Stageverslag

Modelling of Birch pollen concentrations using an atmospheric transport model

Jérôme Hilaire

De Bilt, 2007

KNMI Stageverslag

De Bilt, 2007

PO Box 201
3730 AE De Bilt
Wilhelminalaan 10
De Bilt
The Netherlands
http://www.knmi.nl
Telephone +31(0)30-220 69 11
Telefax +31(0)30-221 04 07

Auteur: Hilaire, J.

Modelling of Birch pollen concentrations using an atmospheric transport model

Jérôme Hilaire

August 31, 2007

Acknowledgments

I would like to thank my tutor, Henk Eskes from KNMI chemistry and climate research for his trust and his advice. Gertie Geertsema from KNMI weather research for her numerical advice. The whole KS-CK department for their advices. Arnold Van Vliet and Letty De Weger for their support and help on this project.
My professor from the Universidad de Chile, Laura Gallardo-Klenner for having giving me the wish to know more about air quality, atmosphere and climate change.
Mikail Sofiev from FMI for sharing his knowledge on this field and for providing his own code.
Laurent Menu from LMD/IPSL for enlighting me on several numerical aspects of Chimere.

Résumé du projet en français

Introduction

L'objectif principal de ce stage de fin d'études au sein de l'Institut Royal Néerlandais de Météorologie (KNMI) a été de développer un modèle de dispersion du pollen. Connu pour son agressivité sous nos latitudes, le pollen de bouleau a été tout naturellement placé au coeur de cette étude. Ainsi, une lecture préliminaire dans la presse scientifique de ses comportements physiques et biologiques a servi de base au projet. Puis une dissection du code du modèle de transport Chimere a permis de meilleures compréhension et approche du problème. L'implémentation des caractéristiques du pollen s'est alors déroulée en s'appuyant sur les deux études décrites précédemment. Finalement, les résultats ont été analysés puis comparés avec les observations.

Etude préliminaire

J'ai consacré la première partie de mon travail sur une riche étude bibliographique (cf chapitre 4) pendant laquelle j'ai réussi à filtrer et dégager l'information nécessaire à la modélisation de la dispersion du pollen (cf chapitre 5). On distingue clairement deux catégories principales d'articles. La première regroupe les documents traitant des modèles statistiques. Ces derniers mettent en avant des corrélations entre certains évènements météorologiques et certains comportements des grains de pollen. Ils permettent donc d'estimer l'importance d'un paramètre par rapport aux autres. La seconde catégorie regroupe, quant à elle, les articles décrivant les modèles numériques qui utilisent généralement des modèles de transport atmosphérique. Ceux-ci sont de très bons supports au développement de nouveaux modèles numériques et sont donc très intéressants pour ce projet. Au terme de cette étude et en accord avec les membres du projet, seuls les paramètrisations d'émission, de transport par masses d'air et de dépôt sec et humide ont été conservés pour la modélisation.

Etude du modèle Chimere

La deuxième partie de mon projet s'est focalisée sur le modèle Chimère (cf chapitre 7). Après une lecture de la documentation et du code en détails et après la réalisation de quelques tests du modèle, j'ai implémenté un nouveau traceur de pollen de bouleau dans Chimère. Puisque le pollen est un aérosol, il était plus aisé d'utiliser directement certaines parties natives du code de Chimere comme celui décrivant le transport par masses d'air. Ainsi,

le pollen est considéré comme un aérosol particulier au sein du modèle car il ne fait pas appel à certains processus tels que la nucléation, la coagulation, etc... En outre, Chimere est un modèle utilisant un environnement multi-processeurs, j'ai donc mené une étude annexe afin de connaître la configuration optimale qui me permettrait de calculer les simulations en un temps minimum (cf Annexe F, page p.77). Finalement, j'ai créé un ensemble de tests qui utilisent différentes cartes de végétation afin de mettre en avant certains caractères météorologiques particuliers liés aux comportements du pollen.

Analyse des résultats

Seulement trois tests sur onze ont pu être réalisés (cf chapitre 8). En effet, le modèle nécessite un peu plus de dix heures pour achever un calcul ce qui nécessite une bonne organisation mais surtout beaucoup de temps pour acquérir des résultats. Avant tout, j'ai tenu à visualiser l'évolution de l'accumulation de chaleur en Europe. On constate bien que ce terme est plus important dans le sud que dans le nord. Par ailleurs aux Pays-Bas, ce terme est plus important à l'ouest qu'à l'est. Quant au premier test, il présente la situation irréaliste d'une seule forêt située aux Pays-Bas dans la zone de De Bilt. On remarque un décalage du pic de concentrations vers la droite, montrant un retard. Il y a fort a parier que le pollen compté à Leiden provient d'arbres locaux. En effet, le deuxième test montre un décalage moins important. Finalement, le troisième test utilise une carte digitalisée à faible résolution. Dans ce cas, on note bien le caractère volatile du pollen, se déplaçant d'un pays à un autre et traversant les mers sur des centaines de kilomètres. Le pic de concentrations est également en retard ce qui se comprend bien puisqu'il n'y a quasiment aucun arbre aux Pays-Bas sur cette carte. Ainsi, le pollen doit probablement provenir d'une zone en Allemagne plus peuplée en arbres.

Conclusion

Les résultats obtenus à l'issue des simulations sont très prometteurs puisqu'ils mettent en exergue le fort caractère volatile du pollen et présentent une bonne corrélation avec les observations. Cependant, une analyse plus poussée du modèle est primordiale pour le valider. Ainsi, il est très important de calculer la série complète de tests. Certains paramètres doivent aussi faire l'objet d'une attention particulière comme le cycle diurne d'émission qui reste très simple dans cette première version et devrait être paramétrisé par le cycle du soleil. De plus, le champ de températures utilisé est celui d'observations prises à deux mètres du sol. Une étude comparative devrait être mené avec le champ tridimensionnel de températures. Finalement, outre de nouvelles connaissances en modélisation atmosphérique et qualité de l'air, ce stage m'a permis d'une part de développer pleinement mes capacités de travail en autonomie sur un projet multi disciplinaire et d'autre part de réaliser à quel point je souhaite m'investir pleinement dans les sciences atmosphériques et le changement climatique à l'issue de ce stage.

Contents

Chapter 1

Introduction

In the frame of education at INSA Toulouse, students have to make a five-month internship to achieve their Degree of Master Engineering. They also have the opportunity to study abroad. Then, after a semester at Universidad de Chile, where I chose to take as my major, courses more or less related to environment and where I had an introduction on atmospheric pollution modelling, I decided to find my internship in the field of air quality and modelling. This subject is really interesting and motivating as it belongs to one of the biggest and current scientific issue namely the climate change.

My researches lead me to the department of Chemistry and Climate at KNMI headquarters in De Bilt in The Netherlands which belongs to the Ministerie van Verkeer en Waterstaat (Ministry of Traffic, Public Works and Water Management). On the one hand, the Royal Dutch Meteorological Institute (KNMI) provides weather forecasts and services to Dutch people and companies and on the other hand, as a national research centre, it also supports projects and researches on a diverse variety of subjects such as weather, climate, climate change, and seismology.

A few month before I arrived, Henk Eskes and Gertie Geertsema from KNMI accepted to join an initiative co-ordinated by Arnold Van Vliet from Wageningen University and research center (WUR) named 'Natuurlijk gezond van dag tot dag[1]'. Thus, the MSA-WUR specialised in biology and phenology of plants, the Leiden University Medical Center (LUMC) specialised in medical sciences and the KNMI decided to drive a study together on the modelling of pollen forecasting for The Netherlands. Indeed, affecting more than fifteen percent of the central and northern parts of European population, pollen epidemics have become a real plague since a few years. Especially, because the climate change by rising temperatures emphasizes pollen production and rare phenology behaviours. Moreover, in our regions, Birch pollen is one of the most agressive species. As a result, we decided to focus on Birch pollen as a first step.

During the period of my internship, my task was to set up a Birch pollen forecast model for The Netherlands. First, I read several scientific papers about Birch pollen and pollen in

[1]Naturally healthy from day to day

general to draw the different behaviours of Birch trees and pollen grains in the atmosphere. Secondly, I modeled the key processes in preparation for implementing them further. In parallel, I studied the Chimere chemical transport model to find out how to develop a pollen module in it. Then, I created the required pollen modules within Chimere. Finaly, I ran a serie of tests and analyses to validate my model.

Chapter 2

Presentation of the project

2.1 Natuurlijk gezond van dag tot dag

This project is part of a bigger one initiated and co-ordinated by Arnold Van Vliet from MSA-WUR known under the name: 'Natuurlijk gezond van dag tot dag[1]' (RGI246). 'Natuurlijk gezond van dag tot dag' involves more than 5 sub-projects and 15 partners aiming to create a national interactive website for monitoring, forecasting, managing and communicating health risks from nature in space and time.

2.2 A project about pollen forecasting

The main objective of this project is to set up a pollen forecast model for the Netherlands using Chimere, a meso-scale atmospheric transport model. This forecast model would be then used to foresee the risk of hay fever due to the bloom of Birch trees. To solve this puzzle, three organisations are involved (refer to figure 2.1):

- KNMI[2]
- LUMC[3]
- MSA-WUR[4]

2.3 KNMI

2.3.1 General presentation

The Koninklijk Nederlands Meteorologisch Instituut, or the Royal Dutch Meteorological Institute, is known mainly for its weather forecasts, warnings and services. However, it

[1]Naturally healthy from day to day

[2]Koninklijk Nederlands Meteorologisch Instituut (Royal Netherlands Meteorology Institute)

[3]Leids Universitair Medisch Centrum (Leiden University Medical Center)

[4]Wageningen Universiteit, Leerstoelgroep Milieusysteemanalyse (Wageningen University, Environmental Systems Analysis Group)

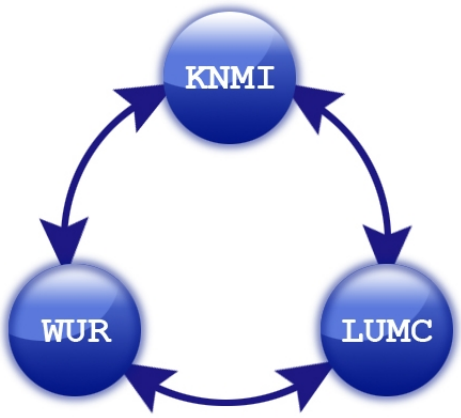

Figure 2.1: Organigram

provides a lot more in its capacity as a national data and knowledge centre for weather, climate and seismology researches. This research institute is located in De Bilt (refer to figure 2.2). Today, KNMI hosts five hundred people in total and counts about 250 researchers working on about hundred projects (refer to figure 2.3).

Figure 2.2: KNMI headquarters in De Bilt

2.3.2 Mission

As an agency of the Ministry of Transport, Public Works and Water Management (Ministerie van Verkeer & Waterstaat), KNMI has the main goal to provide information about weather, climate research and seismology and each of the three divisions ensures to fulfill it. First, meteorological research provides quality meteorological data and knowledges to external users such as the public at large, the government, aviation or the shipping industry. Then, the climate research acts as national research and information centre for climate and climate change at a high scientific level to try to answer the issue of the twenty-first century. Finally, the division of seismology aims to find the causes, effects and risks of earthquakes in The Netherlands.

2.3.3 Presentation of KS-CK division

The main aim of the Chemistry and Climate division is to quantify, understand and predict anthropogenic and natural changes in atmospheric composition and their consequences for climate, air quality and ultraviolet radiation. An important activity is the improvement of models of the life-cycles of non-CO2 greenhouse gases, air pollutants, aerosols and their precursors in order to produce accurate short term air quality forecasts and long term scenarios for the future. A team of 10 people is currently developing and improving TM (Transport Model), previously named CTMK, which is a three-dimensional chemical transport model coupled off-line to ECMWF[5] meteorological fields. Presently, model improvements and process studies focus on taking into account the coupling of atmospheric chemistry with other components of the climate system, including the biosphere, and on data assimilation of chemical composition. Model evaluation is done by reconstructing past observations on time scales of days and decades back to palaeo scales of 100.000 years. Ground-based observations in The Netherlands and Surinam are performed for model evaluation, trend monitoring and satellite applications such as validation of space-based ozone instruments. Air quality work is also done in this division. In this case, the regional chemistry transport models Chimere and LOTOS-EUROS are used. As for Chimere, it is used for daily air quality forecasts.

2.3.4 Objectives for the project

The KNMI team composed by Henk Eskes from Chemistry and Climate research division, Gertie Geertsema from Weather research division and I from INSA Toulouse was willing to implement pollen behaviour in Chimere in order to obtain a pollen forecast model for The Netherlands.

2.4 LUMC

2.4.1 General Presentation and mission

The Leiden University Medical Center strives to provide high quality of both medical and technical information. The research of the Department of Clinical Epidemiology aims at

[5]European Center for Medium Range Weather Forecasts

translating molecular and biochemical abnormalities to consequences for the patients.

2.4.2 Objectives for the project

Since a couple of years, this group is performing routine pollen observations and have developed a pollen forecast based on meteorological indicators which will be useful for the study and the model. Thus, Letty De Weger was able to bring a strong background on medical aspect and a lot of information about pollen observations.

