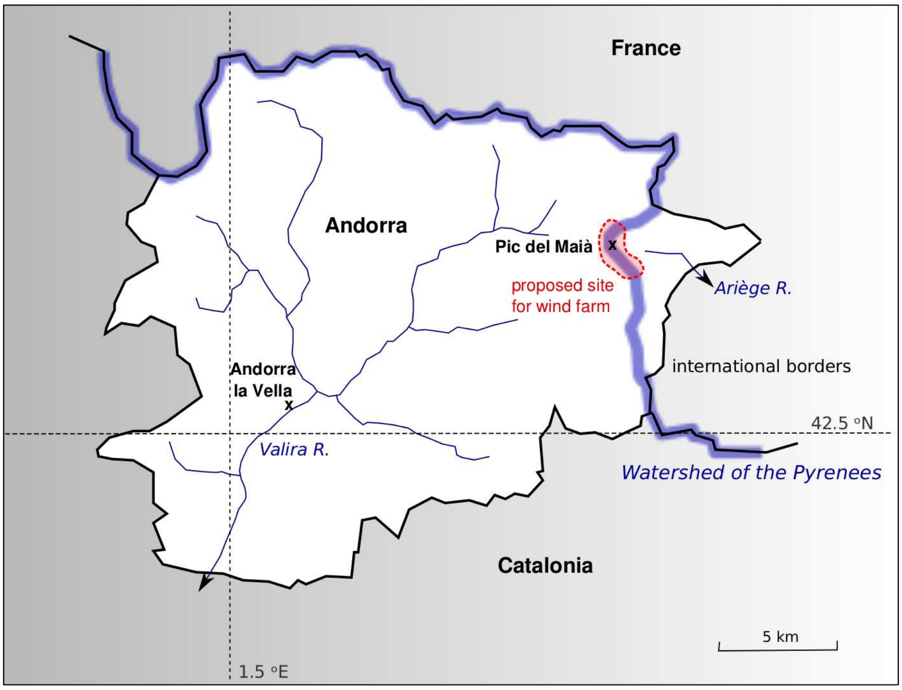


Figure 6.16: Our reference site at Pic del Maià, Encamp, Principality of Andorra.

On the other hand, the geographic situations considered often present transportation issues due to narrow roads with curves, and for this reason a maximum blade length of 25 m is hypothesized. Larger physical HAWT sizes would not be transportable by road since individual turbine blades must be shipped as a complete assembly. As regards transport by air, shipping of larger sizes of HAWT blades may be considered uneconomical. Few commercial helicopter types have sufficient load ratings at circa 2500 m altitude to safely transport blade elements that, at 25 m length, already weigh in excess of 4.3 metric tonnes each (Emerging Technology Corporation, 2015). For this reason the HAWT BEM model was constructed based on an existing type of HAWT, the Enercon E48 (Enercon, 2010). This class of triblade HAWT with 48 m rotor diameter and 800 kW maximum power output is small by modern standards. But it is also at the larger end of the range of machines that could realistically be transported up to installation sites.

Topographical data was taken from the Space Shuttle Radar Topography Mission (T.G. Farr *et al.*, 2007). This data set presents 1 x 1 degree (latitude x longitude) height grids, which at our latitudes give a horizontal resolution of 68 m in the East–West direction, and 93 m North–South. This is the best resolution available at low cost without resorting to interpolation. In order to make efficient use of available computing power, nested meshes on three levels were used as in Case 1 above, with successive horizontal mesh sizes of 81.981 x 111.194, 34.130 x 46.292 and 6.826 x 9.258 km. Each successive mesh has shared points (both on the border and within the mesh volume) with the previous mesh so as to ensure flow continuity.

So as not to produce low-quality (deformed) mesh elements, vertical mesh resolution is limited by horizontal resolution. For this reason, all meshes have a vertical resolution of 25 m, thus giving us several vertical data points in the vicinity of the HAWT.

Since high-altitude flows are considered less expensive to model from a computation standpoint, a uniform vertical resolution is used from the level of the terrain within the 2–3 m range above sea level (a.s.l.) up to an upper limit of 7.5 m. Based on previous work (A. Ward and J. Jorba, 2011), a mesh optimization technique was implemented to treat the problem of mesh element deformation that appears in complex terrain forms.

Air pressure, temperature and density components are calculated using the ISA Standard Atmosphere Model derived from (Talay, 1975) and (NASA, 1976) as pressure input. The temperature gradient (standard: 0.0065 K/m, local: 0.00609 K/m) is adapted to local conditions through analysis of nearby decade-long instrumental temperature records (Forces Elèctriques d'Andorra, 2012), and used to adjust local air density.

The boundary layer was modeled in two ways. In a first approximation, a simple slip boundary condition was used, maintaining zero normal gradient at the ground interface. A second method was then implemented, modeling surface roughness with the k_s method. Since the target was high-altitude HAWT installations, typical ground surfaces in the vicinity of the HAWT are either short grass typical of the alpine mountain stage, or snow and ice. In either case, surface roughness lengths in the $0.002 - 0.5$ m range are to be expected (P. Singh and V.P. Singh, 2001). $z_0 = 0.01$ m was chosen as reference value, giving roughness height $k_s = 30 \cdot z_0 = 0.3$ m and roughness constant $C_s = 9..793/k_s \cdot z_0 = 0.33$ (dimensionless) as suggested by (B. Blocken *et al.*, 2007).

As for the second computer model for the HAWT itself, many HAWT modeling techniques have been proposed. The earliest Blade Element Model (H. Glauert, 1935) has been criticized for its relatively poor comparison with experimental data (P.T. Smulders, G. Lenssen and H. vanLeeuwen, 1981; C.G. Helmis *et al.*, 1995). Modern techniques such as the vortex method (A.A. Afjeh and T.G. Keith, 1986), potential methods (L. Bermúdez, A. Velázquez and A. Matesanz, 2000), particle methods (S. Voutsinas, S.G. Belessis and K.G. Rados, 1994) or, more recently, a lattice method proposed by (S.D. Pesmajoglou and J.M.R. Graham, 2000), take into account wake structure to better approximate forces acting on each turbine blade segment.

Combined methods have also been proposed, for example (J.T. Conway, 2002; I. Dobrev, F. Massouh and M. Rapin, 2007). These methods present the advantage of modeling the wind turbine not as an isolated element, but as one more aerodynamic influence within a global flow, and may be seen as a modern continuation of the actuator disk model proposed originally by (R.E. Froude, 1889) and adapted to wind turbines by (A. Betz, 1920).

A related development has been the actuator line method (J.N. Sorensen and Wen Z.S., 2002) and object of recent studies such as (N. Troldborg, 2008), or the actuator surface (C. Masson and C.S. Watters, 2008).

Methods such as finite elements and multi-body systems to take structure deformation during operation into account -see review of the field in (P. Passon and M. Kühn, 2005)- have been in use since the late 2000s to model HAWTs themselves and immediately surrounding airflow. However, these models suppose a regular environment for the HAWT in order to simplify calculations. We cannot do so, and for this reason think that the blade element model is the most reasonable choice for situations that present a complex input flow such as a high-mountain ridge.

The BEM model for the Enercon E48 turbine has been implemented for an estimated hub height of 50m, minimum and maximum blade heights of 25 and 75 m above ground, and constant angular speed $\omega = 1\ \mathrm{rad} \cdot \mathrm{s}^{-1}$ (Figure 6.17). Hub radius has been taken at 2 m, while individual blades have been broken up into 2.3 m segments for analysis. A standard Clark-Y airfoil profile was chosen to give lift and drag coefficients. Torque and power outputs were calculated each 1^{o} value of rotations position θ for each blade, then combined to obtain the corresponding values for all three blades placed at relative positions separated by 120°.

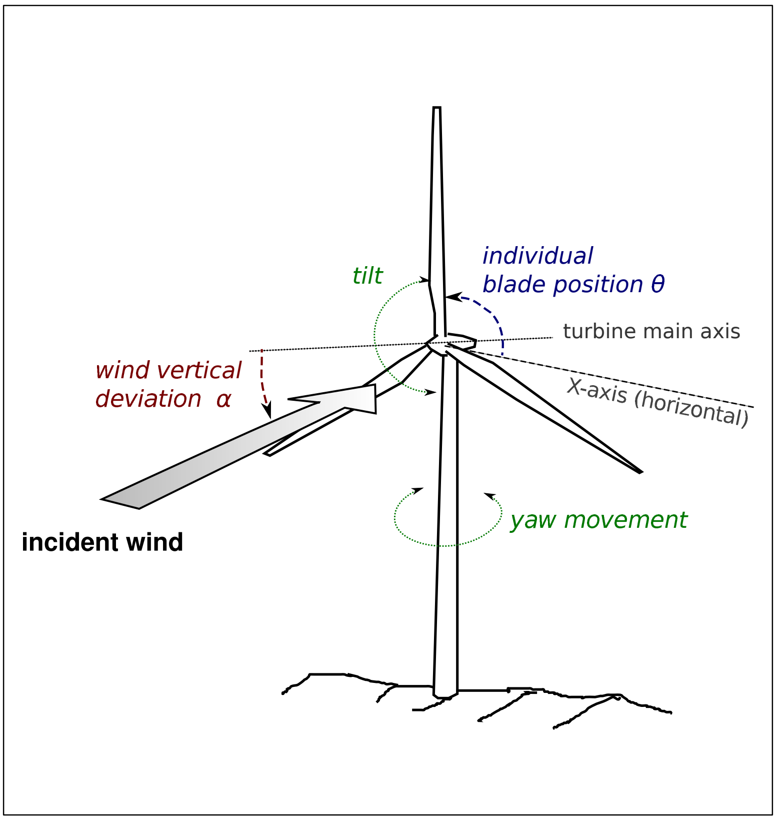


Figure 6.17: HAWT structure and angles.

6.3.2 Circulation model output

An analysis of wind speed and direction data from years 2000 to 2006 from the closest meteorological station at Envalira (2510 m a.s.l.; data available on-line (Govern d'Andorra, 2012) shows that a number of synoptic situations were repeated in this region with high frequencies. The most frequent wind directions and speeds are given in Table 6.1.