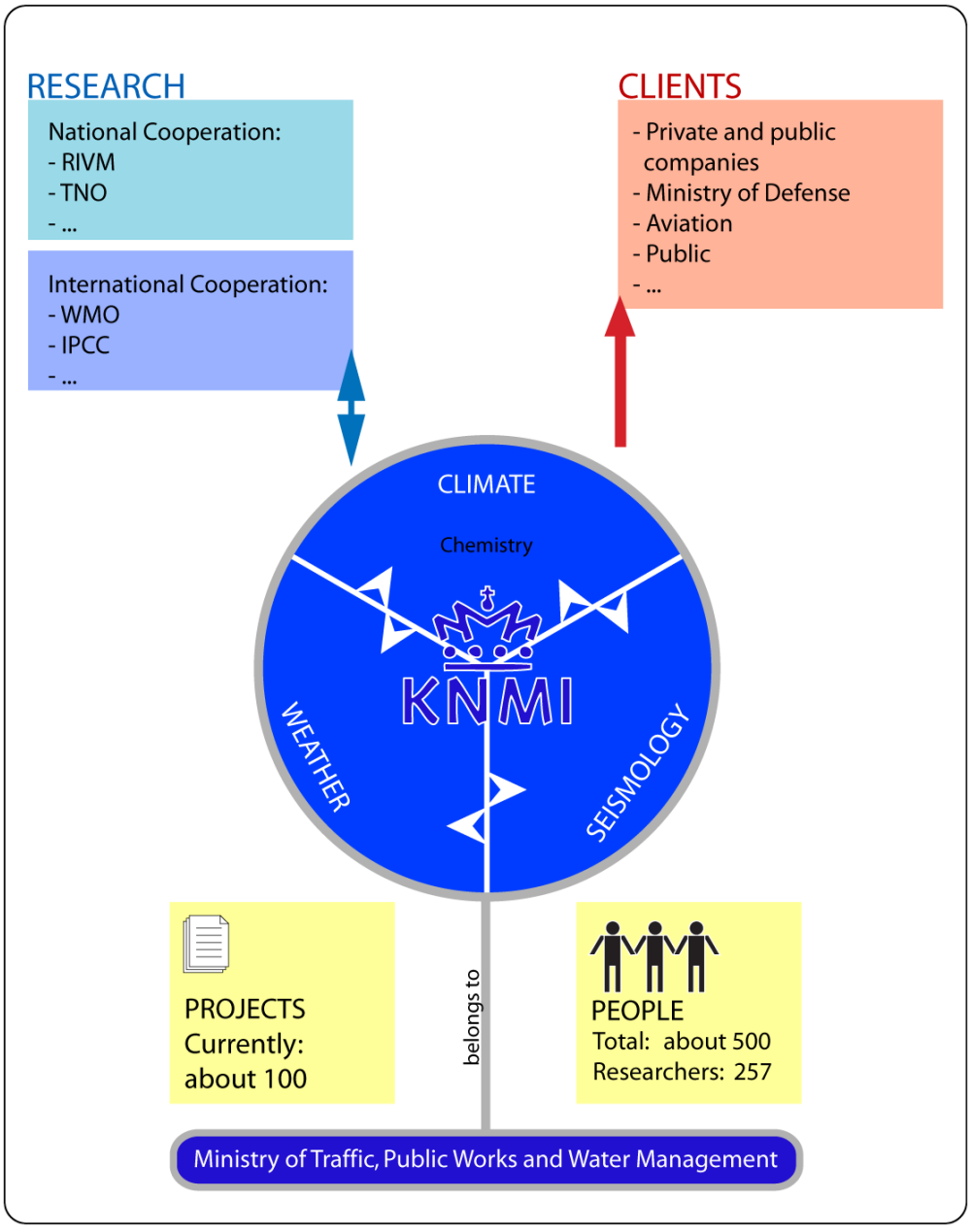

Figure 2.3: KNMI organisation

2.5 MSA-WUR

2.5.1 General Presentation and mission

The Environmental Systems Analysis Groups (refer to figure 2.4)is part of the Department of Environmental Sciences of Wageningen University. Its mission is the improvement of concepts and methods for environmental systems analysis by developing environmental systems analysis using both a substance approach and a regional approach.

2.5.2 Objectives for the project

In this institute, Arnold Van Vliet works on the natural yearly cycle of plants and animals, and changes due to a changing climate[6]. Moreover, he has a large network of volunteers who deliver information on such events. Consequently, he was able to provide useful information about biological and phenological behaviours of plants.

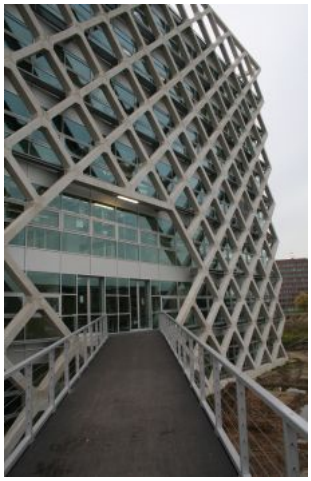

Figure 2.4: Leerstoelgroep Milieusysteemanalyse (Environmental Systems Analysis Group)

[6]www.natuurkalender.nl is a website that provides this type of information, namely timing of yearly events like flowering, occurrence of insects ... in relation to climate change

Chapter 3

Objectives and methods of the project

3.1 Objectives

- Draw the different behaviours of plants and pollen from literature The first objective of this project was to obtain an overview of the pollen emission and key pollen behaviours.
  - Reading scientific papers (refer to chapter 4)
  - Reading Chimere documentation (refer to chapter 6)
  - Researches on Internet to verify the information
- Analyse Chimere code to better implement a pollen model The second objective was to understand how Chimere works in detail.
  - Overview (refer to chapter 6)
  - Follow the code
  - Fine analyse of the code
- Develop a pollen forecast model using Chimere The third objective was to implement all pollen behaviours drawn from the first part.
  - Implementation of atmospherical processes(refer to chapter 7)
  - Implementation of emission model (refer to chapter 7)
- Analyse and interpretate results Finally, the fourth objective was to analyse the results given by our model and interprate them by using our meteorological knowledge.
  - Heat sum
  - Study cases

3.2 GANTT chart

The GANTT chart is given on figure 3.1. The main aspects of the project have been drawn on the GANTT diagram. As for the literature study, it stopped after the meeting with Henk Eskes and Arnold Van Vliet. But we kept reading articles during the period of the project to improve my understanding about pollen.

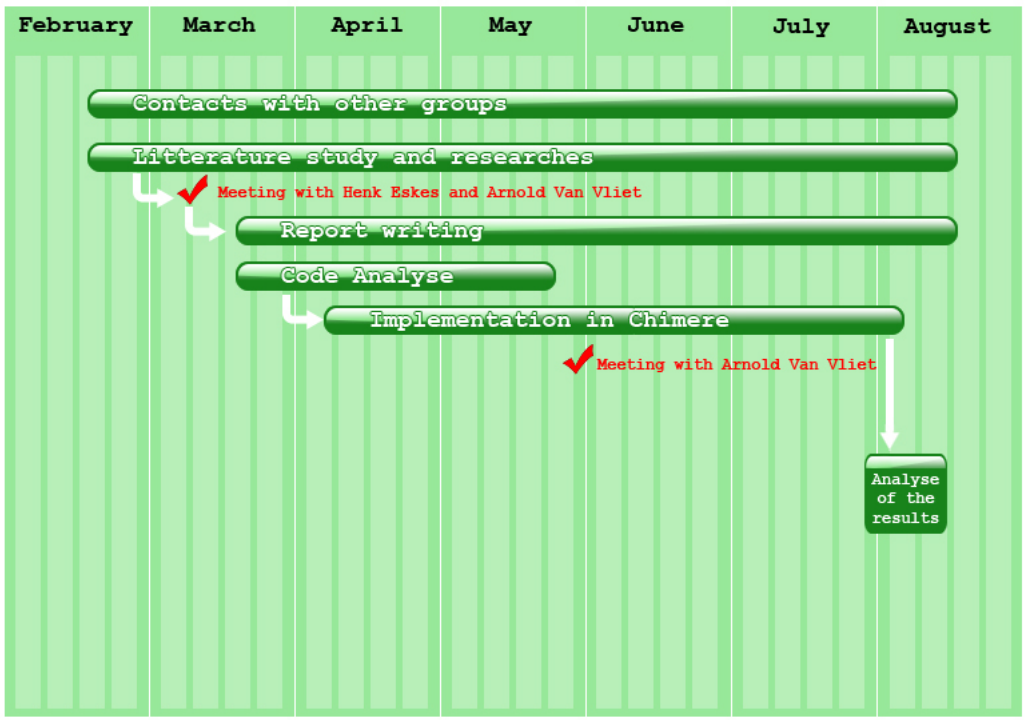

Figure 3.1: GANTT Diagram

3.3 Methodology

Literature study

To perform this objective, I read a large number of scientific papers which is always a good starting point. Then, I verified the statements found in these papers by Internet researches. During this first step, it was important to read a lot of information and then to filter and summarize it. Moreover, it was also useful to be in contact with the other groups involved in pollen issues to confront ideas. As a second stage, a reading of the documentation of Chimere let me have an overview of the next step while starting to elaborate a simple model.

Dissection of the code

During this work, I analysed the code step by step and routine by routine. This study gave me a good understanding of how Chimere works. Consequently, I could draw some information about how to implement the pollen. The reading of Chimere documentation contributed to accelerate the understanding of this model. Nevertheless, this documentation is not sufficient to understand the model. Some aspects still remind blurred, so it is legitimated to read the code in detail. As a result, I dissected the main code of Chimere in order to get a better insight into this model and to improve the rapidity of the implementation.

Implementation

Implementation was quite long, due to the complexity of the existing code and the willing to have an optimized and quite independent pollen module. Moreover, I obtained some issues because of the use of wrong assumptions about units. Nevertheless, I decided to use CVS[1] to secure the development of the model which was very useful to return to a previous version of my work. Finally, I created a serie of eleven tests to better understand pollen behaviours in the atmosphere.

Analyse of the results

I started to analyse the heat sum which is one of the most important parameter in our model. Then, I ran two simple tests in which there was just one cell occupied by Birch trees within the simulation domain. And I ran an extra test with a European Birch tree map. Finally, I visualised pollen concentrations over Europe and The Netherlands. And I compared simulated concentrations in Leiden's cell with real observations from Leiden.

[1]Concurrent Versions System. It is an open-source version control system that keeps track of all work and all changes in a set of files and allows several developers to collaborate. *(source wikipedia)*

Chapter 4

Literature background

While reading pollen literature, one can realize the interest for pollen dispersion started around 1950 and still receives considerable scientific attention. One of the first documents has been released in 1953 in the Journal of Forestry of the Society of American Foresters. Jonathan W. Wright proposed an experiment about pollen dispersion in a forest. Nevertheless, studies are not focused on dispersion only. Nowadays, pollen is also at the heart of biological, medical, paleoclimatical and economical assigments. For instance, a lot of groups are involved in the properties and the mutation of Bet v1[1]. Others try to find out means to fight against hay fever or study the physiology of trees and plants. The advance in all theses domains will let the community increase our understanding of how pollen and trees behave. Considering pollen dispersion, documents are more and more numerous demonstrated by a little study (refer to figure 4.1) on the google website. They can be divided into statistical and numerical approaches which require both data like pollen production, pollen grain properties and plant-specific threshold values. Finally, it has to be noted that the Journal of the International Society of Biometeorology and Aerobiologia, are both extremely relevant sources of information for this study.

4.1 Statistical models

The main purpose of the groups involved in statistical modelling is to find correlations between meteorological data and pollen concentrations, to provide parametrizations for pollen behaviour and emission processes. For instance, Emberlin et al. [10, 11], introduced statistical forecast models to predict the risks of hay fever. In 2006, they managed to created a 30-day-ahead forecast model for grass pollen in north London. Actually, this forecast model uses a significant range of data from 1961 to 1999 and eight multiple regression models, each covering a 10-day period running consecutively fron 21 May to 8 August. During 2000 and 2002, it predicted the pollen severity with an accuracy of 62.5%.

[1]This allergen present in Birch pollen is now well known as a source of allergy

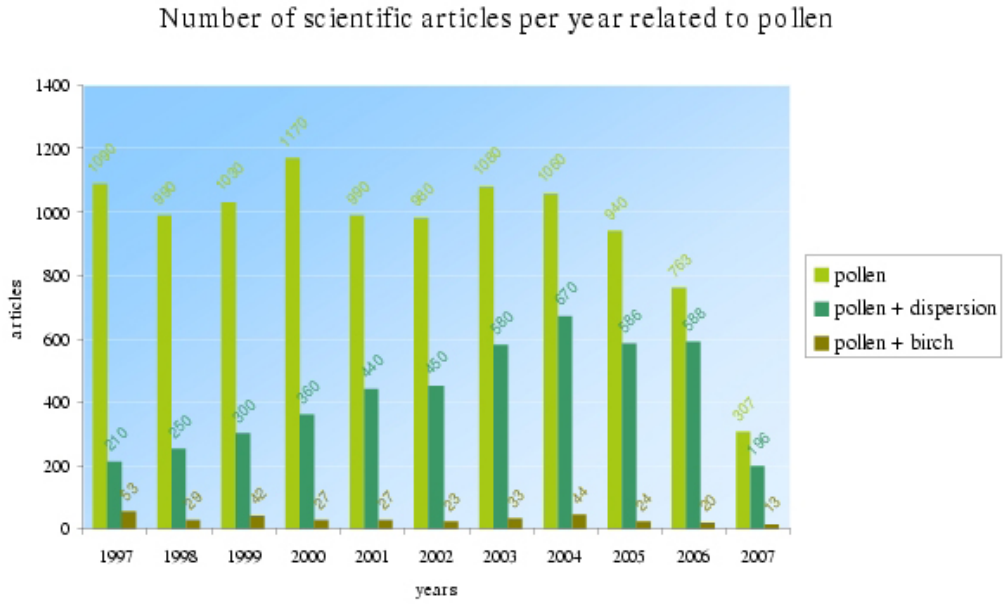

Figure 4.1: A little study on the google website *http://scholar.google.com* proves the increase of scientific participation on this field. Indeed, a research on the words *pollen* and *dispersion* reveals an increase of fifty articles per year for the last nine years that can be explained by the hay fever rising. One can notice that articles about pollen in general keep constant.

4.1.1 Model from LUMC

In The Netherlands, the LUMC has its own statistical forecast model. This model has been set up thirty years ago in Leiden and is still used to predict the severity of hay fever during pollen seasons. To perform this study, Frits Spieksma used a sample of 150 hay fever patients and found some correlations with weather. He classified weather conditions in 10 different categories and indexed complaints. By dividing the pollen season of grass in four parts, he obtained the model described on figures A and A on Appendix A page p.48.

This simple model presents some limitations since it does not take in consideration any plant phenology or atmospheric physical process. Thus, it can be use only for The Netherlands. However, it shows a forecast skill and got 72%, 85% and 88% of accuracy in 1977, 1978 and 1979 respectively[2].

4.2 Numerical models

Numerical models for pollen forecasting have been developed by groups in Germany, Finland and USA. The German article *"Numerical modelling of pollen dispersion on the regional scale"* [16], written by Helbig et al., presents a general approach to model pollen behaviour and emissions. Another one also coming from Germany deals with the dispersal of oak pollen near Hamburg [17]. But the most relevant one for our study is *"Towards numerical forecasting*

[2]This accuracy represents the percentage of days with correct hay fever forecast for days with correct weather forecast.

of long-range air transport of Birch pollen: theorical consideration and feasibility study" [18], written by Mikhail Sofiev et al. This study is dedicated to emission and dispersion of Birch pollen and can be then used as a starting point for our study.

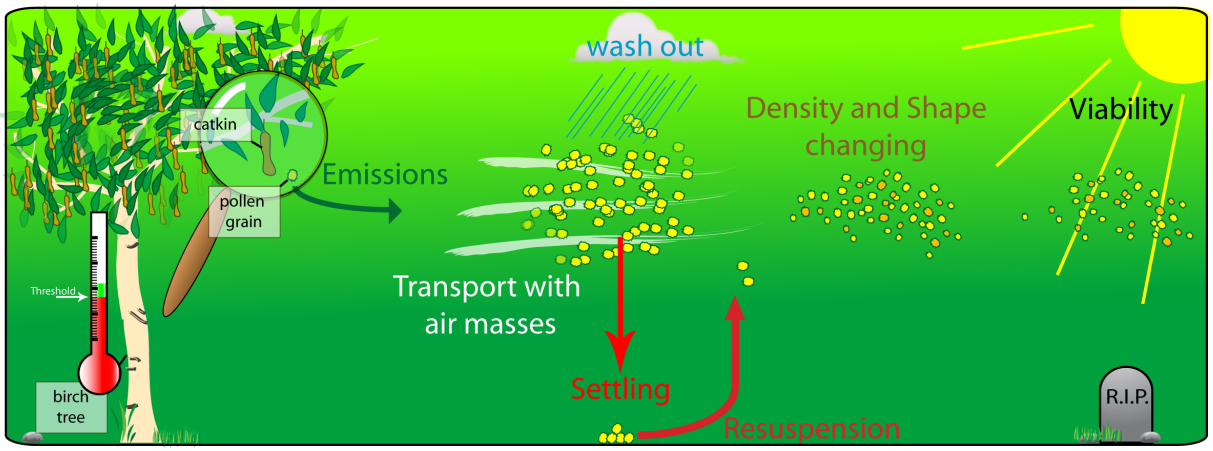

Figure 4.2: Atmospheric processes of relevance for pollen grains.

4.3 The Birch tree and its emission process

In northern latitudes, Birch is considered to produce the most important allergenic tree pollen. Fifteen to twenty percent of the population is assessed to be sensitive to Birch pollen grains and to suffer from hay fever. As a result, several studies are conducted in this field. However, the phenology of Birch trees is complicated and it remains a challenge to predict the total amount of pollen or the emission rate for a given season. Therefore most of the models are still based on statistical data. For instance, Hogda and al. [12] and Adams-Groom and al. [15] showed a link between meteorological phenomena and pollen shedding. The most relevant parametrization seems to be the growing degree day also called heat sum. It is a model for the timing of the blossoming and amount of pollen released that takes into account the summation of the excess temperature starting from a predefined fixed date. The excess temperature is the daily mean temperature minus a fixed predefined temperature threshold, and is added only when positive. For Birch trees, one generally chooses the first of March as starting date, but the temperature threshold is different depending on the definition of the heat sum and the geographical location. For example, Mikhail Sofiev uses 3.5 °C as a threshold for hourly degree days[3] whereas Arnold Van Vliet uses 2.3 °C as a threshold for daily degree days. In these studies, the emission model is often also dependent on the humidity, wind or precipitation. Again, the relevancy of this information is uncertain since the likelihood that these meteorological parameters can influence emissions is unknown. Finally for our study, it will be better to consider, as a first step, the influence of summed temperature only:

- Heat sum starting the first of March

[3]Value founded on Pollen SILAM project website

And then, to add the triggers implemented by Mikhail Sofiev in the SILAM model[4] in order to check the importance of them:

- Humidity For humidity below 50%, there is full release of pollen. For intermediate humidity between 50% and 80%, the release of pollen is given by:

$$e = e_0 \frac{80 - h}{80 - 50}$$

  For humidity above 80%, there is no release of pollen

- Wind

$$e = e_0 \left(5 - e^{-\frac{w_{10m} + v_{conv}}{1.5}}\right)$$

- Precipitation

$$e = e_0 \left(1 - \frac{p}{0.5}\right)$$

where e stands for the variable of emission, e_0 stands for the current value of emission, h represents the relative humidity in %, w_{10m} is the wind at 10 meters in [m/s], v_{conv} is the convective wind in [m/s] and p represents precipitation in [mm/h].

4.3.1 Degree days

Degree days is a useful concept to evaluate the temperature sum and it is employed to calculate the emission of pollen grains during a season. It is a cumulative function of day temperatures with threshold characterized by the following statement:

$$D = \begin{cases} \sum_{d=d_s}^{d_e} \left(\overline{T_d^\circ} - T_{threshold}\right) & \text{if } \overline{T_d^\circ} > T_{threshold} \\ 0 & \text{instead} \end{cases} \tag{4.1}$$

Where $\overline{T_d^\circ}$ is the mean temperature of the current day, $T_{threshold}$ is a threshold parameter to calculate the heat sum, d is a loop variable running over days, d_s characterizes the first day of the cumulative sum and it is generally pollen and tree specific. For Birch trees, this starting date is the first of March. d_e stands for the end date of the degree day calculation.

For degree hours, the concept is the same but it uses hours instead of days. Obviously, the temperature threshold will be different.

4.4 The Birch pollen and its key behaviours

It is generally assumed that pollen behaviour is linked with meteorology and physics because pollen is sensitive for the meteorological conditions like wind, rain and temperature and also sensitive to physical phenomena like settling velocity [2]. For example, the article

[4]In this case, triggers were defined by hourly meteorological data.

about *"Transport of airborne pollen into city of Thessaloniki"* [1] shows the impact of wind direction, speed and persistence on pollen. However, statistical parametrizations still present weaknesses, since we do not know correctly how plant and tree work. Even if the Greek study [1] demonstrates the importance of wind speed, we have to be aware that this model has been run over only three years. Others aspects like shape changes, density changes and resuspension are assumed to occur during the flying phase. These additional aspects are often not implemented in models. One can identify the following main pollen processes (discussed in the following sections):

- Transport with air masses
- Settling velocity
- Wet scavenging

Additional processes, for future implementation and a more detailed approach are:

- Density changing and shape changing
- Resuspension
- Viability

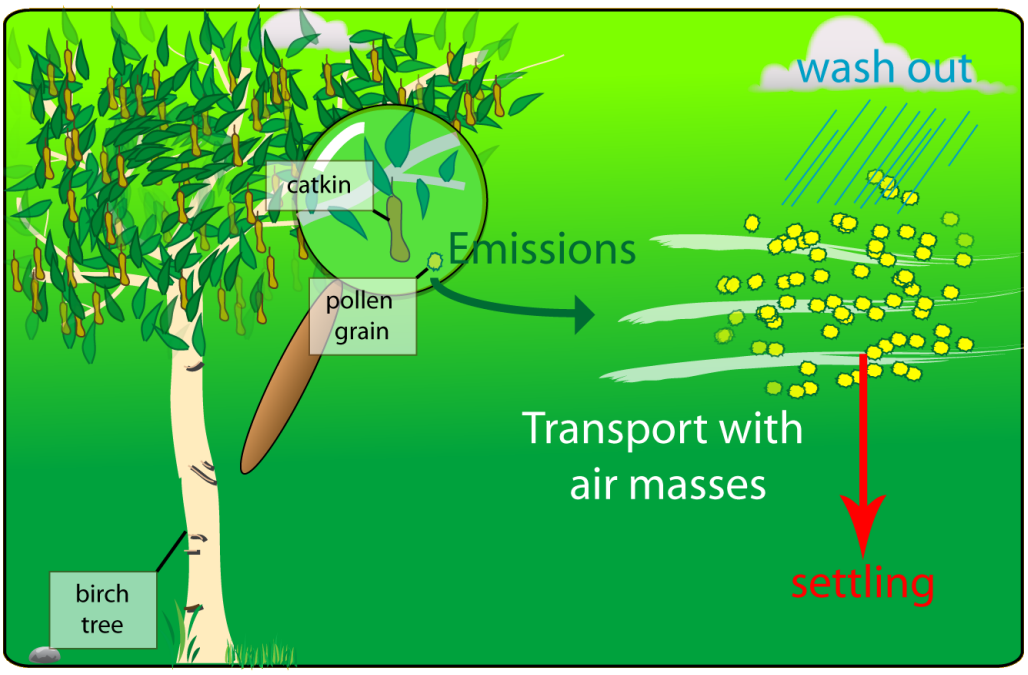

Figure 4.3: The three main behaviours of pollen grains in the atmosphere.

Other aspects of the pollen grains are the diameter and the density. Several articles mention a density value in a range between 800 and 1000 kg/m^3 and about 20 μm for the diameter. Thus, a pollen grain can be considered as an one-mode aerosol ([18], [16]).

4.4.1 Transport with air masses

Pollen atmospheric movement is a complex phenomenon influenced by numerous environmental parameters (described later) with wind playing a major role. Moreover, pollen grains can travel over hundreds of kilometres (*Long distance transport of pollen to Greenland*, Rousseau et al., 2003). The distance pollen travels, once released into the air is closely related to the wind. As pollen can be thought of an aerosol, it is then just advected by wind. Therefore, a simple advection equation combined with a suitable numerical scheme will be able to represent this process.

4.4.2 Settling

Settling is the process by which particles drift to the surface as a result of gravity to the bottom of the atmosphere. Consequently, settling velocity is the rate at which these particles settle. In the case of pollen, it remains a significant parameter [2, 18] and all dry deposition processes can be neglected since the aerodynamic resistance and surface resistance are very small [18]. A settling velocity model has been used by Donald E. Aylor in 2002 [2]. In this study, he derived the following equation for the settling velocity:

$$v_s^2 = \frac{4\rho_p D_p g}{3\rho_a c_d(v_s)} \tag{4.2}$$

Where v_s is the settling velocity, ρ_p is the density of the pollen grain, D_p is the diameter of the pollen grain, g is the gravity constant, ρ_a is the air density and c_d is the drag coefficient. This parametrization was used again by Helbig and al. in 2004 [16] on Hazel and Alter pollen with the KAMM/DRAIS dispersion model.

In 2006, Schueler et al. modeled oak pollen dispersion with the METRAS mesoscale atmospheric model. They showed settling was mainly related to sedimentation[5]. Thus, the dry deposition can be neglected. They chose a constant and standard sedimentation velocity of $2.9cm/s$.

Another study, lead by Mikhail Sofiev in Finland [18] presented the feasability of numerical simulation of large-scale atmospheric transport of allergenic pollen. Using the SILAM model, it showed Birch pollen can stay in the air for a few days (corresponding to a characteristic scale of about one thousand kilometres). A key assumption is that the pollen is transported together with air mass and so follow the airflow. For velocity gradients smaller than $10^{-3}m.s^{-1}.m^{-1}$, the grain can be considerated as noninertial which means that grains will follow air fluxes as soon as air velocity gradients are above $10^{-3}m.s^{-1}.m^{-1}$.

$$v_s = \frac{g\rho_p D_p^2}{18\eta} \tag{4.3}$$

Where v_s is the settling velocity, g represents the gravity, ρ_p is the density of the pollen grain, D_p is the diameter of the pollen grain and η is dynamical viscosity.

Contrary to their shape, these two expressions represent the same phenomenon. Actually, the drag coefficient depends on the size of pollen grain. For particles bigger than about $20\mu m$, the Reynolds number becomes too large and then the assumption of Stokes collapses. Consequently, another formula for the drag coefficient has to be employed.

[5]Sedimentation describes the motion of particles in suspension in response to gravity.

4.4.3 Wet scavenging

All studies assume wet scavenging parametrization for pollen is the same as for aerosol from a numerical point of view. Thus, a simple coefficient $\Lambda(d_p)$ is used to represent this process. This coefficient called scavenging coefficient describes the rate of removal of particles of diameter d_p and is generally given in h^{-1}.

4.4.4 Resuspension

According to Aylor (1976), resuspension occurs when grains has been deposited and are moved back to the atmosphere by strong wind gusts or local acceleration of air flow. Helbig et al. [16] used this previous study to parametrized the vertical resuspension flux.

$$F_r = c_r \cdot K_r \cdot c \cdot v_s \tag{4.4}$$

Where c_r is a pollen specific and plant-specific parameter, K_r is the meteorological limitations for resuspension, c is the density of pollen grains close to the surface and v_s the settling velocity.

4.4.5 Density changing and shape changing

The density and shape changing charaterize the ability of pollen grains to change their density and shape during their stay in the atmosphere. Initially, a pollen grain contains a lot of water. However, because of water uptake, its density and shape get modified during the flight. Consequently, they affect directly the settling velocity. The density of the pollen grain becomes:

$$\rho_p = \rho_{H_2O} \cdot \left(1 - \left(\frac{d_d}{d_e}\right)^3\right) + \rho_{solid} \cdot \left(\frac{d_d}{d_e}\right)^3 \tag{4.5}$$

Where ρ_{H_2O} is the density of water, d_d is the diameter of the dry pollen grain and ρ_{solid} is the density of the solid material of the pollen grain. d_e is the diameter of wet pollen grain and depends on water uptake. However, in Helbig et al. a constant value was used.

4.4.6 Viability

Viability stands for the capacity of pollen grains to keep its reproductive property during and after transport in the atmosphere. This behaviour has only been implemented recently in Schueler et al. [17]. The study of the biological effectiveness of pollen grain has been conducted previously also by Schueler et al. in 2004. The group showed the viability of pollen grains depends on the flight conditions and the duration of the transport (*Viability and sunlight sensitivity of oak pollen and its implications for pollen-metdiated gene flow*, Schueler et al.,2004). They used the following parametrization:

$$F_s = e^{\left(\frac{-r_s \delta t R_{SW}}{250}\right)} \tag{4.6}$$

Where F_s is the pollen survival rate, r_s characterizes the pollen sensitivity to sunlight, δt is the model time step and R_{SW} is the incoming short wave radiation. We can think that this term could be related to allergy issues.