The wind circulation computer model has been run for these synoptic situations. A sample output of the wind circulation model is reproduced in Figure 6.18, for a scenario in which the regional wind is a steady 12 $\mathrm{m \cdot s^{-1}}$ from due West. The model reproduces correctly various phenomena that can be measured on the terrain, including:

- Wind speed slowing down considerably in recessed valleys that are presented cross-wise to the regional wind. Some turbulence is seen, but

Table 6.1: Synoptic situations in Andorra

Quadrant	Frequency of appearance (%)	Average airspeed at Envalira (m/s)
N	12.7	4.1
W	14.8	6.2
SW	9.4	9.0

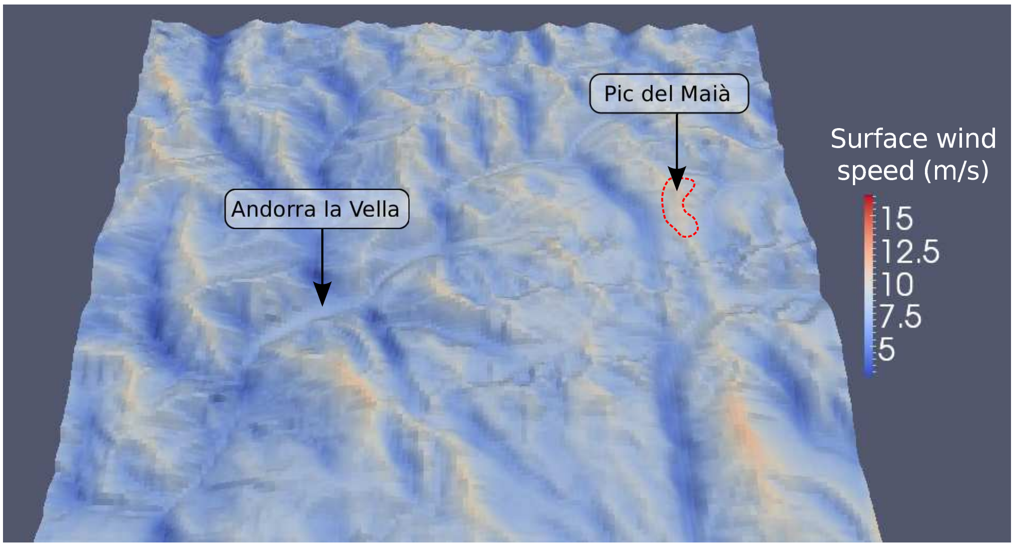


Figure 6.18: Wind circulation model for Pic del Maià, wind from West at 12 $\mathrm{m \cdot s^{-1}}$.

incoming regional winds basically by-pass such valleys.

- Wind speed being maintained along valleys that are a small angle to regional wind. Incident regional wind thus has a longer distance along which to penetrate such valleys, getting down to ground level with relative ease. This can be seen quite well at Andorra la Vella (the capital city of the Principality), where the main valley is oriented approximately WSW-ENE. Incident westerly winds such as this scenario provoke surface wind speeds comparable to regional average.

  However, with winds from due north or due south, surface speed is limited to 0–4 $\mathrm{m \cdot s^{-1}}$ ground layer jitter. This is confirmed by data from meteorological station Roc de Sant Pere near Andorra la Vella, from the same source, where measurements gave speeds of 3–4 $\mathrm{m \cdot s^{-1}}$ (wind direction 330°) in these regional conditions.

- Wind speed below regional average at meteorologic stations situated slightly below the ridges, such as Envalira. With 12 $\mathrm{m \cdot s^{-1}}$ regional wind, our model produced 7 $\mathrm{m \cdot s^{-1}}$ at this point, comparable to the expected value of 6.2 $\mathrm{m \cdot s^{-1}}$.

- Wind speed increasing above regional average along crests situated cross-wise to incident wind. With a 12 $\mathrm{m \cdot s^{-1}}$ regional wind, we observed peaks of up to 16 $\mathrm{m \cdot s^{-1}}$ across such crests.

6.3.3 HAWT model output

Air density may be of concern when designing a high-altitude HAWT installation. In order to compare a similar Enercon E48 model HAWT placed at sea level and at Pic del Maià, the hypothesis of standard conditions (15 C temperature and 101.32 kPa pressure) at sea-level. Lower air pressures tend to give lower air densities, but are slightly offset by lower temperatures. As a combination of both effects, air density is 23.5% lower at Pic del Maià than at sea-level, thus degrading power output. Instead of the 730 kW output foreseen at reference conditions (altitude 0 m a.s.l., wind speed 12 $\mathrm{m \cdot s^{-1}}$ and 0 C vertical and lateral incidence), we now obtain only 560 kW (Figure 6.19).

As is typical in a well equilibrated HAWT, yaw and tilt torque values are small and power output does not vary for different blade positions (angle θ).

The output of the regional wind circulation model gave us a more precise view of circulation at the proposed installation site (Figure 6.20). An effect to be taken into account are differences in wind speed at varying levels above ground. In well-developed winds on flat terrain, the combination of regular surface friction with a long distance over which wind can reach a steady state tends to give surface layer wind speeds a power law structure along the vertical axis. This is not seen in our wind circulation computer model. Instead, above mountain crests it is observed that winds within the first 500 m may be strongly accelerated, to speeds well above regional average.

In the vicinity of ground level, wind speeds increase with height giving rise to wind shear. Thus, with a 12 $\mathrm{m \cdot s^{-1}}$ regional wind speed, we obtain 12.4 $\mathrm{m \cdot s^{-1}}$ at 25 m (lower HAWT blade tip position) but 12.8 $\mathrm{m \cdot s^{-1}}$ at 75 m (higher tip position), a difference of 3%. When taken into account in the BEM model, this produces a cyclic variation both in torque and in power output.

Finally, the wind circulation computer model shows us that incident airflow at the proposed HAWT would contain not only a horizontal component, but also a vertical one. To an observer situated at the top of the turbine nacelle, incident wind would actually appear to come from below the hori-

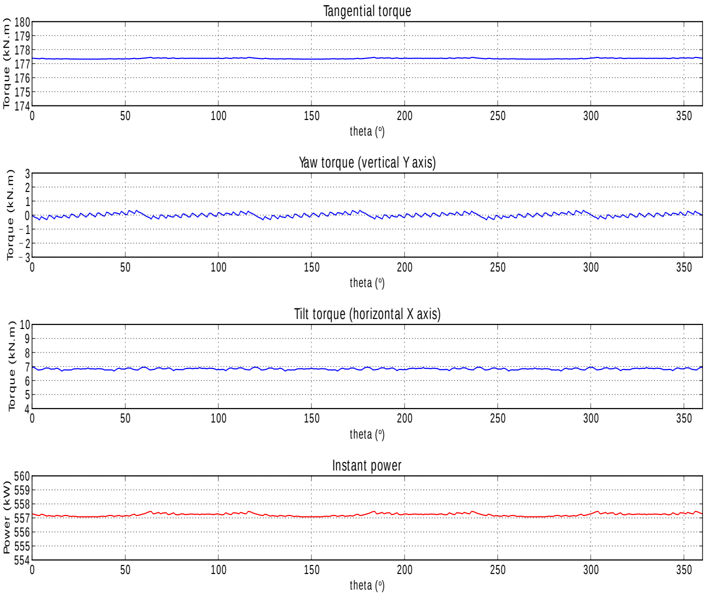


Figure 6.19: Reference HAWT torque and power levels during one cycle. Incident wind horizontal produces low values of yaw torque, though tilt torque pushes back on the rotor assembly with 6 kN.m .

zon. This is a factor that naturally would not be seen in a flatland HAWT installation.

Unfortunately, the experimental setup was not able to give us the third, vertical, component of wind speed, since instrumentation consists only of the classical vertical-axis cup anemometer and wind-vane setup.

Results (Figure 6.21) predict a further decrease of power output. When both lower air density and a vertical incidence of -11.5° are factored in, power output is in continuous oscillation within the 547–560 kW range, which may be compared to the 730 kW nominal output in a steady horizontal airflow at the same speed.

More importantly, these results also foresee the appearance of vertical wind shear. Our HAWT model shows that this in its turn produces high cyclic values of yaw torque, as predicted by (A.C. Hansen, 1992). Such cyclic variations are associated by Hansen and others with increased material fatigue both in the blades and the rotor-head assembly points.

This formation of yaw torque is more preoccupying than the effects of horizontal wind shear. The yaw control mechanism of the turbine is subjected to several thousand kN.m of torque, and must maintain not only a static

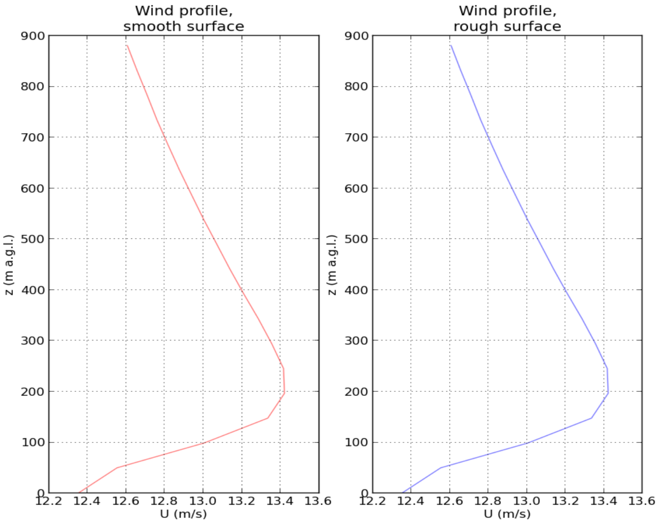


Figure 6.20: Wind profile at proposed installation site. Left diagram: model output with smooth surface boundary layer model. Right diagram: output with rough (k_s model) boundary layer.

equilibrium but furthermore move the turbine head back into the wind when wind direction varies. However, such high levels of yaw are not seen during wind turbine operation under standard conditions with horizontal incident airflow.

Provision must be made to support such stresses within the HAWT head assembly and yaw control mechanism, which can be predicted to increase installation and maintenance costs. On the other hand, decreased power output can diminish returns both directly, and through the losses incurred in isolating the electricity distribution grid from T=3 harmonic signals that are known to be specially destructive to electrical equipment.

6.3.4 Model validation

Unfortunately, validating both the wind circulation model and the VAWT computer model require a substantial investment in material.

The wind circulation model would require data from several locations regard wind speed and direction, at different height levels above the ground. To take into account vertical airflow components, wind direction needs to be recorded non only within the horizontal plane, but also vertically. Most commercial anemometers (both cup and vane-style) are not designed to do

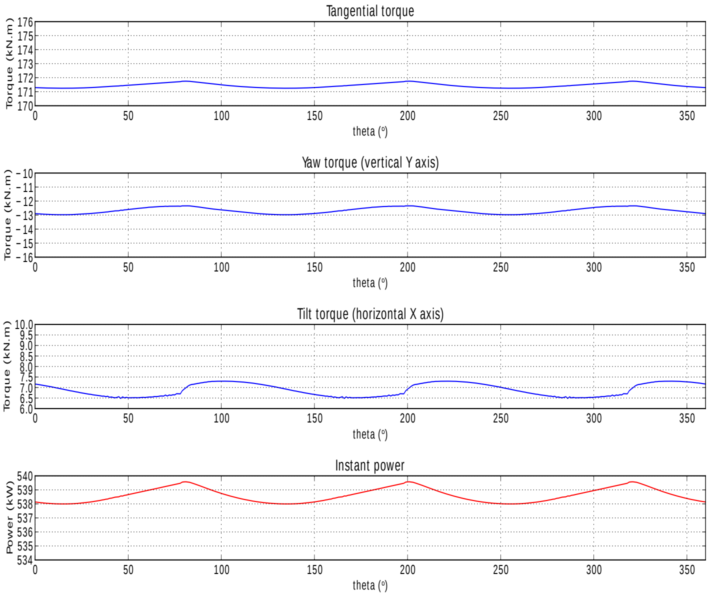


Figure 6.21: HAWT torque and power levels during one cycle. Incident wind at −11.5° below horizon. In this case tilt torque values of 6 kN.m are observed, though with some fluctuation. However, yaw torque exceeds 12 kN.m .

this, so special apparatus may have to be designed.