MODEL	TRANSPORT	SETTLING	WASH OUT	RESUSPENSION	DENSITY & SHAPE	VIABILITY
Schueler et al.	o	o	o	x	x	o
Sofiev et al.	o	o	o	x	x	x
Helbig et al.	o	o	x	o	o	x
Pasken et al.	o	o	o	o	x	x

Table 4.1: Different processes implemented in numerical models. (O means Yes, X means No)

4.5 Meteorological and air quality framework

Obviously, some meteorological and air quality knowledge is useful for this project. The predominant wind direction in The Netherlands is south-west, which causes a moderate maritime climate, with cool summers and mild winters. In The Netherlands, during the Birch pollen season, the climate is in general between the two seasons so every type of weather can occur. The purpose of this section is not to give a course about meteorology but in order to facilitate the reading of this article for the persons who are not comfortable with this subject, here comes some background informations inspired by the course of Laura Gallardo-Klenner [30]. Basically, the atmosphere is governed by chemical and physical laws, and more precisely, by thermodynamic and fluid mechanic laws. Thus, the concentration of a species in the atmosphere is generally assumed to be described by the followed partial differentiate equation of continuity:

$$\begin{cases} \dfrac{\partial c}{\partial t} = -\vec{v} \cdot \nabla c - c\nabla \cdot \vec{v} - \nabla \cdot (\langle c'\vec{v}' \rangle) + P - L \\ + \text{ Initial conditions} \\ + \text{ Boundary conditions} \end{cases}$$

Where c is the concentration of the tracer, v is the velocity of the air, P is the production term and L is the loss term.

In details, $-\vec{v} \cdot \nabla c$ represents the advection transport, $-c\nabla \cdot \vec{v}$ defines the convergent and divergent aspects of air, $-\nabla \cdot (\langle c'\vec{v}' \rangle)$ is the term for small-scale eddies and diffusion which can not be resolved by the discretisation chosen in the numerical model, P represents mainly all types of emission and L represents deposition and chemical processes.

Chapter 5

Modelling the behaviours and processes of Birch pollen and trees

We decided to focus on the main processes to model Birch pollen, namely: emissions determined by heat sum, transport with air masses, settling and wash out.

Small studies have been done to better understand the behaviour of pollen grains. Some of them required meteorological data, that is why we did a little statistical study of meteorological data over the pollen season (refer to Appendix C page p.51).

5.1 Impact of transport

5.1.1 Basic transport analyses

Considering the advection process only, it is useful to check the travelling distance of pollen grains.

Wind [m/s]	Travelling distance in 24 hours [km]
2	173
4	346
6	518
8	691
12	1037

Table 5.1: Range of value for pollen transport. The grains were taken a a maximal height of 1000 metres which corresponds to the top of the boundary layer

Usually, the wind at the surface in The Netherlands is about 4.5 meters per second[1]. Then, considering a constant horizontal wind, no rain and the settling velocity calculated by M. Sofiev [18], we realize that pollen can travel over large distances.

[1] Average calculated over the birch pollen season period

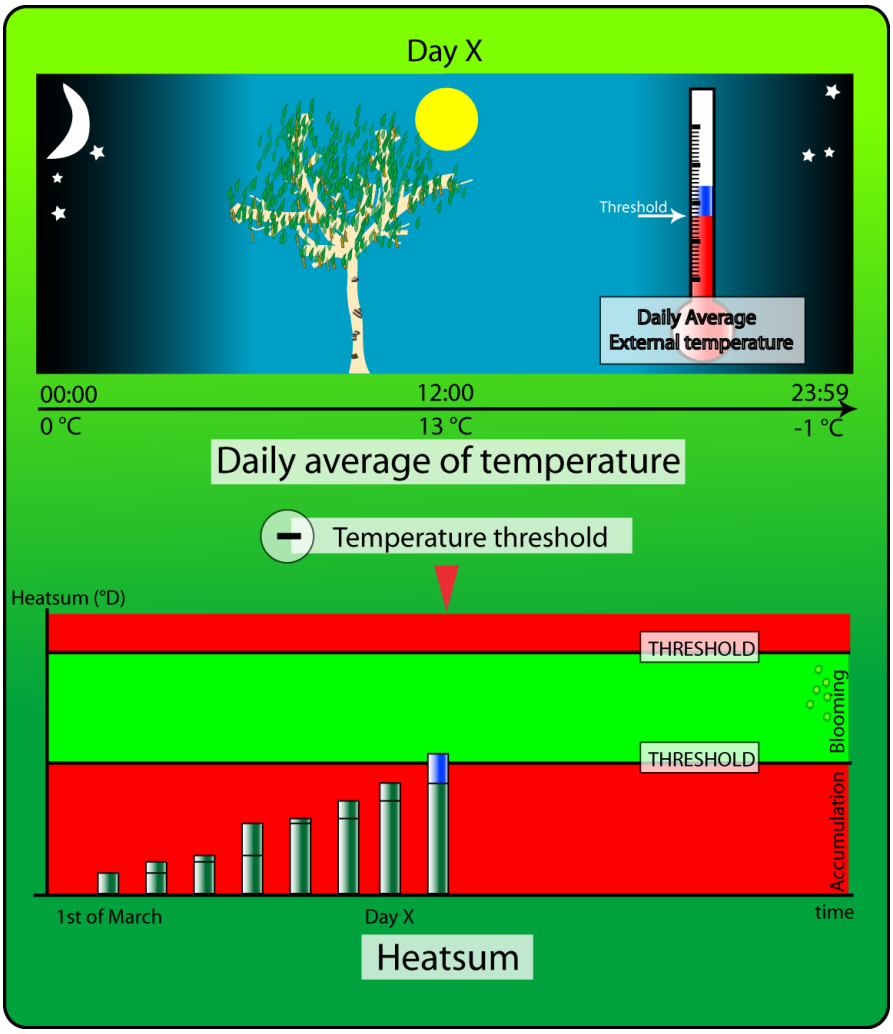

Figure 5.1: A general heat sum model with degree days. Example taken for a day X. The blue parts corresponds to the current value of the degree day, the difference between the daily-mean temperature and a pre-defined threshold. The upper part shows a day X and its corresponding heat sum value. The bottom part presents the heat sum, which is an accumulation of the degree day values. When this sum is between two thresholds the tree is assumed to be blooming.

5.1.2 Enhanced transport analyses

Within the atmosphere, pollen grain movement is governed by Navier-Stokes equations. Assuming air is an incompressible Newtonian fluid and its corresponding Reynolds number is small ($Re << 1$), one can obtain the drag force exerted by the fluid on the pollen grain which is also known as Stoke's law. Moreover, gravity acts also on pollen grains. Then by applying Newton's law:

$$\rho_p V_p \frac{d\vec{v}}{dt} = F_{drag} + F_{grav} = \frac{3\Pi\mu D_p}{Cc}(\vec{u} - \vec{v}) + \rho_p V_p \vec{g} \tag{5.1}$$

Where μ is the viscosity of air, ρ_p is the density of a pollen grain, V_p is the volume of a

pollen grain, D_p is the diameter of a pollen grain, Cc is a correction factor, u is the velocity of air and v is the velocity of a pollen grain.

This equation can be rewritten as:

$$\tau \frac{d\vec{v}}{dt} = \tau \vec{g} + u - v$$

with

$$\tau = \frac{\rho_p V_p Cc}{3\Pi \mu D_p}$$

Assuming $\vec{u} = 0$ and $t >> \tau$, namely the particle reaches its terminal settling velocity $v_t = \tau g$, we have:

$$v_t = \frac{D_p^2 \rho_p g Cc}{\mu}$$

We can realize we find the expression given by Mikhail Sofiev in his study.

The diameter of birch pollen grains are assumed to be $20\mu m$ and the density is estimated at $800 kg/m^3$. Moreover, the viscosity of air is calculated by the following equation, derived from the Sutherland's equation (valid between $0 < T < 555K$):

$$\eta = \eta_0 \frac{T_0 + C}{T + C} \left(\frac{T}{T_0} \right)^{3/2} \tag{5.2}$$

where η is the viscosity $[Pa.s]$ at input temperature T, η_0 is the reference viscosity $[Pa.s]$ at reference temperature T_0, T represents the input temperature in kelvin, T_0 defines the reference temperature in kelvin and C is the Sutherland's constant. Here,

C $[K]$	T_0 $[K]$	η_0 $[10^{-6} Pa.s]$
120	291.15	18.27

Then, for a range of temperatures values between 0 and 20 $°C$ (273.15 and 293.15 $°K$), it appears that the value of pollen settling velocity is about 1.2 cm/s.

As a result, under typical weather conditions, pollen grains can travel over very long distance and can reach in particularly cases thousands of kilometers. Finally, we can see that boundary conditions are crucial for pollen studies.

5.2 Effect of rain

Precipitation scavenging of particles is assumed to be given by the following expressions:

$$\frac{dM_p}{dt} = -\Lambda M_p$$

$$\Lambda = \frac{\pi}{4} D_p^2 U_t(D_p) E(D_p, d_p) N_D$$

Figure 5.2: A disk of 1,000 kilometres radius around Amsterdam reveals the ability of pollen grains from other countries to reach The Netherlands. Thus, boundary conditions are extremely important in pollen studies.

$$p_0 = \frac{\pi}{6} D_p^3 U_t(D_p) N_D$$

Where M_p defines the mass of pollutant, Λ is the scavenging coefficient, D_p is the diameter of the rain drop, U_t is the terminal velocity of the rain drop, E is the collision efficiency , N_D is a number of concentration and p_0 is the rainfall intensity.

For example, assuming a rainfall intensity of 1 mm h^{-1}, a raindrop size of 2 mm and a particle size of 20 μm, the scavenging coefficient Λ is 0.8 h^{-1}. So, a relaxation time of 1 hour and 15 minutes, which determines a characteristic time after which sixty percent of pollen grains have been removed in the rainy area.

5.3 Predicting starting date of emissions and pollen release rate

Several studies deal with the time of pollen emissions but a lack of information clearly exists. Nevertheless, the MSA-WUR has a model as well as the FMI. The first one implements a basic but efficient heat sum model whereas the second one also introduces heat sum combined with weather triggers which influence the pollen release rate.

5.3.1 Model from MSA-WUR

Arnold Van Vliet proposed a basic model based on heat sum only to calculate the percentage of pollen grains released in the atmosphere. Then for some heatsum values, we know the quantity of pollen released in air and can make basic linear interpolations between these integrated points (cf. 5.3).

Percentage (%)	Temperature (Celsius)
1	147
5	187
10	206
20	223
25	230
30	236
40	247
50	260
60	274
70	289
75	302
80	315
90	359
95	435
99	833

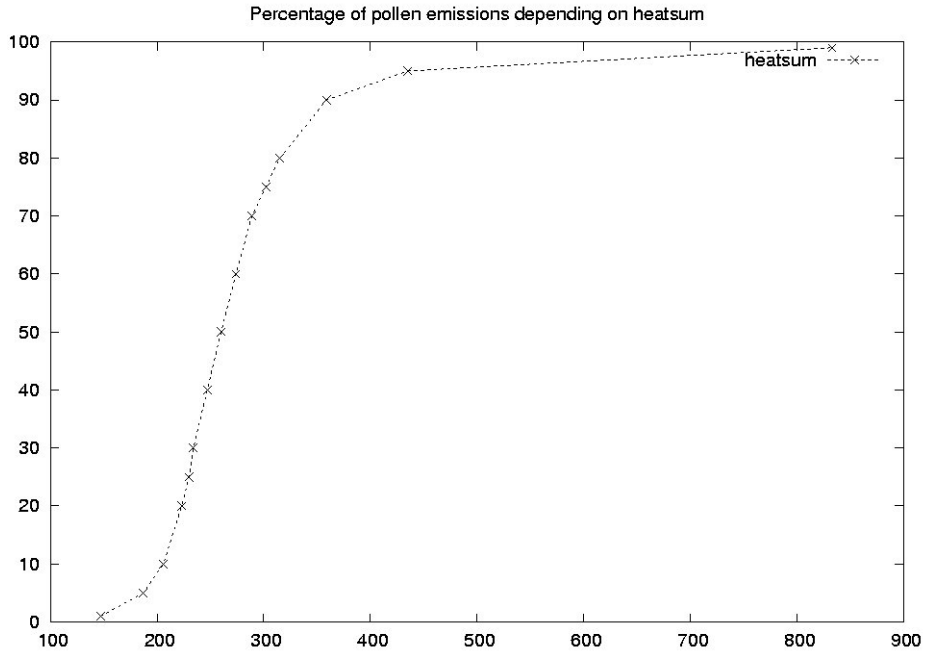

Figure 5.3: The heat sum model proposed by the WUR with linear interpolations between points.

Then, we extrapolate linearly the extremities in order to estimate the values at 0% and 100%. We find respectively 137°C and 932.5 °C.

For instance, for the value at 0%:

$$y = ax + b$$

$$a = \frac{5-1}{187-147} = 0.1,\ b = 1 - 0.1 \times 147 = -13.7$$

Then,

$$x = \frac{y-b}{a} = \frac{0+13.7}{0.1} = 137$$

5.3.2 Model from FMI

Mikhail Sofiev from the Finnish Meteorological Institute and his team have also developed an emission model. This one is more complex since it takes in consideration, not only heat sum, but also meteorological parameters such as: wind, precipitations and humidity. Thus, we get the following model:

$$C = T_{2m}$$

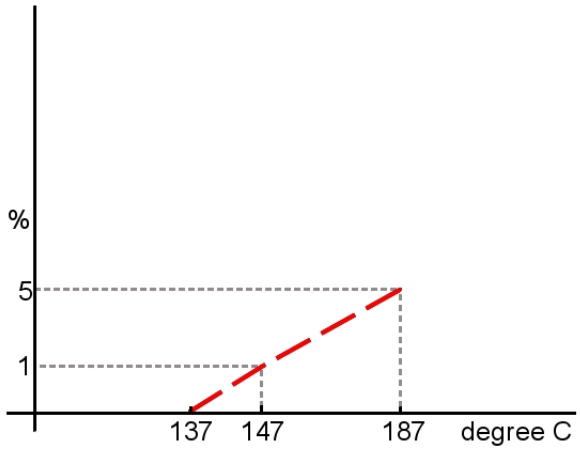

Figure 5.4: Extrapolation method to find boundaries.

$$\times \left(w_{maxScale} * e^{-\frac{(w_{10m}+v_{conv})}{w_{satur}}} \right)$$
$$\times (1 - p/p_{thresh}) \, ()$$
$$\times P_{totalamount} \times C_{term}$$

where T_{2m} is the temperature at 2 meters, $w_{maxScale}$ represents the maximum impact of wind speed (its value is 1.5), w_{10m} is the wind at 10 meters, v_{conv} is the convection velocity, w_{satur} is the saturation level of wind speed (its value is $5m/s$), p stands for the precipiations, p_{thresh} is the precipitation threshold (its value is $0.5mm/h$), $P_{totalamount}$ is the total amount of pollen release at a given day and C_{term} is an annual corrective term.