The locations chosen (mountain ridges) must also be equipped with structures high enough to reach typical HAWT operating heights. As a general guide, blade tip maximal height ranges from 75 m to 100 m above ground level for the Enercon E48 model used in the previous discussion. The FEDA tower data used in validating our model from (Forces Elèctriques d'Andorra, 2011) had been constructed specifically to study wind patterns before installing a HAWT in this area, but had only two data registration points at 25 m and at 50 m. Few other structures of suitable height may be foreseen, unless an actual HAWT is installed.

For this reason, a viable option to validate both computer models may be using existing HAWT installation in other regions. For example, the Swiss commercial company MeteoTest installed a (smaller) Enercon E30 model HAWT at Alpine Test Site Guetsch, Switzerland. This project was active from years 2005 to 2008, with the target of instrumenting the HAWT in order to detect and study ice formation on the turbine blades (Alpine Test Site Guetsch, 2005). Although the original project has been terminated, it may be possible to re-use either existing material, or data obtained during the period in which it was in operation.

6.4 Chapter conclusions

Three cases are presented in this chapter, all three from sites in the Principality of Andorra. The first concerns the installation of artificial snow producing apparatus on ski slopes. Using nested meshes, three concentric models of wind circulation are prepared. Regional air flow is used as to parametrize boundary conditions in the first; successive meshes use the pressure and wind flow fields from the preceding mesh for the same purpose.

In a specific test in which a snow cannon and a snow lance were used at the beginning of the 2013-14 winter season, air flow around the production installation was modeled and found to produce a flow rising up-slope along the ski slopes, as was observed in practice on-site. This situation leads to the dispersion of artificially produced snow off-slope, at times behind the snow cannons themselves, and with the consequent loss of the electric power and water consumed during production.

It is thus proposed that preliminary assessment of snow production installations could benefit from computer modeling of at least the more common synoptic weather situations that each proposed site would offer.

In the second case study, both initial primary snow deposit and secondary snow transport around a high altitude road, neighboring slopes and other obstacles such as a vehicle have been modeled. The computer model shows in the first place how the forms given to lateral slopes during road construction channel cross-wise air flow. Roads cut into gentle slopes tend to receive more snow deposits on their surface, though if winds are strong enough after the main snowfall episode, secondary transport may partially remove the layer (case (a)). If the cut-out constructed to hold the road is abrupt (case (b)), snow is deposited in an irregular fashion on the road-bed, and a zone of lower air-speeds is formed which does not allow snow evacuation during the phase of erosion. If the slopes on either side of the road are too profiled and hold little asperities, some of the layer deposited will be eroded, but in a partial manner and leaving a snow-drift in place on the inner (mountain) side of the way. Road slope planning is thus important to help evacuate part of the snow layer through passive means.

To complete this scenario, it can be how a large vehicle that remains in place during the snowfall and secondary snow transport affects the formation of snow-drifts on the road-bed. It is clear that these effects may also be expected induced by the presence of road-side constructions such as buildings or containment walls.

On the other hand, little is known about the use of fixed artificial elements such as those presented in (R. M. Lang and G.L. Blaisdell, 1998) to create vortices and channel air-flow into areas of interest. For this reason, it is suggested that further studies, both physical and computer-based, are required to investigate in what form and under which circumstances such elements could help reduce snowdrifts in a passive manner, thus also reducing workload on road maintenance crews and the energy dependency of the country as a whole.

Finally, a study of the implantation of a Horizontal Axis Wind Turbine was performed. The wind circulation model was coupled with a Beam Element Model of turbine blades. Interesting results include the appearance of wind-shear along high mountain ridges and at the proposed wind turbine installation side, as well as a non-null vertical component to incident airflow. Wind turbine power output is reduced by these factors and by decreasing air density to values fluctuating within the 547–560 kW range, down from 730 kW nominal output at 12 $\mathrm{m \cdot s^{-1}}$ wind speed. Moreover, large values of cyclic torque appear that affect not only the turbine tower structure, but also its pointing mechanism that directs the head assembly permanently into the wind.

Under these circumstances, it seems reasonable to propose that deeper insight into the economic aspects of HAWT installation and maintenance on high mountain ridges is necessary to ensure project long-term feasibility and financial returns.

It may be noted that the applicability of the techniques described are limited only by the availability of topographical data at a suitable scale for the region under study, and by the fact that an initial wind pattern at regional scale must be used as input.

In all three situations that have been studied here, these two main points concentrated most effort during model construction:

1. Terrain shape modeling.

2. Obtaining local air circulation pattern data.

As noted in the introduction of this work, the specific problem situation considered involves mountain terrain. The relatively complex shapes encountered mean that obtaining precise geographical data can be complex. The radar data made public by the Space Shuttle topography mission in (T.G. Farr *et al.*, 2007) helped as lot by giving a general Digital Elevation Model with a resolution of approximately 10 m between data points. This is sufficient to cover regional grids, and some grids at smaller scale. However, even more precise data may be necessary some applications such as the study of individual buildings or pathways between constructions.

Digital Elevation Models similar to that used in this work are available for a large proportion of the globe, and regional wind patterns are studied for the needs of commercial aviation, among others. For this reason, it is foreseen that further applications may be found in geographical areas other than the Pyrenees. Since the model described does not depend on absolute pressure values, the results found may be translated to models at higher or lower altitudes without loss of generality.

On the other hand, local air circulation patterns will be needed to verify the coherence between computer model output and real conditions recorded on the terrain. However, meteorological stations are few and far between. For this reason, when contemplating a computer model for a specific situation, it may be useful to plan for one or several mobile stations, that could at the very least give the investigator the possibility to record wind speed and direction at key points of reference across the area to be modeled.

All software tools used in this project are either readily available from the corresponding projects' web-pages (CFD modeling toolkit), or may be supplied by the author of this thesis. In this sense, it can be said that the main effort involved in building the computer model is not in the domain of computing itself, but pertains rather to the description of the physical environment to be modeled, both from the topographical and meteorologic aspects.

Chapter 7

Concluding notes

7.1 Analysis of contributions

The goals of this thesis concerned specific aspects of using commercially available consumer-grade computing equipment to model snow transport and delivery over a complex orography. They included addressing the challenges posed by modeling airflow over mountainous terrain, bridging the gap between large- and small-scale modeling, and finally handling the complex physical nature of snow particles and their transport.

- **Goal 1** of this thesis was:

  > *Considering how a computer representation (mesh) may be built to accurately represent such complex terrain, and how to optimize the mesh to increase efficiency while solving the mathematical model applied using computers.*

  Modeling the air volume above mountainous terrain was studied with a view to optimizing the discretization (meshing) of this fluid domain. It was pointed out that mesh quality is of importance to make the mathematical description of fluid flow as simple as possible, and thus use the least computational resources when transforming the Navier-Stokes equations into matrix systems and solving then. A measure of mesh quality was proposed, and used to evaluate the convergence of several schemes to optimize the mesh.

  It was found that the Simulated Annealing works best for this purpose, although the physical nature of mountainous terrain is such as that little gain in mesh element quality can be achieved for those elements in direct contact with mesh frontiers: at the lower mesh boundary where the fluid volume is formed by the complex terrain itself, and at the upper boundary where the limit with surrounding air is supposed flat and thus made up of mesh elements that initially already take a regular form.

- **Goal 2** of this thesis was:

  > *To study how a computer model created at regional level may be refined and applied to increasingly smaller areas, making an efficient use of existing CFD toolkits.*

  Existing studies are aimed at modeling either large-scale meteorological flows, or local small-scale models in the immediate proximity of individual buildings or installations. In this work, however, the relationship between regional and local models has been explored through the use of successive grids that cover smaller areas each time. Regional wind movement data has been used as inputs for the largest grid, and the fluid movement calculated by modeling on larger grids has been used as model inputs on smaller grids. Experimental results taken at local level are in agreement with values predicted based on the regional flow in place at that point in time.

  This gives us a tool allowing the production of local models based upon regional models, which is of interest specifically in regions where airflow may vary across small distances due to orographic effects, and in which meteorological stations may be lacking in strategic points. An existing CFD software toolkit was used to implement this scheme in an efficient way from the point of view of solving the model with computational means.

- **Goal 3** of this thesis was:

  > *Identifying which parameters must be taken into account and built into the computer model to correctly represent the relationship between the snow flake and its physical characteristics, and the supporting airflow.*

  The requirement that the computer model handle various physical characteristics of snow flakes during transport has been handled at two levels.

  When studying the coupling between the two fluids, existing general theory concerning two-phase fluids has been applied to the specific case of the transport of snow particles by air as a carrier fluid. It has been found that the flow Stokes number S_t are such that the presence of snow particles tends to enhance the dissipation of turbulent structures

> within the airflow. Individual particles should be in quasi-equilibrium with the carrier fluid and follow its movement closely.
>
> A further finding is that with typical snowfall density values and the relative density of snow particles to air fluid Φ_p, the mixed fluid is at the borderline between a domain in which a simple one-way coupling between the solid and gaseous parts of the mixed fluid suffices, and that in which a two-way coupling needs to be implemented to correctly represent the influence that the presence of snow may have on the carrier airflow.
>
> When implementing a singly coupled model, particle trajectories are governed by airflow and the only parameterizable aspects of their behavior are the mass of snow injected into the system at input level, and the speed of fall (terminal velocity). If a two-way coupling is introduced, the first parameter will have an implication on the relative density of snow per volume unit, and thus also govern the snow-air coupling. Further parametrization, to take into account snow transformation during fall, implies solving the Navier-Stokes equation for energy conservation as well, to take into account heat exchange between the two fluid components.

The results described above have been applied to the construction of a computer model linking airflow and snowfall above a complex orography. This model was built using exclusively existing open-source software, mainly the OpenFoam CFD toolkit.

The use of an open-source fluid dynamics toolkit has been found beneficial for the purposes of adapting an existing software package for the necessities of this specific problem set. The open nature of the software and its associated file structure used to describe the physical domain, the scheme for solving the mathematical model and controlling computer model execution has been of assistance to simplify adaptation to our objectives, and also to decompose the physical domain in order to perform calculations in parallel.

In addition, the fact that the complete software package runs on an efficient operating system (GNU/Linux) that has been optimized for running on many different hardware platforms -including consumer-grade equipment- gives us some leeway as to the hardware used to execute computer models. However, it was found that some congestion does arise when meshing or solving the fluid flow in parallel on a single multi-core CPU. Since disk access requirements are low, the loss of efficiency may be attributed on the one hand to network saturation, and on the other to small (2 x 6 MByte) Level-2 CPU cache sizes. For this reason, although consumer-grade equipment may be envisaged as a hardware platform, it stands to reason some attention must be given to the network and CPU characteristics of the hardware used.

7.2 Applications of findings

Human activities in the higher parts of mountain areas must content, even in Mediterranean countries, with high winds during the year and snowfall in winter. Building a computer model of air circulation and snowfall is one available tool, among others, that may help understand phenomena that may not always be directly studied due to difficulties in accessing and instrumenting areas of interest. A computer model may also be of help in the planning stage, before large investments have been made in buildings or technical installations.

The applications of the findings of this thesis may then be classified as:

1. Preparing for a future construction.
2. Understanding an existing installation.
3. Optimizing planning and taking into account collateral consequences.

7.2.1 Preparing for a future construction

Since wind speeds in the mountains may be higher than at lower altitudes, a computer model performed before physical construction begins can give insight into the airflow and maximum speeds to be expected around man-made constructions, such as:

- roads and associated constructions (bridges, tunnel mouths)
- buildings
- utility installations (towers and poles)
- ski-slope installations (ski lifts, snow-making equipment)

This can help dimension constructive elements correctly.

In Figure 7.1, a computer model is used to generate air streamlines around a building with a concave facade and an external lift column. Air vorticity and low-pressure areas are represented by coloring the streamlines and ground plane respectively, giving a precise idea of how the building interacts with airflow and snowfall. In this example, it can be seen that winds passing

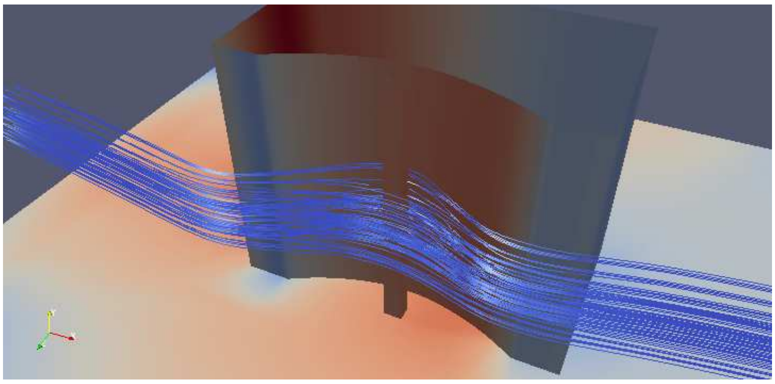


Figure 7.1: Computer model of airflow around a building.

sideways across the building's front facade tend to accelerate airflow, creating turbulent vortices in the immediate area around the main entrance at the foot of the lift column, as well as the front corners of the building where the garage ramp entry point is located.

The building design, as presented by the architects, can be shown to participate in passive snow clearing from before the doors, thus helping reduce maintenance costs in winter.

On the negative side, however, it can also be foreseen that entry and exit on foot will be hindered whenever strong winds are encountered.

The same reasoning may be used when projecting larger constructions such as high altitude roads and bridges, as discussed in Chapter 6.

7.2.2 Understanding an existing installation

Where an installation has already been constructed, the computer model may help understand unexpected effects that would have been difficult to foresee merely from an engineering point of view.

To take an example, the presence of a building is known to alter wind patterns in its immediate vicinity, allowing snowdrifts to form either during primary or secondary transportation (Figure 7.2). Performing a computer simulation of airflow the building during snowfall, taking into account wind patterns in that specific site, can help form a more precise idea of how the drifts are formed.

The computer model may also be use when considering how to remedy such as situation. Alterations may be incorporated into the model, such as placing a barrier at a strategic location, and changes in snow drift formation evaluated before construction is undertaken.

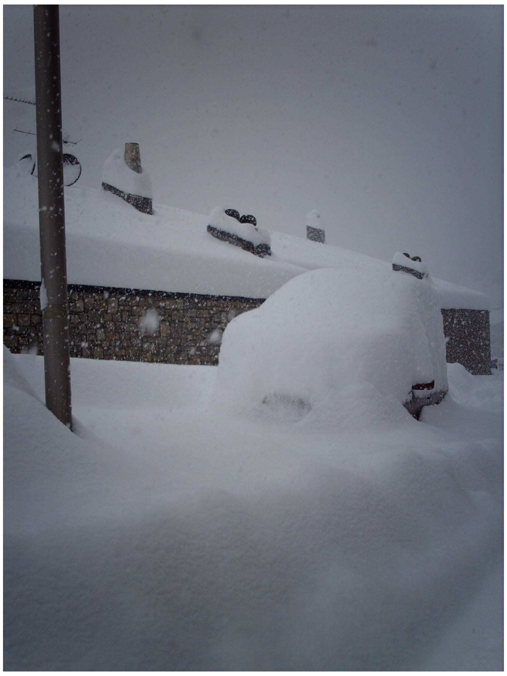


Figure 7.2: Wind-swept snow blocking access to a building.

7.2.3 Optimizing planning and taking into account collateral consequences

Finally, the use of a computer model may help assess the costs of a project, both from a financial point of view, and from the perspective of its effects on the environment.

Ski-slopes are placed in situations where variations may occur in exposure to winds and snow deposition. Although some generalities may be observed, such as the fact that most ski-slopes are constructed on north-facing slopes with low sun coverage, ski-slope planning is still up to a point an artisan process where human specialist intervention is of importance.

When designing the artificial snow-making installation for a new ski-slope, or modifying an existing slope to accept snow canons, the installer may in some cases benefit from the use of mobile snow generating equipment (Figure 7.3). This may be placed at different points and operated, giving a precise idea of the amount and type of snow-making devices needed to outfit the ski slope. Thus, the project may be better dimensioned both from the engineering point of view (electrical and water supplies needed) and for the financial coverage needed.

However, the use of such mobile apparatus still requires adequate supplies

Figure 7.3: Mobile snow canon.

of energy and water, so temporary supply networks must be set up just to experiment the snow-making process in the context of the specific ski-slope that is being designed. This is naturally an expensive and time-consuming process, that can be facilitated from both points of view.

Although the final decision must still be made by the human specialist, a complete computer model of wind flow and snowfall in the area of interest is a tool that may help in the decision-making processes by pointing out portions of the domain that are either most or least suited for the overall objective.

7.3 Open questions for future investigation

During the redaction of this dissertation, several open questions arose that could form the nuclei of further investigation.

7.3.1 Dynamic load balancing during model execution

As seen in Section 4.3.1, balancing the quantity of computation that takes places on each node is necessary in order to optimize scaling when tasks are executed in parallel. However, the complex shape of the physical domain leads to less optimized forms of mesh elements at the limit between the air volume considered and the terrain, and the various distribution of orographical shapes gives different workloads for subdivisions of the mesh, even though they are constructed with similar shapes and equal numbers of elements.

A possible direction of investigation to gain efficiency during parallel computation could be though the use of the timing results of the first iterations of calculation, in order to adapt the workload for each node during successive steps. Thus, the proposed algorithm would be similar to this:

```
1. Prepare mesh

2. Construct initial mesh division, assigning equal
   numbers of mesh elements to each node.

3. Execute one iteration of the PISO method, noting
   the time required by each node.

4. Adjust mesh divisions, assigning supplementary
   mesh elements to nodes that finished in least time.
   Return to step 3.
```

In order to implement such a scheme, it may be necessary to modify slightly the OpenFOAM solver code to record and take into account the time spent in each MPI task. The availability of source code is, in this case, a distinct advantage compared to the use of a commercial CFD product.

7.3.2 Model parameter influence

Several parameters have been used to construct the computer model in this work, such as snowflake density, snowfall density and duration. These may depend, in turn, on other physical parameters such as air temperature or humidity.

To construct the computer model, medium values have been used, that reproduce the conditions found during a average snowfall. However, it is clear that snowfall conditions vary in time, as the winter and spring seasons advance. They also vary from a geographical standpoint, depending on the major physical factors that determine local climate influence.

For this reason, it may be useful to complete this work with further studies that would examine the role of each model parameter and its relationship to model applicability in various scenarios.

7.3.3 Snow transformation during and after transport

A further aspect of snowfall is the transformation of the snow. The complex structure of snowflakes (seen in Chapter 2) make possible not only the phase change between solid and liquid or gaseous forms of water, but also between the different shapes of crystals. Some of these transformations take place over extended periods, such as that of the deposited snow-pack over the course of a winter period. However, some other transformations may take place during a single snowfall, or as the individual particles fall to the ground.

Modeling these transformations imply taking into account the exchange of heat between air and snow particles. The third Navier-Stokes equation, for conservation of energy, must be solved over the domain in parallel with the previous two. However, even if the effects of direct sunlight are not postulated:

- The number of parameters is increased.
- Heat exchange may take place between the fluids in adjacent elementary mesh volumes, or between the border volumes and the exterior of the fluid volume.

  For example, snow at -2 C falling on a slightly warmer ground surface at $+1$ C will, if in sufficient quantity, lower the ground surface temperature while part of the snow itself melts and runs off.
- Heat exchange may also take place between components of the mixed fluid.

  A cold carrier air can cause cold water droplets to ice, forming snow crystals of gradually increasing size.

Modeling these effects from a mathematical standpoint and solving the associated equations presents several challenges, as both the dimensions of the dataset and the computational workload increase. Unless care is taken, the limits of what is practical to achieve with consumer-grade computation equipment may be surpassed.

7.3.4 Dynamic variations during snowfall

In the investigation leading to this thesis, it has been noted that there exists no technical reason why snowfall modeling could be modified to adapt to changing atmospheric conditions during snowfall. During a fall 30 minutes in duration, for example, low wind conditions could take place during the first 10 minutes, followed by 5 minutes of stronger winds and finally 15 minutes of high-density snowfall carried by winds from a different quadrant.

In such cases, the ability to couple existing weather detection information systems with the snowfall model would be of help for public officials, on the terrain handling potentially difficult situations on a blow-by-blow basis. Systems have already been tested that deliver wind speed or precipitation predictions to mobile handsets based on continuously assessed input data (Sean Gallagher, 2012) (Figure 7.4).