We have to be careful with these triggers. Studies showed more or less good correlation between observations pollen concentrations in the atmosphere and weather data, but it does not mean that pollen emissions are indeed affected by these parameters.

Chapter 6

The Chimere model from a numerical point of view

6.1 General presentation

The Chimere model is an eulerian chemical transport model designed to forecast concentrations of atmospheric pollutants such as ozone or aerosols over a determined domain. This domain belongs to a range of spatial scales from several thousand kilometres (regional scale or mesoscale) to hundreds of kilometres (urban scale) with resolutions between 1 kilometre and 100 kilometres and usually use chemical timesteps of 10 minutes while interpolating meteorological parameter fields over one hour to ensure a good result accuracy. Moreover, Chimere proposes many different options for simulations which make it also a powerful research tool for testing parameterizations and assumptions (refer to figure 6.1).

6.2 Numerical solvers

The purpose of this section is not really to study the schemes in detail, but just to give some relevant information to understand better the results of Chimere.

6.2.1 Time solver: TWOSTEP method

The numerical method for the temporal solution of the stiff system of partial differential equations is adapted from the second-order TWOSTEP algorithm originally proposed by Verwer in 1994 [29] for gas phase chemistry only. It is based on the application of a Gauss-Seidel iteration scheme to the 2-step implicit backward differentiation (BDF2) formula:

$$c^{n+1} = \frac{4}{3}c^n - \frac{1}{3}c^{n-1} + \frac{2}{3}\Delta t R\left(c^{n+1}\right) \tag{6.1}$$

with c^n being the vector of chemical concentrations at time t_n, t the time step leading from time t_n to t_{n+1} and $R(c) = \dot{c} = P(c) - L(c)c$ the temporal evolution of the concentrations due to chemical production and emissions (P) and chemical loss and deposition (L). Note

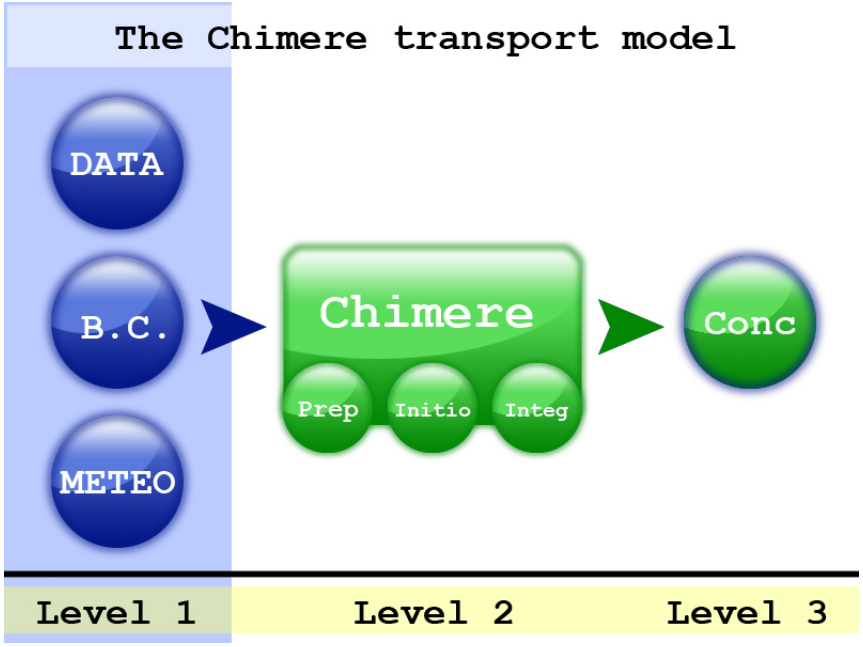

Figure 6.1: The Chimere model. Data such as meteorological data (*meteo*), boundary conditions (*B.C.*), aerosol and gases characteristics (*data*) are provided to the main model to be treated (*prep*). In a second stage, an initialization (*initio*) procedure sets up all variables and creates output files. Finally, the integration (*integ*) algorithm is runned over the number of hour simulation to generate concentrations of species (*conc*).

that L is a diagonal matrix here. After rearranging and introducing the production and loss terms this equation reads:

$$c^{n+1} = \left(I + \frac{2}{3}\Delta t L\left(c^{n+1}\right)\right)^{-1}\left(\frac{4}{3}c^{n} - \frac{1}{3}c^{n-1} + \frac{2}{3}\Delta t P\left(c^{n+1}\right)\right) \tag{6.2}$$

The implicit nonlinear system obtained in this scheme can be solved pertinently with a Gauss-Seidel method (Verwer, 1994 [29]). In CHIMERE, the production and loss terms P and L in equation are replaced by the modified terms $\tilde{P} = P + P_h + P_v$ $\tilde{L} = L + L_h + L_v$, respectively. P_h and P_v denote the temporal evolution of the concentrations due to horizontal (only advection) and vertical (advection and diffusion) inflow into the concerned grid box, L_h and L_v the temporal evolution due to the respective outflow divided by the concentration itself. With this integration scheme, at each time step all physical and chemical processes are updated simultaneously. The "operator splitting" technique (e.g. [McRae et al., 1982], which is still a standard way of solving the 3D transport-chemistry-problem, and the associated "splitting error" are therefore avoided. Further advantages of the scheme are its stability even for quite long time steps due to the implicitness of the formulation and the simplicity of the code which facilitates the development of secondary models (adjoint, tangent linear) enormously. In practice, time stepping has two time steps: A coarse time step, where all physical variables are calculated, and a refined step in order to gain accuracy on chemistry. During loops over fine steps, physical variables are maintained constant. In the provided version for the continental models, we chose a "very quick formulation", with a 10-minute

physical step, no sub-chemical steps, i.e. all processes are stepped to 10 minutes, with only 1 Gauss-Seidel iteration. It is strongly recommended, in order to have more accurate results, to use 2 Gauss-Seidel iterations, which increases by 2 the computer time.

6.2.2 Spatial solver: PPM method

The PPM scheme is a higher-order extension of Godunov's method of a type first introduced by Bram van Leer with the MUSCL algorithm. It represents a substantial advance over both these versions of MUSCL in several respects. First, the introduction of parabolæas the basic interpolation functions in a zone allows for a more accurate representation of smooth spatial gradients. as well as steeper representation of captured discontinuities, particularly contact discontinuities. Second, the representation of the nonlinear wave interactions used to compute fluxes is substantially simpler, giving rise to a less complicated and more robust algorithm.
To solve a linear advection equation:

$$\begin{cases} \dfrac{\partial a}{\partial t} + u\dfrac{a}{\xi} = 0 \\ a(\xi, 0) = a_0(\xi) \end{cases} \tag{6.3}$$

The advection scheme is then constructed following the approach to advection taken by Bram van Leer in his "Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection" article. As we know explicitly the exact solution of equation (6.3) , the integrated average value for a cell grid is given by:

$$\begin{cases} a_j^{n+1} = \dfrac{1}{\Delta\xi_j}\displaystyle\int_{t_{j-1/2}}^{t_{j+1/2}} a\left(\xi - u\Delta t, t^n\right) d\xi \\ \Delta\xi_j = \xi_{j+1/2} - \xi_{j-1/2} \end{cases} \tag{6.4}$$

Where $a\left(\xi - u\Delta t, t^n\right)$ is a piecewise polynomial interpolation function that replaces the average value of the solution a and constrained in such a way that no new extrema appear in the interpolation function which do not appear in the a_j^n's and Δt satisfies the stability condition $u\Delta t \leq min_j \Delta\xi_j$.
Then the scheme is uniquely determined by our choice of interpolation polynomial. The PPM scheme uses an interpolation which is piecewise continuous, with a given by a parabolic profile in each cell as:

$$\begin{aligned} a(\xi) &= a_{L,j} + x\left(\Delta a_j + a_{6,j(1-x)}\right) \\ x &= \frac{\xi - \xi_{j-1/2}}{\Delta\xi_j},\ \xi_{j-1/2} \leq \xi \leq \xi_{j+1/2} \end{aligned} \tag{6.5}$$

Where $a_{L,j} = lim_{\xi\to\xi j-1/2}$, $a_{R,j} = lim_{\xi\to\xi j+1/2}$, $\Delta a_j = a_{R,j} - a_{L,j}$ and $a_{6,j} = 6(a_j^n - \frac{1}{2}(a_{L,j} + a_{R,j}))$
The values $a_{L,j}$ and $a_{R,j}$ are computed before using $a_{j+1/2}$ the value of a at $\xi_{j+1/2}$ as follow:

$$a_{j+1/2} \quad = \quad a_j^n + \frac{\Delta\xi_j}{\Delta\xi_j + \Delta\xi_{j+1}}\left(a_{j+1}^n - a_j^n\right) + \frac{1}{\sum_{k=-1}^{2}\Delta\xi_{j+k}}$$

$$\times \left\{ \frac{2\Delta\xi_{j+1}\Delta\xi_j}{\Delta\xi_j + \Delta\xi_{j+1}} \left[\frac{\Delta\xi_{j-1} + \Delta\xi_j}{2\Delta\xi_j + \Delta\xi_{j+1}} - \frac{\Delta\xi_{j+2} + \Delta\xi_{j+1}}{2\Delta\xi_{j+1} + \Delta\xi_j} \right] \left(a^n_{j+1} - a^n_j\right) \right. \tag{6.6}$$
$$\left. - \Delta\xi_j \frac{\Delta\xi_{j-1} + \Delta\xi_j}{2\Delta\xi_j + \Delta\xi_{j+1}} \delta_m a_j + \Delta\xi_{j+1} \frac{\Delta\xi_{j+1} + \Delta\xi_{j+2}}{\Delta\xi_j + 2\Delta\xi_{j+1}} \delta_m a_{j+1} \right\}$$

Where δ_j is the average slope in the jth cell of the parabolas and is given by:

$$\begin{aligned} \delta_m a_j &= min\left(|\delta a_j|, 2|a^n_{j+1} - a^n_j|, 2|a^n_j - a^n_{j-1}|\right) sgn(\delta a_j) \\ &\quad \text{if } \left(a^n_{j+1} - a^n_j\right)\left(a^n_j - a^n_{j-1}\right) > 0 \qquad (6.7) \\ &= 0 \text{ otherwise} \qquad (6.8) \end{aligned}$$

with

$$\delta a_j = \frac{\Delta\xi_j}{\Delta\xi_{j-1} + \Delta\xi_j + \Delta\xi_{j+1}} \left[\frac{2\Delta\xi_{j-1} + \Delta\xi_j}{\Delta\xi_{j+1} + \Delta\xi_j} \left(a^n_{j+1} - a^n_j\right) + \frac{\Delta\xi_j + 2\Delta\xi_{j+1}}{\Delta\xi_{j-1} + \Delta\xi_j} \left(a^n_j - a^n_{j-1}\right) \right] \tag{6.9}$$

And for the simplest case, when the cells are equally spaced and $\delta a_j = \delta_m a_j$, we have:

$$a_{j+1/2} = \frac{7}{12}\left(a^n_j + a^n_{j+1}\right) - \frac{1}{12}\left(a^n_{j+2} + a^n_{j-1}\right) \tag{6.10}$$

Thus, we obtain values $a_{L,j}$ and $a_{R,j}$ as follow:

$$\begin{aligned} a_{L,j} \leftarrow a^n_j \text{ and } a_{R,j} \leftarrow a^n_j \quad & \text{if } \left(a_{R,j} - a^n_j\right)\left(a^n_j - a_{L,j}\right) \\ a_{L,j} \leftarrow 3a^n_j - 2a_{R,j} \quad & \text{if } (a_{R,j} - a_{L,j})\left(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})\right) > \frac{(a_{R,j} - a_{L,j})^2}{6} \qquad (6.11) \\ a_{R,j} \leftarrow 3a^n_j - 2a_{L,j} \quad & \text{if } (a_{R,j} - a_{L,j})\left(a^n_j - \frac{1}{2}(a_{L,j} + a_{R,j})\right) < -\frac{(a_{R,j} - a_{L,j})^2}{6} \end{aligned}$$

Now, it is easy to write down the averages of the interpolation functions:

$$\begin{aligned} f^a_{j+1/2,L}(y) &= \frac{1}{y}\int_{\xi_{j+1/2}-y}^{\xi_{j+1/2}} a(\xi)d\xi \\ f^a_{j+1/2,R}(y) &= \frac{1}{y}\int_{\xi_{j+1/2}}^{\xi_{j+1/2+R}} a(\xi)d\xi \end{aligned} \tag{6.12}$$

where y is assumed to be positive. Then :

$$\begin{aligned} f^a_{j+1/2,L}(y) &= a_{R,j} - \frac{x}{2}\left(\Delta a_j - \left(1 - \frac{2}{3}x\right)a_{6,j}\right), \qquad \text{for } x = \tfrac{y}{\Delta\xi_j} \\ f^a_{j+1/2,R}(y) &= a_{L,j+1} - \frac{x}{2}\left(\Delta a_{j+1} + \left(1 - \frac{2}{3}x\right)a_{6,j+1}\right), \quad \text{for } x = \tfrac{y}{\Delta\xi_{j+1}} \end{aligned} \tag{6.13}$$

Finally, one obtains the value of a at next time step:

$$a^{n+1}_j = a^n_j + u\frac{\Delta t}{\Delta\xi_j}\left(\bar{a}_{j-1/2} - \bar{a}_{j+1/2}\right) \tag{6.14}$$

Where

$$\begin{array}{rcll} \bar{a}_{j+1/2} & = & f^a_{j+1/2,L}(u\Delta t), & \text{if } u \geq 0 \\ \bar{a}_{j+1/2} & = & f^a_{j+1/2,L}(-u\Delta t), & \text{if } u \leq 0 \end{array} \tag{6.15}$$

Finally, this method has been chosen because of its capacity of representing correctly discontinuities without involving a loss of concentration with small numerical differences. It has also the good property to not inducing oscillations in the solution.

6.2.3 Spatial solver: UPWIND method

This method is one of the simplest methods to approximate advection and it is given by the following equation:

$$\begin{array}{l} \dfrac{u_j^{n+1} - u_j^n}{\Delta t} + c\dfrac{u_j^n - u_{j-1}^n}{2\Delta x} = 0 \text{ if } c > 0 \\ \dfrac{u_j^{n+1} - u_j^n}{\Delta t} + c\dfrac{u_{j+1}^n - u_j^n}{2\Delta x} = 0 \text{ if } c < 0 \end{array} \tag{6.16}$$

These scheme has good stability property but produces numerical diffusion.

In Chimere and for a given cell, depending on the wind direction, tracer concentrations will be deplaced to the adjacent downstream cell.

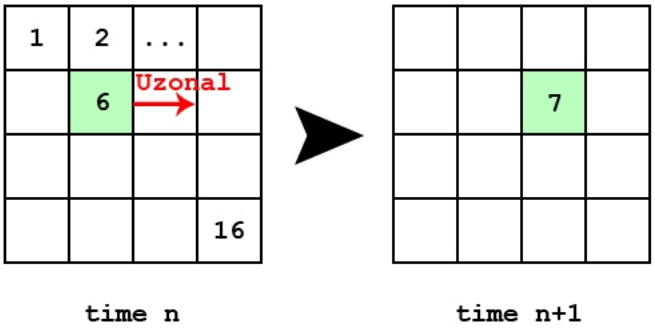

Figure 6.2: The UPWIND method. Example of interpolation, a part of the concentration present in the cell *6* is moved to the cell *7* because of the wind *Uzonal*.

6.3 Units in Chimere

The Chimere CTM uses $molecules/cm^3$ to calculate concentrations of gas and particules species. Thus, emission files have values expressed in $molecules/cm^3/s$. A transformation

occurs at the end of any Chimere runs to obtain comparable concentrations. Usually, in ppb[1] for gases and $\mu g/m^3$ for aerosols.

6.4 Parallelism

MPI[2] has been implemented in this version of Chimere[3] and let a huge gain of time. Basically, the master processor sends variables to workers whose number is equal to the number of processors minus one (refer to Appendix D page p.53).

[1] parts per billion

[2] The Message Passing Interface (MPI) is a language-independent computer communications descriptive application programming interface (API) for message-passing on a parallel computer.[33]

[3] Chimere version V200606A

Chapter 7

Implementation of a pollen tracer in Chimere

Using the Chimere model let us gain precious time. Indeed, pollen is a chemically inert aerosol and aerosols are already implemented in this model. Thus, we just had to add a few changes to the original code (refer to figure 7.1). For this first version, we used the aerosol version with sea salts[1]

7.1 Implementation of a pollen tracer

First, we implemented a new environment for the pollen tracer by creating new input data, modifying some aerosol files and setting new variables.

7.1.1 New input data

In order to calculate some processes in Chimere, we needed some information about pollen grains. All this information is stored in the following files:

- pollen/pollenProperties
- pollen/treeProperties
- pollen/pollenBehaviours
- pollen/heatsumModel.XXX
- pollen/maps/map.XXX.dat
- chemistry/POLLEN.nacl
- PEMISSIONS.nc

[1]Version meca=MELCHIOR2 mph=nacl. See chimere documentation for more details.

7.1.2 Modified aerosol files

Since we used some native aerosol routines, we had to modify a few aerosol files:

- AEROSOL.nacl
- WETS_SPEC.MELCHIOR2
- PPM_SPECIES.MELCHIOR2
- chemistry/ACTIVE_SPECIES.MELCHIOR2.nacl
- chemistry/CHEMISTRY.MELCHIOR2.nacl
- chemistry/REACTION_RATES.MELCHIOR2.aero
- chemistry/REACTIONS.MELCHIOR2.aero
- output_species/OUTPUT_SPECIES.MELCHIOR.nacl.tonest

7.1.3 New variables

Finally, we set new variables mainly present in *chimere_common.f90*:

- nemisp (number of emitted pollen species)
- npol (number of pollen species)
- idpollen (identifyer for pollen species)
- emisp (array of emissions)
- emisploc (array of emissions)

7.1.4 Initialization routines

During the initialization phase, several operations are done. The routine *inichimere* calls the initialization of aerosols and pollen, chemistry, emissions, concentrations, etc... First, the routines *inicdf* and *iniread* set some file handles which points to pollen property files and pollen emission file *PEMISSIONS.nc*. Secondly, the main pollen properties are read and imported in Chimere while pollen is initialized as an aerosol. Moreover, thanks to the variable *idpollen*, a flag is set to identify species as pollen or not. Then, emissions are read from pollen emission file and are converted into $molecules/cm^3$(Unit uses for all species in Chimere.). Finally, pollen concentrations are initialized using native Chimere initialization.

7.2 Implementation of a transport model

Secondly, we implemented a transport model for pollen. As we know now, pollen grains are mainly transported by air masses while removed by settling and rains.

7.2.1 Transport with air masses

This process uses the PPM numerical scheme already set in Chimere.

7.2.2 Settling

This process is a bit different from the native one. We implemented the parametrization used by Mikhail Sofiev in his model which is a bit simpler. For instance, there is no parametrization of collisions between pollen grains. The code is contained in the file *model/deppol.f90* (refer to Appendix E page p.54).

7.2.3 Wash out

This process uses the wash out employed for aerosols in Chimere in file *wdeposition.f90*.

7.3 Implementation of an emission model

Finally, we implemented an emission model in file *diag/diagpol.f90* that calculates emissions prior to run the model. We assumed before that pollen release is mainly governed by heat sum and has a diurnal cycle. Consequently, this script imports first pollen data like grain density and grain size and an heat sum model and a vegetation map to calculate daily heat sums and hourly emissions for the current day.

7.3.1 Daily heat sum

The heat sum model has been provided by MSA-WUR and has been extrapolated to calculate threshold points. Then, linear interpolations between points are calculated by *diag/diag_maths.f90* routine and further imported in the *diagpol* routine. For the first day, heat sum is calculated from the 1st of march of the current year and a file *HS.nc* is created. For the following days, the file is read to retrieve the previous heat sum value and calculate the next one.

7.3.2 Hourly emissions

Once the daily heat sum is calculated, *diagpol* can calculate the percentage of pollen emissions released in the atmosphere during the day using *diag/diag_science.f90*. Afterwards, the main routine reshapes the daily emissions into hourly emissions using a parabola shape. In order to stay consistent, the integral value of the parabola corresponds to the daily emission value. Finally, the value of emissions over the domain is saved in the *PEMISSIONS.nc* file.

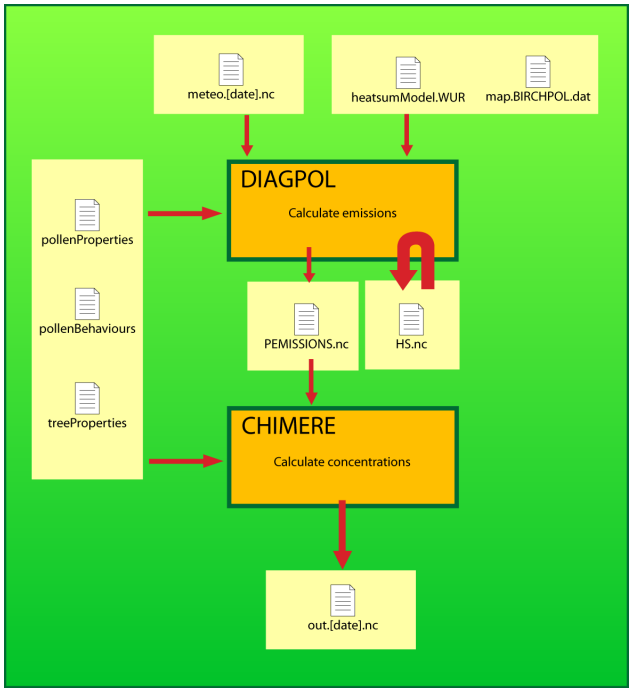

Figure 7.1: Flow of Chimere with the Pollen module.

7.4 Generating a vegetation map

Because of the difficulties to find an existing map of birch trees, we had to create one. We achieved to digitalize a picture of the birch fraction in Europe. This picture has been found on FMI pollen project website and got several transformations (refer to figure 7.2). First, we used an image editor to remove the title, the legend and all not relevant information. Then, we wrote a script in IDL and ran it to obtain a digital version of this map.

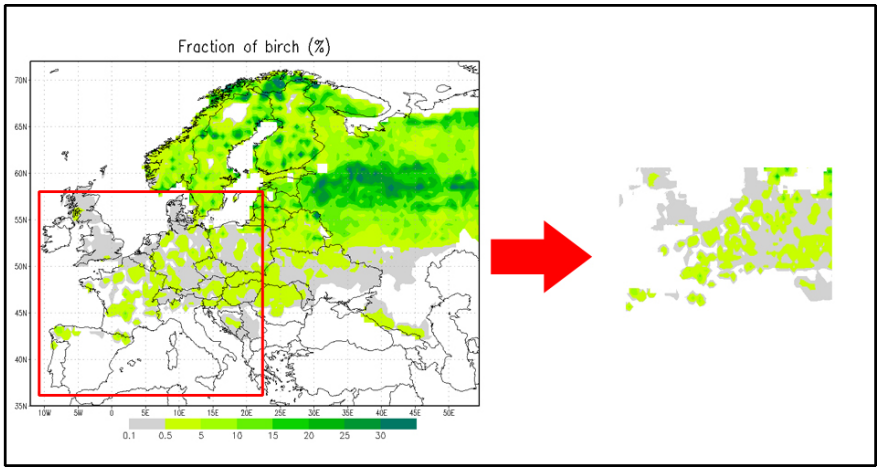

Figure 7.2: Transformation of the map processed by Mikhail Sofiev.

The MSA-WUR is currently working on Birch tree map to be used in future pollen simulations. But the work has not finished yet.

Chapter 8

Experimentations and Analysis

8.1 Heat sum

An heat sum model is local but should not vary a lot from a region to another. Consequently, we can see the trend of heat sum in Europe. Obviously, the warmer regions in the south come first to the threshold. Thus, pollination of Birch starts in the South. We can also notice at a regional scale over The Netherlands that heat comes from the ocean. In other words, pollen season starts first in the western part and then continue to the eastern part. A delay of 2 days can be see between the start in the west and the one in the east.

8.2 Pollen concentrations

A set of eleven experiments were set up. Unfortunately, only a few of them have been run because of the time needed to run simulations[1]. On the one hand, some virtual and puncutal Birch forests have been created as an input map to see relevant meteorological patterns. On the other hand, two other tests using LDU and Sofiev's map has been created to see the impact of other countries. Another important aspect of this analysis that should be highlighted is the small quantity of available observation data. Indeed, there are only two counting stations in The Netherlands (refer to Appendix B page p.50)

We chose to use a logarithmic scale to show the results. A concentration of $10gr/cm^3$ can be assimilated as a threshold for hay fever.

8.2.1 First experimentation: The lone Dutch forests

The purpose of this first experimentation is thus to show some meteorological patterns during the pollination season.

[1]About 10 hours for 1 month. A little study has been made to see the efficiency of Chimere with multiprocessoring (refer to Appendix F page p. 77)

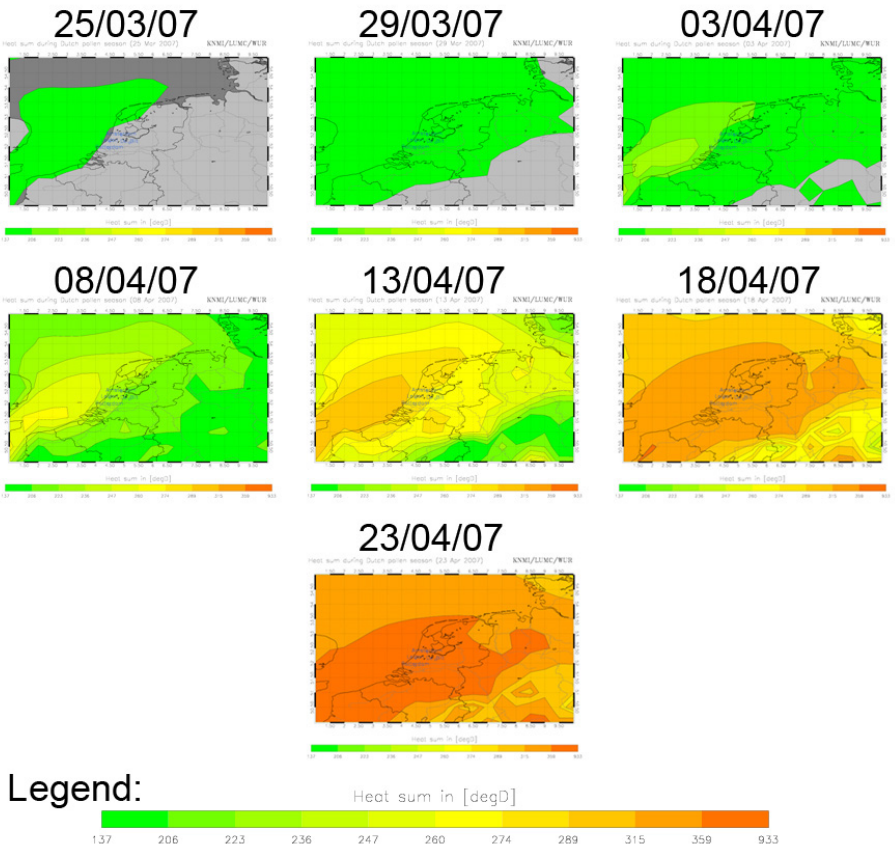

Figure 8.1: Heat sum during pollen season.