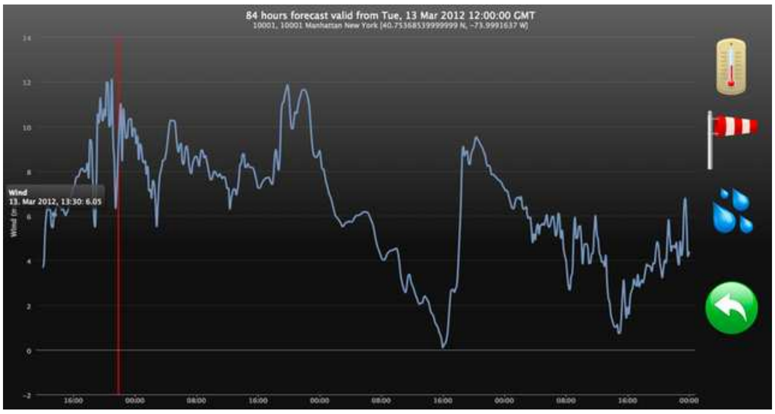

84 hours forecast valid from Tue, 13 Mar 2012 12:00:00 GMT
10001, 10001 Manhattan New York (40.75368539999999 N, -73.9991637 W)
Wind
13. Mar 2012, 13:30: 6.05
Wind (
14
12
10
8
6
4
2
0
-2
16:00
00:00
08:00
16:00
00:00
08:00
16:00
00:00
08:00
16:00
00:00

Figure 7.4: IBM's Deep Thunder: weather prevision feed viewed on a mobile device.

Our proposal concerns similar prediction systems, but adapted to high-altitude situations through the inclusion of snowfall and with a high lateral precision to take into account the varying situations encountered in separate mountain valleys.

However, concerns may arise due to the different data formats used, and also due to the latency with which some data may be processed. Incoming data from automatic stations may suffer disruptions due to the weather itself. Manual inputs may be of help, but will probably not be homogeneous in their timing.

For this reason, such a coupling may require the application of novel techniques in order to generate model output from real-time data inputs, backtracking and adapting calculation as older but still significant data points become available.

Appendix A

Publications associated with this thesis

During the preparation of this thesis, several publications were accepted through a peer-review process in journals and conferences.

A.1 An iterative method for the creation of structured hexahedral meshes over complex orography

In this paper (A. Ward and J. Jorba, 2011), a technique for measuring the quality of hexahedral Cartesian meshes used to model meso-scale atmospheric circulation in 3D was proposed. It is used to verify the progress of a novel method for satisfying the Delaunay criterion for structured hexahedral meshes over complex orography with high gradients and wide gradient variability. Based on a simile with potential energy, the iterative method of mesh smoothing is shown to improve mesh quality with logarithmic convergence. The method is evaluated in a practical application in a specific geographic location.

Applied Mathematics and Computation is a peer-reviewed journal published by Elsevier that "addresses work at the interface between applied mathematics, numerical computation, and applications of systems – oriented ideas to the physical, biological, social, and behavioral sciences, and emphasizes papers of a computational nature focusing on new algorithms, their analysis and numerical results." This journal has an Impact Factor of 1.317 and a 5-year IF of 1.338, as measured by Thomson Reuters' *Journal of Citation Reports 2011.* (SCImago, 2015) reports this journal as of the first quartile (Q1) in Computational Mathematics for year 2011.

A.2 Harmonic buffeting in a high-altitude ridge-mounted triblade Horizontal Axis Wind Turbine

In this paper (A. Ward and J. Jorba, 2013), it is observed that harvesting energy with wind turbines presents new challenges when the turbines are placed on high-altitude mountain ridges.

The results of a computer model solving the Reynolds-Averaged Navier-Stokes equations for incompressible flows above such a ridge are presented in the context of a case study. A theoretical blade-element model of a triblade Horizontal-Axis Wind Turbine (HAWT) was implemented. By combining both models it is shown that a wind turbine placed at such a location may receive less dense incoming air with repercussions on power output, as well as other unforeseen effects due to airflow negative vertical incidence such as the appearance of harmonic cyclic torque both in the turbine main shaft and nacelle yaw-control system.

The *Journal of Wind Engineering and Industrial Aerodynamics*, also published by Elsevier, has a scope that includes the "social and economic impact of wind effects; wind characteristics and structure, local wind environments, wind loads and structural response, diffusion, pollutant dispersion and matter transport, wind effects on building heat loss and ventilation, wind effects on transport systems, wind power generation, and codification of wind effects." It has an Impact Factor of 1.698 and a 5-year IF of 2.124 according to *Journal of Citation Reports 2013.* According to (SCImago, 2015), this journal was in the first quartile (Q1) for Mechanical Engineering and the second quartile (Q2) for Renewable Energy, Sustainability and the Environment for year 2013.

A.3 Planning passive snowdrift reduction on high-altitude roads with lateral obstacles to wind flow

The prediction of snowdrift formation on high-altitude roads is of interest for road maintenance tasks and planning of ski resorts. In this conference presentation (A. Ward and J. Jorba, 2014), a three-dimensional time-dependent computer model of drift formation is presented, that takes into account the effect of natural orographic formations, natural obstacles such as trees, man-made obstacles, the form of the road bed and its adjacent embankments. An open-source CFD tool-kit is used to implement a 3D air flow model, coupled to a snow transport model and parametrized with time-variable air speed, temperature and snowfall density. The snow transport model takes into account both primary and secondary transport. Domain morphology and surface friction alterations due to accumulated snow are tracked and used in successive calculation time steps.

Direct applications of the resulting tool are discussed: modeling and understanding existing road configurations under different wind and snowfall conditions, but also predicting the effects of proposed alterations to the road bed and the elements around it. With careful design of embankments both above and beneath the roadbed, it is foreseen that airflow can be canalized in such a way to induce passive snowdrift removal with no need for mechanical intervention under favorable conditions. Techniques include embankment profiling and the addition of fixed aerodynamic elements in order to increase secondary snow transport away from the roadbed and into the adjoining spaces. Their possible application to a real-world scenario in Andorra is presented.

The *Standing International Road Weather Commission* (SIRWEC) is a forum for the exchange of information relevant to the field of road meteorology, including management, maintenance, road safety, meteorology and environmental protection. The 17th International Road Weather Conference took place in La Massana, Andorra from Jan. 30 to Feb. 1, 2014.

A.4 Other activities

In November 2011, the author acted as a reviewer for the *World Journal of Modelling and Simulation*, at the request of Editor in Chief Dr. Xu Jiuping.

Bibliography

Chapter 1: Introduction and goals

Jennifer W. Burt and Kevin J. Rice, “Not all ski slopes are created equal: disturbance intensity affects ecosystem properties.” *Ecological Applications*, 19(8) (2009). Pages 2242-2253.

Carmen de Jong, “Artificial snow drains mountain resources”, *Environmental research web* (Aug. 8, 2007). URL: http://environmentalresearchweb.org/cws/article/opinion/30703 (retrieved August 4, 2014).

Christian Rixen, Veronika Stoeckli and Walter Ammann, “Does artificial snow production affect soil and vegetation of ski pistes? A review.” *Perspectives in Plant Ecology, Evolution and Systematics*, 5(4) (2003). Pages 219-230.

J. L. Innes, “High-altitude and high-latitude tree growth in relation to past, present and future global climate change.” *The Holocene* 1(2) (1991). Pages 168-173.

Martin Beniston, H. F. Diaz and R. S. Bradley, “Climatic change at high elevation sites: an overview.” *Climatic Change* 36(3-4) (1997). Pages 233-251.

Cécile H. Albert *et al.*, “Land use change and subalpine tree dynamics: colonization of Larix decidua in French subalpine grasslands.” *Journal of Applied Ecology* 45(2) (2008). Pages 659-669.

Jan Esper *et al.*, “Orbital forcing of tree-ring data.” *Nature Climate Change*, 2(12) (2012). Pages 862-866.

Shardul Agrawala, *Climate change in the European Alps: adapting winter tourism and natural hazards management.* Organisation for Economic Co-operation and Development (OECD) (2007).

Marc Pons *et al.* , “Modeling climate change effects on winter ski tourism in Andorra.” *Climate Research* 54 (2012). Pages 197-207.

Herbert Simon, “The New Science of Management Decision.” In proceedings of *The 33rd Conference of the Operational Research Society of New Zealand*, (1960).

T.G. Farr *et al.*, “The shuttle radar topography mission.” *Reviews of Geophysics* 45, RG2004 (2007).

Sigma map server, Institut d’Estudis Andorrans (IEA) and Universitat Autònoma de Barcelona (UAB). URL: http://www.sigma.ad (retrieved September 1, 2009) *In Catalan.*

James D. Meindl, “Beyond Moore’s Law: the interconnect era.” *Computing in Science and Engineering*, 5(1) (2003). Pages 20-24.

Ronald G. Dreslinski *et al.*, “Near-threshold computing: Reclaiming moore’s law through energy efficient integrated circuits.” *Proceedings of the IEEE*, 98(2) (2010). Pages 253-266.

T. Sterling *et al.*, “BEOWULF : A parallel workstation for scientific computation”. *International Conference on Parallel Processing*, 1 (1995). Pages 11-14.

C. Magono and C. W. Lee, “Meteorological Classification of Natural Snow Crystals”, *Journal of the Faculty of Science*, Hokkaido University (1966). Pages 321-335.

Chapter 2: Snow transport and deposition theory

J. Kepler, *De nive sexangula (The six-cornered snowflake)*, Paul Dry Books (1966). ISBN: 1-58988285-7.

W.A. Bentley and W.J.Humphreys, *Snow Crystals*, McGraw-Hill (1931). Reedited Dover (2013). ISBN 0-48615525-0.

U. Nakaya, *Snow Crystals: Natural and Artificial*, Harvard University Press (1954).

D. Ebbing and S.D. Gammon, *General Chemistry, Enhanced Edition*, Cengage Learning (2010). ISBN 0-53849752-1.

P. Pomper, “Lomonosov and the Discovery of the Law of the Conservation of Matter in Chemical Transformations”. *Ambix*, 10(3) (1962). Pages 119–127

D. Dikakis and W. Rider, *High Resolution Methods for Incompressible and Low-Speed Flows*, Springer-Verlag, Berlin (2005). ISBN 3-54022136-0

M. Griebel, T. Dornseifer and T. Neunhoeffer, *Numerical Simulation in Fluid Dynamics*, SIAM Monographs on Mathematical Modeling and Computation, SIAM, Philadelphia (1998). ISBN 0-89871-398-6

A.J. Smits and J-P Dussauge, *Turbulent shear layers in supersonic flow*, Birkhäuser (2006). ISBN 0-38726-140-0

P.J. Linstrom and W.G. Mallard, Eds., *NIST Chemistry WebBook*, NIST Standard Reference Database Number 69, National Institute of Standards and Technology, Gaithersburg MD. URL: http://webbook.nist.gov (retrieved April 24, 2014).

J.H. Ferziger and N. Perić, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin (2002). ISBN 3-54042074-6

T.A. Talay, "Introduction to the Aerodynamics of Flight", NASA SP-367, National Aeronautics and Space Administration, Washington D.C. (1975). Pages 6-9.

U.S. Standard Atmosphere, 1976, NASA-TM-X-74339, National Aeronautics and Space Administration, Washington D.C. (1976).

M. Takeuchi, "Vertical profile and horizontal increase of drift snow transport", *J. Glaciol*, 26 (1980).

T. Uematsu et al., "Three-dimensional numerical simulation of snowdrift", *Cold regions science and technology*, 20.1 (1991). Pages 65-73.