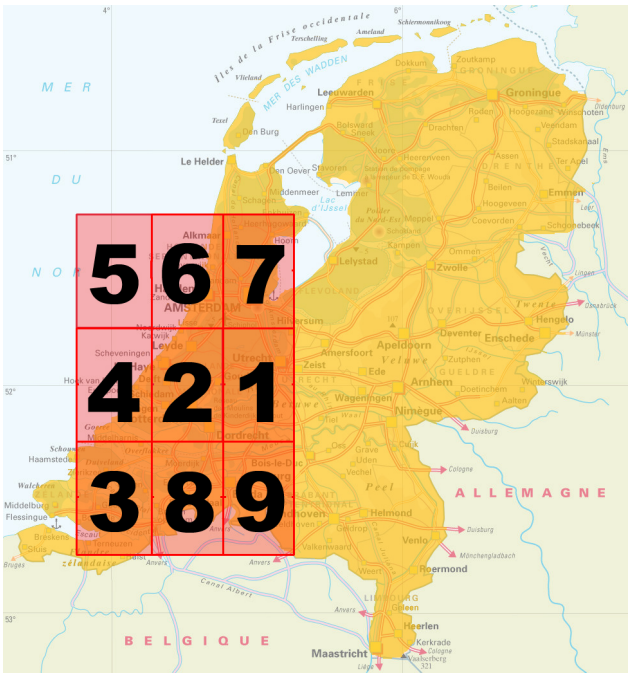

Figure 8.2: Serie of point release tests created to see typical meteorological patterns.

The lone forest in De Bilt

In that case, we run the study 1 in De Bilt and get some values in Leiden (which correspond to cell number 2 on figure 8.2). It has to be noted that this year, the weather was very sunny in April with almost no rain. As a result, it is difficult to evaluate the impact of the wash out parameter. Moreover, the wind direction was North-East at the beginning of the season, but turned to South-West around the 15^{th} of April until the end of the season. Consequently, most of the pollen should come from the eastern part of the country during this season.

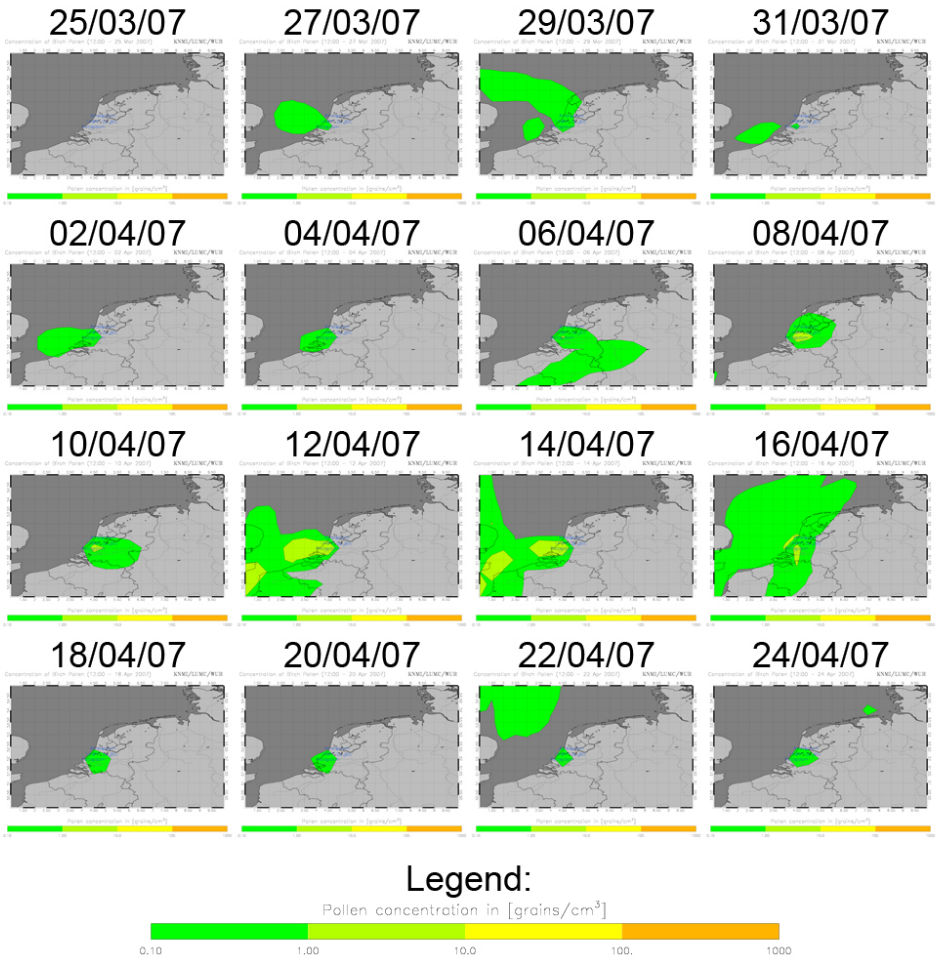

Figure 8.3: Study case n°1: One forest in De Bilt. Temporal evolution of pollen concentrations over the pollen season.

First, we can notice that the peak period occurs around the 10^{th} of April which is consistent with the reality. Secondly, we realize clearly the impact of long-range transport to other countries. That is why sensitive persons can feel sometimes the symptoms of hay fever prior to the blooming season.

Despite the factor used to amplify the simulations, we can notice directly that the two graphics present the same shape. Nevertheless, the simulation curve shows a few earlier small peaks of pollen which can be explained by the abnormal and high presence of Birch

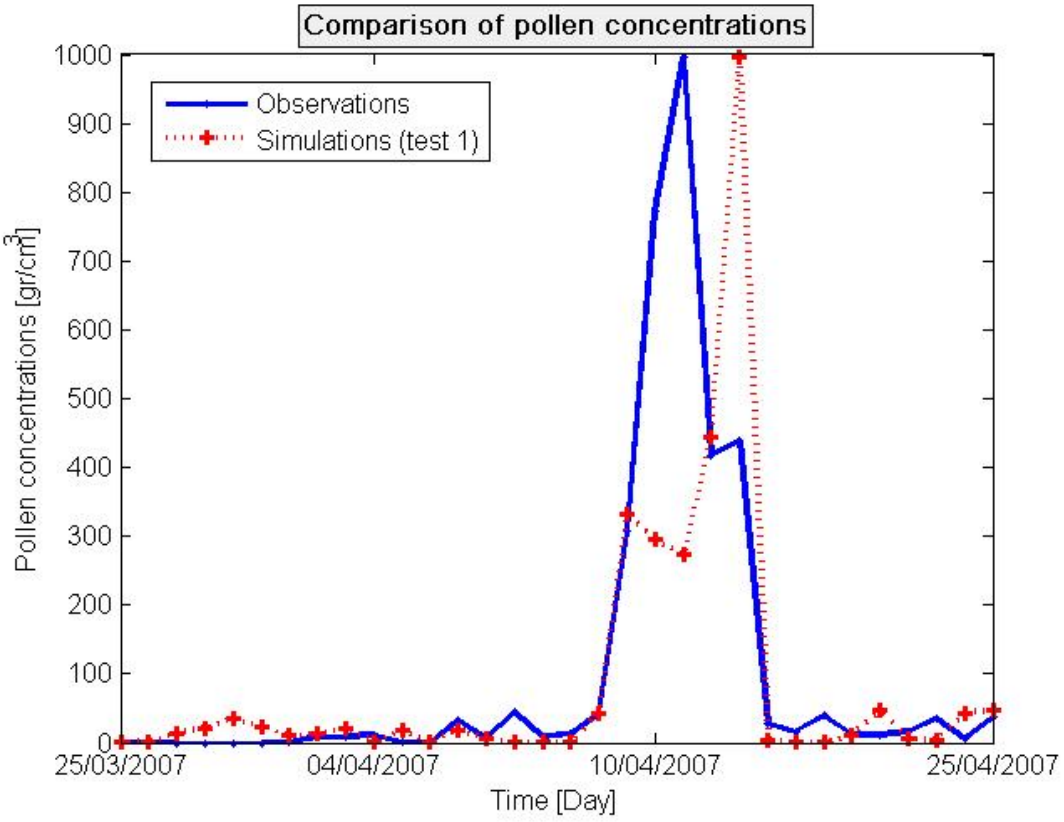

Figure 8.4: Study case n°1: One forest in De Bilt. Comparison between simulations and observations in Leiden.

trees in Leiden compared to reality. Moreover, the observation curve has a small second peak during the high release period which is absent on the other curve. Because the wind has a west direction this day, we can think that this pollen comes from Germany or other forest present in the east. We can realize that the concentrations are very small which can be explained by the poor knowledge of Birch forests (distribution and number of trees) and the density of trees. Note that this is just a scale factor.

The lone forest in Leiden

This time, we placed the forest in Leiden and observed the results in cell number 3[2] and we clearly can see an earlier peak of concentration. We can then think most of the pollen concentration measured in Leiden come from the vicinity.

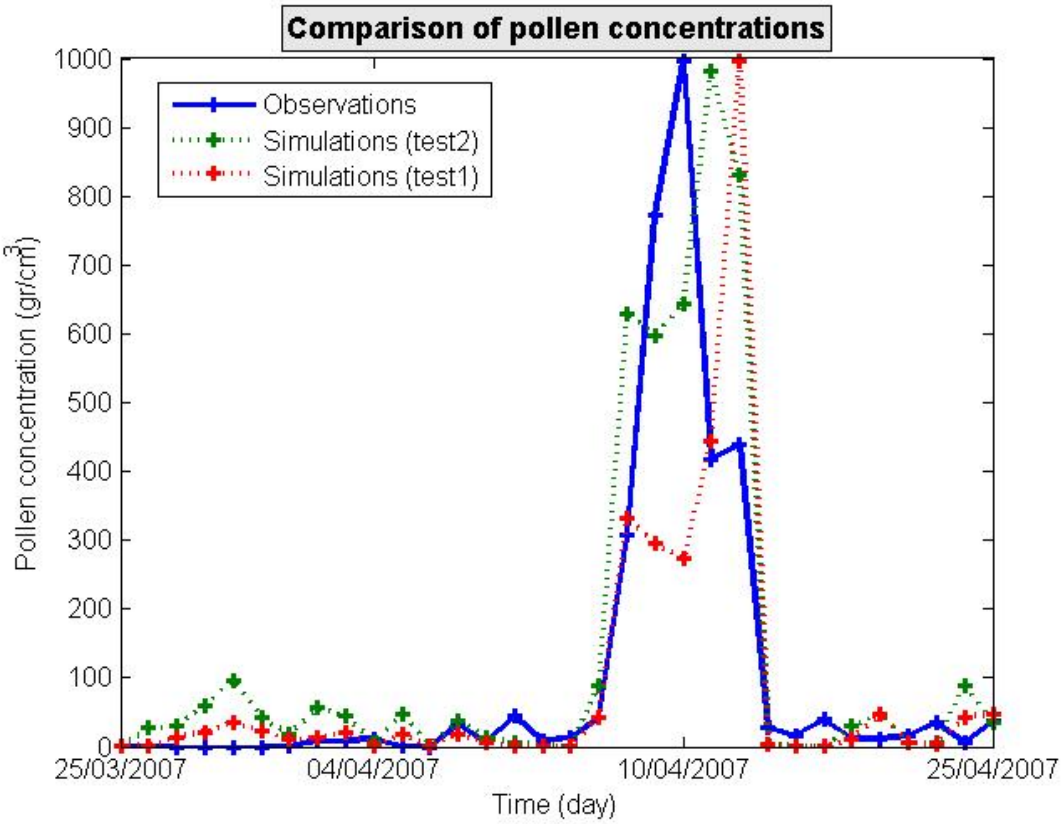

Figure 8.5: Study case n°2: One forest in Leiden. Comparison between simulations and observations in Leiden (and aside).

8.2.2 Second experimentation: Sofiev's Birch forest

This experiment shows the impact of other countries on The Netherlands. Indeed, contrary to the previous test, pollen is already present in the atmosphere the 25th of March due to

[2]Observations in Leiden were too chaotic

pollen coming from the South.

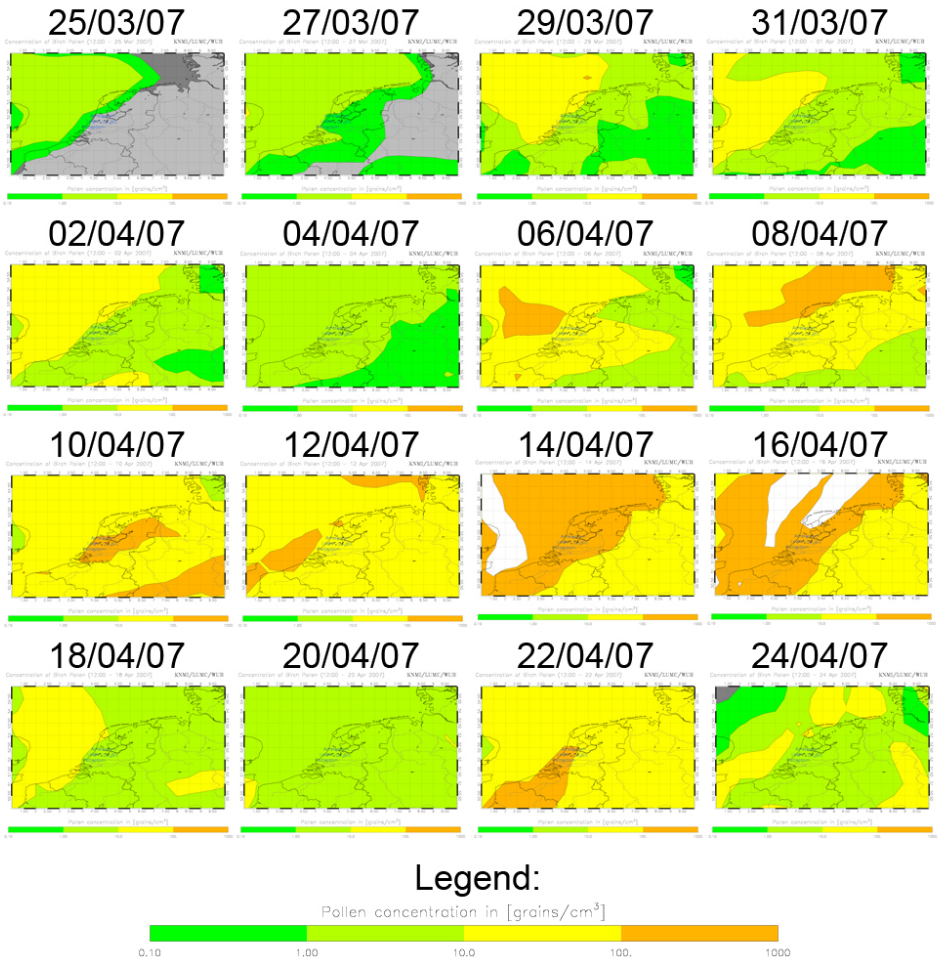

Figure 8.6: Study case n°11: European forest. Temporal evolution of pollen concentrations over the pollen season.