Y. Tominaga et al., "Development of system for predicting snow distribution in built-up environment", *The Fifth International Symposium on Computational Wind Engineering* (CWE2010), Chapel Hill, North Carolina, USA (May 2010).

J.W. Pomeroy et al., "The Prarie Blowing Snow Model: characteristics, validation, operation", *J. Hydrology*, 144 (1993). Pages 165-192.

P.A. Sundsbø, "Numerical simulations of wind deflection fins to control snow accumulation in building steps", *J. Wind Eng. Ind. Aero.*, 74–76 (1998). Pages 543–552.

J.H.M. Beyers et al., “Numerical simulation of three-dimensional, transient snow drifting around a cube”, *J. Wind Eng. Ind. Aero.*, 92 (2004). Pages 725-747.

G.E. Liston and M. Sturm, “A snow-transport model for complex terrain.”, *J. Glaciol.*, 44 (1998). Pages 498-516.

E.M. Greene, *Simulation of Alpine snow distributions in the Northern Colorado Rocky Mountains using a numerical snow-transport model*. Master’s thesis, Colorado State University (1999).

M. Lehning *et al.*, “ALPINE3D: A detailed model of mountain surface processes and its application to snow hydrology”, *Hydrol. Process.*, 20 (2006). Pages 2111-2128.

W. Schmidt, “Eine unmittelbare Bestimmung des Fallgeschwindigkeit von Regentropfen”, *Sitz. Akad. Wiss. Wien*, 118(2A) (1909). Pages 71-84.

A.F. Spilhaus, “Raindrop size, shape and falling speed”, *J. Meteor.*, 5 (1948). Pages 108-110.

R. Gunn and G.D. Kinzer, “The terminal velocity of fall for water droplets in stagnant air”, *J. Meteor.*, 6 (1949). Pages 243-248.

U. Nakaya and T. Terada, “Simultaneous observation of the mass, falling velocity and form ofsuoxv crystals. *J. Fac. Science Hokkaido Univ.*, Series II (Physics), 1(7) (1935). Pages 191-201.

M.P. Langleben, “The terminal velocity of snowflakes”, *Quart. Jour. Royal Met. Soc.*, 80 (1954). Pages 171-181.

M. Hanesch, *Fall Velocity and Shape of Snowflakes*, PhD Thesis, Swiss Federal Institute of Technology, Zurich (1999).

R. Schefold, B. Baschek, M. Wuest and E. Barthazy, “Fall velocity and axial ratio of snowflakes”, in proceedings of *ERAD 2002* (2002).

Chapter 3: Optimizing domain discretization

Fluid Flow in T-Junction of Pipes, Master’s Thesis, Lappeenranta University of Technology (2007).

Alan Ward and Josep Jorba, “An iterative method for the creation of structured hexahedral meshes over complex orography”, *Applied Mathematics and Computation*, 218 (2011). Pages 3847–3855.

Giles, Mike, and Robert Haimes, "Advanced interactive visualization for CFD", *Computing Systems in Engineering* 1(1) (1990). Pages 51-62.

W. Kelly and B. Gigas, "Using CFD to predict the behavior of power law fluids near axial-flow impellers operating in the transitional flow regime", *Chemical Engineering Science* 58(10) (2003). Pages 2141-2152.

Xinghua Liang and Yongjie Zhang, "Hexagon-based all-quadrilateral mesh generation with guaranteed angle bounds", *Computer Methods in Applied Mechanics and Engineering* 200(23) (2011). Pages 2005-2020.

T.H. Pulliam, "Solution methods in computational fluid dynamics". Notes for the *von Kármán Institute For Fluid Dynamics Lecture Series* (1986).

G. Albertelli and R.A. Crawfis, "Efficient Subdivision of Finite-Element Datasets into Consistent Tetrahedra", *in proceedings of IEEE Visualization '97*, IEEE Computer Society Press (1997). Pages 213-219.

V.I. Weingarten, "The controversy over hex or tet meshing", *Machine design*, 66:8 (1994). Pages 74-76.

D.N. Kenwright and D. A. Lane, "Interactive Time-Dependent Particle Tracing Using Tetrahedral Decomposition", *IEEE Transactions on Visualization and Computer Graphics*, 2(2) (1996). Pages 120-129.

D. Kenwright, "Automatic detection of open and closed separation and attachment lines", in Proceedings of *IEEE Visualization'98*, ACM Press (1998). Pages 151–158.

S. Bryson *et al.*, "Visually exploring gigabyte data sets in real time.", *Communications of the ACM* 42(8) (1999). Pages 82-90.

Alan Ward, "Rodes hidràuliques de la vall d'Ordino: càlcul de la potència instal·lada", *Papers de Recerca Històrica*, Societat Andorrana de Ciències, 3 (2008). Pages 40-48. *In Catalan.*

B. Delaunay, "Sur la sphère vide", *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7 (1934). Pages 793-800.

C.L. Lawson, "Software for C1 Surface Interpolation", *Mathematical Software III*, (1977). Pages 161-194.

D.F. Watson, "Computing the Delaunay Tesselation with Application to Voronoi Polytopes", *The Computer Journal*, 24(2) (1981). Pages 167-172.

T.J. Baker, “Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation”, *Engineering with Computers*, 5 (1989). Pages 161-175.

P.L. George, F. Hecht and E. Saltel E., “Automatic Mesh Generator with Specified Boundary”, *Computer Methods in Applied Mechanics and Engineering*, North-Holland, 92 (1991). Pages 269-288.

N.P. Weatherill and O. Hassan, “Efficient Three-dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints”, *International Journal for Numerical Methods in Engineering*, 37 (1994). Pages 2005-2039.

D.A. Field, “Qualitative measures for initial meshes”, *International Journal for Numerical Methods in Engineering*, 47(4) (2000). Pages 887-906.

V. Parthasarathy and S. Kodiyalam, “A constrained optimization approach to finite element mesh smoothing”, *Finite Elements in Analysis and Design*, 9 (1991). Pages 309-320.

C.K. Lee and S.H. Lo, “A new scheme for the generation of a graded quadrilateral mesh”, *Computers and Structures*, 52(5) (1994). Pages 847-857.

P. Knupp, “Algebraic mesh quality measures”, *SIAM Journal on Scientific Computing*, 23(1) (2001). Pages 193-218.

J. Robinson, “Some new distortion methods for quadrilaterals”, *Finite Elements in Analysis and Design*, 3 (1987). Pages 183-197.

L. Freitag and C. Ollivier-Gooch, “A Comparison of Tetrahedral Mesh Improvement Techniques”, in proceedings of *5th International Meshing Roundtable*, (1995).

T. Zhou and K. Shimada, “An angle-based approach to two-dimensional mesh smoothing”, in proceedings of *9th International Meshing Roundtable* (2000). Pages 373-384.

L. Freitag and C. Ollivier-Gooch, “Tetrahedral Mesh Improvement Using Swapping and Smoothing”, *International Journal For Numerical Methods In Engineering*, 40 (1997). Pages 3979-4002.

T. Blacker, “Meeting the challenge for automated conformal hexahedral meshing”, *9th International Meshing Roundtable*, 11-20 (2000).

R. Gaffney, H. Hassan and M. Salas, “Euler Calculations for Wings Using Cartesian Grids.” AIAA Paper 87-0356, *AIAA Journal* (Jan. 1987).

M.L. Staten, R.A. Kerr, S.J. Owen, T.D. Blacker, M. Stupazzini and K. Shimada, “Unconstrained plastering – Hexahedral mesh generation via advancing-front geometry decomposition”, *International Journal for Numerical Methods in Engineering* on-line, doi:10.1002/nme.2679 (2009).

M.J. Aftosmis, M.J. Berger and J.E. Melton, “Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry”, AIAA Paper 97-0196, *AIAA Journal* (1998).

M.A. Yerry and M.S. Shepard, “A modified quadtree approach to finite element mesh generation”, *IEEE Comp. Graph. Appl.*, 3(1) (1983). Pages 39-46.

M.A. Yerry and M.S. Shephard, “Three-Dimensional Mesh Generation by Modified Octree Technique”, *International Journal for Numerical Methods in Engineering*, 20 (1984). Pages 1965-1990.

M.S. Shephard and M.K. Georges, “Three-Dimensional Mesh Generation by Finite Octree Technique”, *International Journal for Numerical Methods in Engineering*, 32 (1991). Pages 709-749.

R. Lohner, P. Parikh and C. Gumbert, “Interactive Generation of Unstructured Grid for Three Dimensional Problems”, in proceedings of *Numerical Grid Generation in Computational Fluid Mechanics ‘88*, Pineridge Press (1988). Pages 687-697.

S.H. Lo, “Volume Discretization into Tetrahedra-I. Verification and Orientation of Boundary Surfaces”, *Computers and Structures*, 39(5) (1991). Pages 493-500.

S.A. Canann, “Plastering and Optismoothing: New Approaches to Automated, 3D Hexahedral Mesh Generation and Mesh Smoothing”, Ph.D. Thesis, Brigham Young University, Provo, UT. (1991).

T.D. Blacker and R.J. Myers, “Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm”, *Engineering With Computers*, 2 (1993). Pages 83-93.

T.J. Tautges, T. Blacker and S.A. Mitchell, “The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element Meshes”, *International Journal of Numerical Methods in Engineering* (1996).

Haskell B. Curry, “The method of steepest descent for nonlinear minimization problems”, *Quart. Appl. Math*, 2(3) (1944). Pages 250-261.

M.R. Hestenes and E. Stiefel, “Methods of Conjugate Gradients for Solving Linear Systems”, *Journal of Research of the National Bureau of Standards*, 49,(1952). Pages 409-436.

R. Fletcher and M.J.D. Powell, “A Rapidly Convergent Descent Method for Minimization”, *Computer Journal* (1963). Pages 163-168.

S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, “Optimization by Simulated Annealing”, *Science*, 200(4598) (1983). Pages 671-680.

John H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press (1975).

S.A. Canann, J.R. Tristano and M.L. Staten, “An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes”, in proceedings of *7th International Meshing Roundtable* (1998). Pages 479-494.

Threads Extension for Portable Operating Systems (Draft 6), IEEE, P1003.4a (Feb 1992).

D.K.R. Babajee and M.Z. Dauhoo, “An analysis of the properties of the variants of Newton’s method with third order convergence”, *Applied Mathematics and Computation*, 183(1) (December 2006). Pages 659-684.

M.H. Alrefaei and A.H. Diabat, “A simulated annealing technique for multi-objective simulation optimization”, *Applied Mathematics and Computation*, 215(8) (October 2009). Pages 3029-3035.

M. Mitchell, J.H. Holland and S. Forrest, “When will a genetic algorithm outperform hill climbing?”, *Advances in Neural Information Processing Systems*, Morgan Kaufmann, 6 (1993). Pages 51-58.

Chapter 4: Solving the Navier-Stokes equations

OpenFOAM, The OpenFOAM Foundation. URL: http://openfoam.org (retrieved July 15, 2014).

Russel Naughton, *Hargrave: the Pioneers. Aviation and Aeromodelling - Interdependent Evolutions and Histories*, Monash University Department of Engineering (1999). URL:

http://www.ctie.monash.edu.au/hargrave/wenham.html (retrieved December 27, 2014).

Airbus S.A.S., “Airbus Launched the A380’s Certification Campaign”, Airbus Video Gallery, 2014. URL: http://videos.airbus.com/search/?q=wind+tunnel+testing (retrieved December 27, 2014).

Anthony F. Molland, Stephen R. Turnock and Dominic A. Hudson, *Ship Resistance and Propulsion: Practical Estimation of Propulsive Power*, Cambridge University Press, 2011.

Ansys FLUENT. URL: http://www.ansys.com/Products/Simulation+ Technology/Fluid+Dynamics/Fluid+Dynamics+Products/ANSYS+Fluent (retrieved July 15, 2014).

COMSOL Multiphysics. URL: http://www.comsol.com/comsol-multiphysics (retrieved June 9, 2015).

Code Saturne, EDF. URL: http://code-saturne.org/cms/ (retrieved June 9, 2015).

CAE Associates Inc., “Computational Fluid Dynamics: ANSYS CFX and FLUENT CFD Software”, 2014 URL: https://caeai.com/ansys-software-support/ansys-software/computational-fluid-dynamics-ansys-cfx-and-fluent-cfd-software (retrieved December 27, 2014).

Goong Chen, Qingang Xiong, Philip J. Morris, Eric G. Paterson, Alexey Sergeev, and Yi-Ching Wang, “OpenFOAM for Computational Fluid Dynamics”, *Notices of the AMS* 61(4) (April 2014). pp. 354-363.

Hrvoje Jasak, *Error analysis and estimation for the finite volume method with applications to fluid flows.* PhD thesis, Imperial College London (University of London) (1996).

Onno Ubbink, *Numerical prediction of two fluid systems with sharp interfaces.* PhD thesis, Imperial College London (University of London) (1997).

David Paul Hill, *The computer simulation of dispersed two-phase flow.* PhD thesis, Imperial College London (University of London) (1998).

Daniel Brennan, *The numerical simulation of two phase flows in settling tanks.* PhD thesis, Imperial College London (University of London) (2001).

Luca Mangani, *Development and validation of an object oriented cfd solver for heat transfer and combustion modeling in turbomachinery application.* PhD thesis, Università degli Studi di Firenze, Dipartimento di Energetica (2008).

Amy Henderson, Jim Ahrens and Charles Law, *The ParaView Guide*, Kitware Inc., Clifton Park, NY (2004).

OpenFOAM Wiki, "BuoyantBoussiesqPisoFoam" (2014). URL: https://openfoamwiki.net/index.php/BuoyantBoussinesqPisoFoam (retrieved: February 22, 2015)

W. P. Jones and B. E. Launder, "The prediction of laminarization with a 2-equation model of turbulence." *International Journal of Heat and Mass Transfer*, 15 (1972). Page 301.

Osborne Reynolds, "On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion." *Philosophical Transactions of the Royal Society of London*, 186 (1895). pp. 123-164.

P. J. Oliveira and R. I. Issa, "An Improved PISO Algorithm for the Computation of Buoyancy-Driven Flows", *Numerical Heat Transfer*, Part B, 40 (2001). Pages 473-493.

D.A.H. Jacobs, "Preconditioned Conjugate Gradient methods for solving systems of algebraic equations", *Central Electricity Research Laboratories*, (1981).

D.A.H. Jacobs, "A generalization of the conjugate-gradient method to solve complex systems", *IMA journal of numerical analysis* 6(4) (1986). Pages 447-452.

H.A. Van Der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems", *SIAM Journal on Scientific and Statistical Computing*, 13(2) (1992). Pages 631–644.

Uli Göhner, "Performance of Implicit Solver Strategies on GPUs." In proceedings of *LS-DYNA Forum*, Bamberg (2010).

Vratis, *SpeedIT FLOW: a big step forward faster CFD*. Online presentation. URL: http://vratis.com/blog/ (retrieved February 21, 2015)

Edgar Gabriel *et al.*, "Open MPI: Goals, concept, and design of a next generation MPI implementation." *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Springer (2004). Pages 97-104.

Marc Snir *et al.*, *MPI: the complete reference, Scientific and engineering computation series.* MIT Press (1996).

OpenFOAM User Guide, "Running applications in parallel". URL: http://www.openfoam.org/docs/user/running-applications-parallel.php (retrieved July 20, 2014)

Sagi Katz and Ayellet Tal, "Hierarchical mesh decomposition using fuzzy clustering and cuts". *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 22(3) (2003). Pages 954-961.

G. Lavoué, F. Dupont and A. Baskurt, "A new CAD mesh segmentation method, based on curvature tensor analysis", *Computer Aided Design*, 37 (2005). Pages 975-987.

Stefano Berretti, Alberto Del Bimbo and Pietro Pala, "3D Mesh decomposition using Reeb graphs." *Image and Vision Computing*, 27(10) (2009). Pages 1540-1554.

Wu Poting and Elias N. Houstis, "Parallel adaptive mesh generation and decomposition." *Engineering with Computers*, 12(3-4) (1996). Pages 155-167.

George Karypis and Vipin Kumar, "Multilevel k-way Partitioning Scheme for Irregular Graphs", *J. Parallel Distrib. Comput.*, 48(1) (1998). pp. 96-129.

Jesús Antonio Izaguirre Paz, "Evaluation of Parallel Domain Decomposition Algorithms." In *1st National Computer Science Encounter*, (1997).

François Pellegrini and Jean Roman, "Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs." *High-Performance Computing and Networking*, Springer (1996). Pages 493-498.

Gene M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities." *Proceedings of the April 18-20, 1967, Spring joint computer conference*, ACM (1967).

John L. Gustafson, "Reevaluating Amdahl's law." *Communications of the ACM*, 31(5) (1988). Pages 532-533.

Shi Yuan, "Reevaluating Amdahl's law and Gustafson's law." Computer Sciences Department, Temple University (MS: 38-24) (1996).

Mark D. Hill and Michael R. Marty, “Amdahl’s Law in the Multicore Era.” *IEEE Computer*, 41(7) (2008). Pages 33-38.

Håkan Nilsson, “Some experiences on the accuracy and parallel performance of OpenFOAM for CFD in water turbines.” *Applied Parallel Computing, State of the Art in Scientific Computing*, Springer (2007). Pages 168-176.

Orlando Rivera, Karl Fürlinger and Dieter Kranzlmüller, “Investigating the scalability of openFOAM for the solution of transport equations and large eddy simulations.” *Algorithms and Architectures for Parallel Processing.* Springer (2011). Pages 121-130.

John Ruge and Klaus Stüben, *Efficient solution of finite difference and finite element equations by algebraic multigrid AMG*. Gesellschaft f. Mathematik u. Datenverarbeitung (1984).

Massimiliano Culpo, “Current bottlenecks in the scalability of OpenFOAM on massively parallel clusters.” PRACE white paper, (2011).

Josep Jorba Esteve, *Analisis automático de prestaciones de aplicaciones paralelas basadas en paso de mensajes*, PhD Thesis, Universitat Autònoma de Barcelona (2006).

Intel Corporation, “Intel Xeon Processor E5440” (2008). URL: http://ark.intel.com/products/33082/Intel-Xeon-Processor-E5440-12M-Cache-2_83-GHz-1333-MHz-FSB (retrieved December 28, 2014).

Chapter 5: Coupling snowfall with transport fluid motion

Alan Ward and Josep Jorba, “Planning passive snowdrift reduction on high-altitude roads with lateral obstacles to wind flow.” In proceedings of *SIRWEC 17th International Road Weather Conference*, Andorra (January 2014).

Clayton T. Crowe, Efstathios Michaelides, John D. Schwarzkopf, *Multiphase Flow Handbook*, CRC Press (2005).

J. Rutqvist, Y-S Wu, C-F Tsang, G. Bodvarsson, “A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock”, *International Journal of Rock Mechanics and Mining Sciences* 39 (2002). Pages 429–442

Niels G. Jacobsen, David R. Fuhrman, and Jørgen Fredsøe, "A wave generation toolbox for the open-source CFD library: OpenFoam®.", *International Journal for Numerical Methods in Fluids*, 70(9) (2012). Pages 1073-1088.

Tso-Ren Wu *et al.*, "Dynamic coupling of multi-phase fluids with a moving obstacle", *Journal of Marine Science and Technology* 19:6 (2011). Pages 643-650

Thomas J.R. Hughes, Wing Kam Liu and Thomas K. Zimmermann, "Lagrangian-Eulerian finite element formulation for incompressible viscous flows", *Computer Methods in Applied Mechanics and Engineering*, 29:3 (December 1981). Pages 329–349

Alan Ward, *MPIGL: un projecte d'utilització del Programari Lliure en la visualització gràfica en xarxes de càlcul científic.* Master's Thesis, Universitat Oberta de Catalunya (2009). *In Catalan.*

Daniel V. Schroeder, *An Introduction to Thermal Physics*, Addison Wesley (2000).

C. Donald Ahrens, *Essentials of Meteorology: An Invitation to the Atmosphere*, Cengage Learning (2011). ISBN 0840049331, 9780840049339.

F. Naaim-Bouvet, M. Naaim and J-L. Michaux, "Snow fences on slopes at high wind speed: physical modelling in the CSTB cold wind tunnel." *Natural Hazards and Earth System Sciences* 3,4 (2002). pp 137–145.

Zhou Xuan-yi and Gu Ming, "Numerical Simulation of Snow Drift on the Surface of a large-span Roof Structure", in proceedings of *Fourth International Symposium on Computational Wind Engineering* (CWE2006), Yokohama (2006). pp. 889-892.

Perry Bartelt and Michael Lehning, "A physical SNOWPACK model for the Swiss avalanche warning: Part I: numerical model." *Cold Regions Science and Technology*, 35(3) (2002). Pages 123-145.

Gernot Michlmayr *et al.*, "Application of the Alpine 3D model for glacier mass balance and glacier runoff studies at Goldbergkees, Austria." *Hydrological processes*, 22(19) (2008). Pages 3941-3949.

Florian Kobierska *et al.*, "Climate change effects on snow melt and discharge of a partly glacierized watershed in Central Switzerland (SoilTrec Critical Zone Observatory)." *Applied Geochemistry*, 26 (2011). Pages S60-S62.

S. Elghobashi, “On predicting particle-laden turbulent flows.” *Applied Scientific Research*, 52(4) (1994). Pages 309-329.

Aurélia Vallier, *Eulerian and Lagrangian cavitation related simulations using OpenFOAM*, Eng. Thesis, University of Lund (2010).

André Kaufmann, *Towards eulerian-eulerian large eddy simulation of reactive two-phase flows*, PhD Thesis, Institut National Polytechnique de Toulouse (2004).

Don W. Green(Ed.), *Perry's Chemical Engineers' Handbook*, Section 6 Fluid and Particle Dynamics, McGraw Hill Professional (2007). ISBN: 0071542132, 9780071542135

J. Nelson, “Origin of diversity in falling snow.” *Atmospheric Chemistry and Physics* 8(18), (2008). Pages 5669-5682.

Andrew J. Heymsfield and Masahiro Kajikawa, “An improved approach to calculating terminal velocities of plate-like crystals and graupel.” *Journal of the Atmospheric Sciences* 44(7), (1987). Pages 1088-1099.

C. T. Crowe, T. R. Troutt and J. N. Chung, “Numerical models for two-phase turbulent flows.” *Annual Review of Fluid Mechanics* 28(1), (1996). Pages 11-43.

Suzuki Yuji, Motofumi Ikenoya and Nobuhide Kasagi, “Simultaneous measurement of fluid and dispersed phases in a particle-laden turbulent channel flow with the aid of 3-D PTV.” *Experiments in fluids* 29(1), (2000). Pages S185-S193.

Hu Zhiwei, Luo Xiaoyu and Luo Kai H., “Numerical simulation of particle dispersion in a spatially developing mixing layer.” *Theoretical and Computational Fluid Dynamics* 15(6) (2002). Pages 403-420.

Servei Meteorològic de Catalunya, *Anuari de la Xarxa d'Estacions Meteorològiques Automàtiques*, (2013). URL: http://www20.gencat.cat/portal/site/meteocat/menuitem.0733ee5bfae8638c5c121577b0c0e1a0/?vgnextoid=f936014bb3823210VgnVCM1000000b0c1e0aRCRD (retrieved July 20, 2014) *In Catalan.*

R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting”. *Journal of Computational Physics*, 62(1) (1986). Pages 40-65.

Alan Ward and Josep Jorba, “Harmonic buffeting in a high-altitude ridge-mounted triblade Horizontal Axis Wind Turbine.” *J. Wind Eng. Ind. Aerodyn.*, 121 (2013). Pages 106-115.

Chapter 6: Case Studies

Daniel Scott and Geoff McBoyle, “Climate change adaptation in the ski industry.” *Mitigation and Adaptation Strategies for Global Change*, 12(8) (2007). Pages 1411-1431.

Jörgen Rogstam and Mattias Dahlberg, *Energy usage for snowmaking*, Energi and Kylanalys AB, Älvsjö (2011).

Christian Rixen, *et al.*, “Winter tourism and climate change in the Alps: an assessment of resource consumption, snow reliability, and future snowmaking potential.” *Mountain Research and Development* 31(3) (2011). Pages 229-236.

Catherine M. Pickering and Ralf C. Buckley, “Climate response by the ski industry: the shortcomings of snowmaking for Australian resorts.” *Ambio*, 39(5-6) (2010). Pages 430-438.

Servei Meteorològic de Catalunya, *Estacions Automàtiques* On-line Meteorological Database. URL: https://www.meteo.cat/ (retrieved on August 14, 2014). *In Catalan.*

Servei de Meteorologia, Govern d’Andorra, On-line Meteorological Database. URL: https://www.meteo.ad/ (retrieved on August 14, 2014). *In Catalan.*

Pere Esteban *et al.*, “Atmospheric circulation patterns related to heavy snowfall days in Andorra, Pyrenees.” *International Journal of Climatology* 25(3) (2005). Pages 319-329.

Vijay P. Singh, Pratap Singh and Umesh K. Haritashya, *Encyclopedia of Snow, Ice and Glaciers*, Springer Science and Business Media (2011). ISBN: 9048126428, 9789048126422

Y. Lappin, “Roads to Jerusalem closed as huge storm batters Israel”. Jerusalem Post (December 13, 2013). URL: http://www.jpost.com/National-News/Police-IDF-called-in-to-help-motorists-stranded-in-snow-in-Jerusalem-BG-Airport-shut-temporarily-334920 (retrieved January 1, 2014)

R.D. Tabler, *Design Guidlines for the Control of Blowing and Drifting Snow* SHRP-H-381, Strategic Highway Research Program, National Research Council, Washington (1994).

Canal 324, "S'incrementa el risc d'allaus a Catalunya després de les últimes nevades", Televisió de Catalunya (November 9, 2011). URL: http://www.324.cat/noticia/28302/societat/Sincrementa-el-risc-dallaus-a-Catalunya-despres-de-les-ultimes-nevades (retrieved July 25, 2014) *In Catalan.*

R. M. Lang and G.L. Blaisdell, "Passive snow removal with a vortex generator at the Pegasus runway", *Antarctica. Ann. Glaciology*, 26 (1998). Pages 231-236.

Diari d'Andorra, *FEDA descarta estendre l'ús de la calefacció amb energia geotèrmica* (December 22, 2013). URL: http://www.diariandorra.ad/index.php?option=com_k2&view=item&id=29959&Itemid=413 (retrieved July 25, 2014) *In Catalan.*

El Periòdic d'Andorra, *L'únic cotxe elèctric matriculat a Andorra és el de FEDA*, (June 21, 2014). URL: http://www.elperiodicdandorra.ad/societat/34651-lnic-cotxe-elctric-matriculat-a-andorra-s-el-de-feda.html (retrieved July 25, 2014) *In Catalan.*

G.T. Bitsuamlak, T. Stathopoulos and C. Bédard, "Numerical evaluation of wind flow over complex terrain: review." *Journal of Aerospace Engineering* 17(4) (2004). pp. 135-145.

Forces Elèctriques d'Andorra, *La viabilitat de l'energia eòlica a l'estudi amb una instal.lació situada al Pic del Maià* (November 16, 2011). URL http://premsa.feda.ad/notesdepremsa/detall/id/59 (retrieved May 21, 2012) *In Catalan.*

Enercon, *Enercon Wind energy converters: product overview*, ENERCON Gmbh., (July 2010). URL: http://www.scribd.com/doc/46971599/En-Productoverview-0710 (retrieved August 10, 2012)

Emerging Technology Corporation, "Horizontal Axis Wind Turbine - 1 MW". URL: http://www.etcgreen.com/horizontal-axis-wind-turbine-1mw (retrieved January 9, 2015)

Forces Elèctriques d'Andorra, On-line Meteorological Database. URL: https://www.feda.ad/cat/coneixnos/comunicacio/meteo.aspx (retrieved on June 7, 2012). *In Catalan.*

P. Singh and V.P. Singh, *Snow and Glacier Hydrology.* Water Science and Technology Library, 37, Springer (2001).

B. Blocken *et al.*, "CFD simulation of the atmospheric boundary layer: wall function problems". *Atmospheric Environment* 41 (2007). Pages 238-252.

H. Glauert, "Airplane propellers". *Aerodynamic theory*, 4, Division L. Julius Springer, Berlin (1935). Pages 169-360.

P.T. Smulders, G. Lenssen and H. vanLeeuwen, "Experiments with wind rotors in yaw". In proceedings of *the International Symposium on Environmental Problems*, Patras, Greece (1981).

C.G. Helmis *et al.*, "An experimental study of the near wake structure of a wind turbine operating over complex terrain". *Solar Energy*, 54 (6) (1995). Page 413.

A.A. Afjeh and T.G. Keith, "A simplified free wake method for horizontal axis wind turbine performance prediction", *Trans. ASME*, 108 (1986). Page 400.

L. Bermúdez, A. Velázquez and A. Matesanz, "Numerical Simulation of Unsteady Aerodynamics Effects in Horizontal-Axis Wind Turbines", *Solar Energy*, 68(1) (2000). Pages 9-21.

S. Voutsinas, S.G. Belessis and K.G. Rados, "Investigation of the yawed operation wind turbines by means of a vortex particle method", *AGARD CP-552*, Aerodynamics and Aeracoustics of Rotorcraft (1994).

S.D. Pesmajoglou and J.M.R. Graham, "Prediction of aerodynamic forces on horizontal axis wind turbines in free yaw and turbulence", *Journal of Wind Engineering and Industrial Aerodynamics*, 86 (2000). Pages 1-14.

J.T. Conway, "Application of an exact nonlinear actuator disk theory to wind turbines", in proceedings of *ICNPAA 2002*, Melbourne, Florida, USA (15-17 May 2002).

I. Dobrev, F. Massouh and M. Rapin, "Actuator surface hybrid model", *J. Phys.: Conference Series*, 75 conference no. 012019 (2007).

R.E. Froude, “On the part played in propulsion by differences of fluid pressure”, *Trans. Inst. Naval Archit.*, 30 (1889).

A. Betz, “Das Maximum der theoretisch moglichen Ausnutzung des Windes durch Windmotoren”, *Z. Gesamte Turbinewesen*, 26 (1920). *In German.*

J.N. Sorensen and Wen Z.S., “Numerical Modeling of Wind Turbine Wakes”, *J. Fluids Eng.*, 124(2) (June 2002). Pages 393-400.

N. Troldborg, *Actuator Line Modeling of Wind Turbine Wakes*, Ph.D. thesis, Technical University of Denmark (2008).

C. Masson and C.S. Watters, “Moving actuator surfaces: A new concept for wind turbine aerodynamic analysis.”, in proceedings of the *International Conference on Renewable Energies and Power Quality (ICREPQ08)* (2008).

P. Passon and M. Kühn, “State-of-the-art and Development Needs of Simulation Codes for Offshore Wind Turbines”, proceedings of *Copenhagen Offshore Wind 2005 Conference and Exposition*, Copenhagen, Denmark (October 2005).

Govern d’Andorra, “Web de meteorologia de l’Àrea de Transport i Energia”. URL: http://www.meteo.ad/climatologia.php?idioma=0 (retrieved October 10, 2012) *In Catalan.*

A.C. Hansen, *Yaw Dynamics of Horizontal Axis Wind Turbines: Final Report*, NREL/TP-442-482, National Renewable Energy Laboratory, Division of Midwest Research Institute, Salt Lake City (May 1992). Page 10.

Alpine Test Site Goetsch, Meteorological measurements and wind turbine performance analysis COST Action 727: measuring and forecasting atmospheric icing on structures. URL: http://www.meteotest.ch/cost727/index.html (retrieved June 12, 2015).

Chapter 7: Concluding notes

SCImago Journal and Country Rank, “Journal of Applied Mathematics and Computing”. URL: http://www.scimagojr.com/journalsearch.php?q=23821&tip=sid (retrieved January 9, 2015)

SCImago Journal and Country Rank, “Journal of Wind Engineering and Industrial Aerodynamics”. URL:

http://www.scimagojr.com/journalsearch.php?q=110486&tip=sid (retrieved January 9, 2015)

Sean Gallagher, “How IBM’s Deep Thunder delivers hyper-local forecasts 3-1/2 days out”. Ars Technica (March 14, 2012). URL: http://arstechnica.com/business/2012/03/how-ibms-deep-thunder-delivers-hyper-local-forecasts-3-12-days-out/ (retrieved January 18, 2015)