During the blooming season of 2007, we can see that some pollen is coming from eastern countries like Germany. This reinforces the assumption we made before about the second peak on observation curve. The white parts on the graphics correspond to a concentration over $1000 gr/cm^3$, these concentrations are of course too high compared to the reality. In this case again, this error can be explained by poor knowledge of Birch forests and density of trees.

Again, despite the factor used to reduce the simulations, we can see that the two graphics present a similar shape. This time we do not notice any early small peaks which strengthen the assumption made before. Indeed, on this vegetation map the density of Birch pollen is much more lo than on the map of the first study case. Nevertheless, no extra peak can be seen during the high pollination period but we also know that the vegetation presents a lack of information especially in The Netherlands.

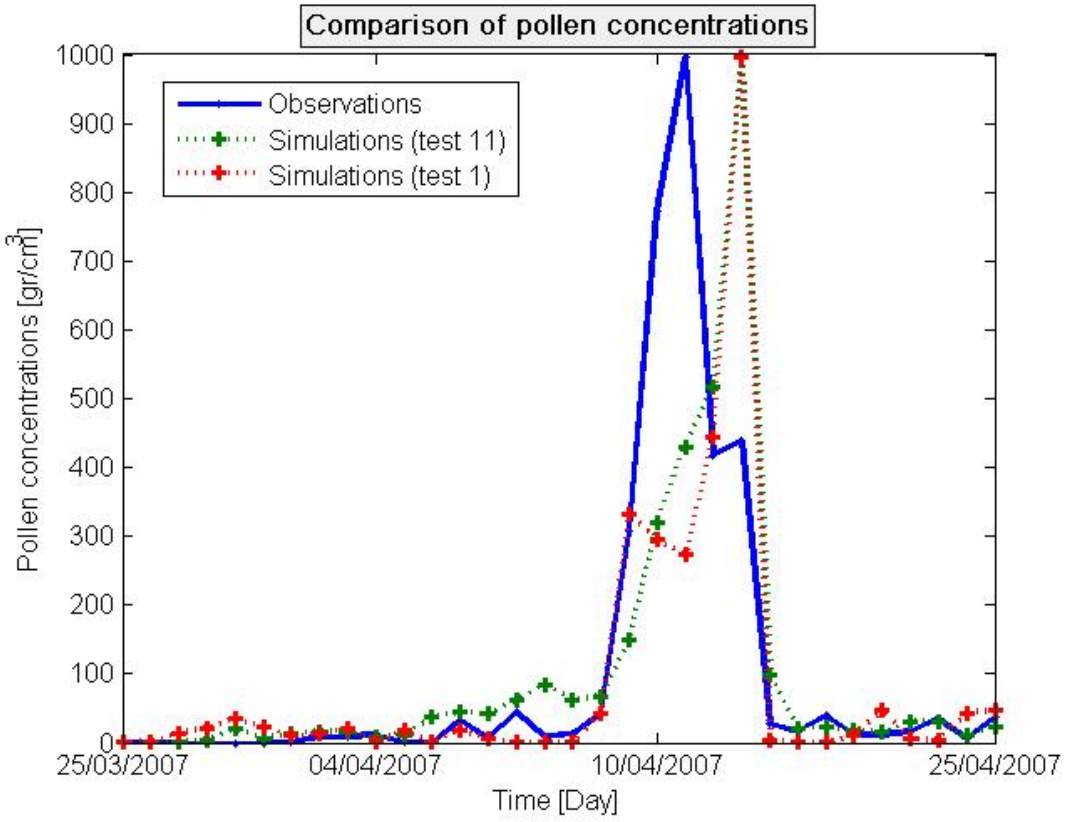

Figure 8.7: Study case n°11: European forest. Comparison between simulations and observations in Leiden.

Chapter 9

Conclusion

To conclude, the literature study showed an increase of interest in this field which promise a better understanding of the different processes involved in atmospheric pollen transport and pollen emission. We deplored a lack of information especially on biology of pollen and vegetation map which are very likely crucial to make operational forecasts. Nevertheless, the results were good and very promising. Indeed, they presented the same shape as observations and bring us new knowledge about the impact of meteorological phenomena on pollen transport. Moreover, the animations gave us an interesting feedback to understand the real distribution of pollen concentration in The Netherlands.

From my point of view, this project was very interesting and motivating not only because of its challenging and multidisciplinary aspects. First, it fitted perfectly with my education and research background. Thus, I used a lot of my skills and knowledge learned during my education at INSA and Universidad de Chile. On the one hand, my background in applied mathematics and computering let me understand quickly how the Chimere model works. On the other hand, the knowledge gained in Chile on atmospheric modelling was a strong support to understand the different processes of pollen grains within the atmosphere. Secondly, I gained a lot of knowledge especially in biology and phenology of plants and trees but I also managed to use IDL to display the results and I became familiar with both parallel and multiprocessor environments. Then, I collaborated with other researchers, not only with the members of the project but also with other institutes. Finally, I succeed in integrating all this disciplines into one project while adapting myself to a new culture.

As a future step, this project will need a good vegetation map and a good understanding of biological processes of pollen grains. Moreover, an hourly heat sum model should improve the emission process by supressing the need of using a reshape of emissions. Then, boundary conditions has to be implemented in the future. Results from Mikhail Sofiev could be used as boundary conditions. Furthermore, a fine grid with a resolution of 10x10 square kilometers should be employed to improve the quality of the results. Finally, the full serie of tests has to be run to finish the validation of the model.

Bibliography

[1] A. Damialis, D. Gioulekas, C. Lazopoulou, C. Balafoutis, D. Vokou **"Transport of airborne pollen into city of Thessaloniki: the effects of wind direction, speed and persistence"**, *Journal of the International Society of Biometeorology* (Thessaloniki, Greece: 19-10-2005): 139–145

[2] D. E. Aylor **"Settling speed of corn (Zea mays) pollen"**, *Journal of Aerosol Science*, **vol. 33**, (New Haven, USA: Pergamon, 02-07-2002)

[3] Malgorzata Latalowa, Miroslaw Mitus, Agnieszka Uruska **"Seasonal variations in the atmospheric Betula pollen count in Gdańsk (southern Baltic coast) in relation to meteorological parameters"**, *Aerobiologia*, **vol. 18**, (Gdańsk, Poland: Kluwer Academic Publishers 2002): 33–43

[4] J. Bartková-Scevková **"The influence of temperature, relative humidity and rainfall on the occurence of pollen allergens (Betula, Poaceae, Ambrosia artemisiifolia) in the atmosphere of Bratislava (Slovakia)"**, *Journal of the International Society of Biometeorology*, **vol. 48**, (Bratislava, Slovakia: 2003): 1–5

[5] J.-J. Delaunay, C. Seymour, V. Fouillet **"Investigation of short-range cedar pollen forecasting"**, *The American Physical Society*, **Physical Review E**, (Kanagawa, Japan: 2004): 70

[6] J.-J. Delaunay, R. Konishi, C. Seymour **"Analysis of cedar pollen time series: no evidence of low-dimensional chaotic behavior"**, *Journal of the International Society of Biometeorology*, **vol. 50**, (Tokyo, Japan: 2006): 154–158

[7] J.Emberlin, M. Smith, R. Close, B. Adams-Groom **"Changes in the pollen seasons of the early flowering trees Alnus spp. and Corylus spp. in Worcester, United Kingdom"**, *Journal of the International Society of Biometeorology*, **vol. 51**, (Worcester, United Kingdom: 2007): 181–191

[8] D. E. Aylor, M. T. Boehm **"Quantifying aerial concentrations of maize pollen in the atmospheric surface layer using remote-piloted airplanes and lagrangian stochastic model"**, *Journal of applied meteorology and climatology*, **vol. 45**, (New Heaven, Connecticut: 2006): 1003–1015

[9] M. A. Hart, R. de Dear, P. J. Beggs **"A synoptic climatology of pollen concentrations during the six warmest months in Sydney, Australia"**, *Journal of the International Society of Biometeorology*, **vol. 51**, (Sydney, Australia: 2007): 209–220

[10] M. Smith, J. Emberlin **"Constructing a 7-day ahead forecast model for grass pollen in north London, United Kingdom"**, *Journal of clinical and experimental allergy*, **vol. 35**, (London, United Kingdom: Blackwell Publishing Ltd, 2005): 1400–1406

[11] M. Smith, J. Emberlin **"A 30-day-ahead forecast model for grass pollen in north London, United Kingdom"**, *Journal of the International Society of Biometeorology*, **vol. 50**, (Worcester, United Kingdom: ISB, 04-01-2006): 233–242

[12] K. A. Hogda, S. R. Karlsen, I. Solheim, H. Tommervik, H. Ramfjord **"The start dates of birch pollen seasons in Fennoscandia studied by NOAA AVHRR NDVI data"**, *IEEE*, (Scandinavia: 2002)

[13] C. Porsbjerg, A. Rasmussen, V. Backer **"Airborne pollen in Nuuk, Greenland, and the importance of meteorological parameters"**, *Aerobiologia*, **vol. 19**, (Nuuk, Greenland: Kluwer Academic Publishers 2003): 29–37

[14] P Radičić, B. Šikoparija **"Betula spp. pollen in the atmosphere of novi Sad"**, *Aerobiologia*, **vol. 21**, (Serbia and Montenegro: Pergamon, 05-10-2004): 63–67

[15] B. Adams-Groom, J. Emberlin, J. Corden, W. Millington, J. Mullins **"Predicting the start of birch pollen season at London, Derby and Cardiff, United Kingdom, using a multiple regression model, based on data from 1987 to 1997"**, *Aerobiologia*, **vol. 18**, (Worcester, United Kingdom: 2002): 71–123

[16] N. Helbig, B. Vogel, H. Vogel, F. Fiedler **"Numerical modelling of pollen dispersion on the regional scale"**, *Aerobiologia* (Karlsruhe, Germany: Kluwer Academic Publishers, 09-12-2003)

[17] S. Schueler, K. H. Schlünzen **"Modelling of oak pollen dispersal on the landscape level with a mesoscale atmospheric model"**, *Environmental Modelling and Assessment*, **vol. 11**, (Nr. 3), (Grosshansdorf, Germany: Springer science, 18-02-2006): 179–194

[18] M. Sofiev, P. Siljamo, H. Ranta, A. Rantio-Lehtimäki **"Towards numerical forecating of long-range air transport of birch pollen: theorical considerations and feasibility study"**, *Journal of the International Society of Biometeorology* (Helsinki, Finland: SPI, 24-02-2006).

[19] F.T.M Spieksma, **"Daily hay fever forecast in the Netherlands. Radio broadcasting of the expected influence of the weather or subjective complaints of hay fever sufferers."**, *Allergy*, **vol. 35**, (Nr. 7), (Leiden, The Netherlands, 1980): 593–603

[20] OECD, **"Consensus document on the biology of european white birch (Betula pendula Roth)"**, ENV/JM/MONO(2003)12, 22-07-2003

[21] R. Pasken, J. Pietrowicz **"Using mesoscale meteorological models as a tool to forecast pollen concentrations"**, (Saint Louis, USA: 2000)

[22] R. Pasken, J. Pietrowicz **"Testing of mesoscale meteorological models as a tool to forecast pollen concentrations"**, (Saint Louis, USA: 2000)

[23] J.-J. Delaunay, K. Fedra, M. Kubat **"Cedar pollen forecasting in the Kanto region"**, *Archives of Complex Environmental Studies*, **vol. 14, N. 34**, (Kanagawa, Japan: 2002): 59–646

[24] M. Hjelmroos **"Long-distance transport of Betula pollen grains and allergic symptoms"**, *Aerobiologia*, **vol. 8**, (Stockholm, Sweden: 1992): 231–236

[25] G. S. Raynor, E. C. Ogden, J. V. Hayes **"Dispersion and Deposition of Ragweed Pollen from Experimental Sources"**, (Albany: Pergamon, 12-1970)

[26] J. Kukkonen, E. Auranen **"Proposal for new COST Action: Assessment of production, release, distribution and health impact of allergenic pollen in Europe (EUPOL)"**, *COST Action* (Helsinki, Finland: 14-09-2006)

[27] F. Di-Giovanni, P. M. Beckett **"On the Mathematical Modelling of Pollen Dispersal and Deposition"**, *Journal of Applied Meteorology* (Hull, England: 29-05-1990)

[28] P. Colella, P. R. Woodward **"The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations"**, *Journal of Computational Physics* **vol. 54**, (University of California, Berkley, California: 1984): 174–201

[29] J. G. Verwer **"Gauss-Seidel Iteration for Stiff ODEs from Chemical Kinetics"**, *Mathematics Subject Classification: Primary: 65L05. Secondary: 80A30, 80A32. CR categories: G1.8 G1.1* (P.O. Box 94079, 1090 GB Amsterdam, The Netherlands: 1991)

[30] L. Gallardo-Klenner **"Modelacíon atmosférica orientada a la dispersion de contaminantes"**, *Curso GF515-2006* (Departamento de Geofísica, Universidad de Chile, Chile: 2006)

[31] John H. Seinfeld, Spyros N. Pandis **"Atmospheric Chemistry and Physics - From Air Pollution to Climate Change"**, (Wiley-Interscience: 2006)

[32] Institut Pierre-Simon Laplace (C.N.R.S.), INERIS, LISA (C.N.R.S.) **"Documentation of chemistry-transport model, Chimere version V200606A"**, *http://euler.lmd.polytechnique.fr/chimere/* (France, 2005)

[33] **Wikipedia**, the free encyclopedia *http://en.wikipedia.org/*

[34] **KNMI**, the KNMI website *http://www.knmi.nl/*

Appendix A

LUMC model

The two main tables describing the model developed at LUMC are given here under.

Table 1
Scoring lists for expected weather

Score	Sun	Clouds	Rain	Score
1	dismal	heavily clouded	prolonged rain	1
2	no sun	overcast	rains	2
3	occasional sun	increasingly cloudy	rainy periods	3
4	short sunny spells	half clouded; changing cloudiness	occasional rains; showers	4
5	sunny spells	bright intervals	occasional showers	5
6	increasingly sunny	clearing sky	some local showers	6
7	pretty sunny	clear sky	locally a shower	7
8	sunny	almost cloudless	perhaps a single shower	8
9	sunny and warm	cloudless	mostly dry	9
10	sunny and very warm		dry	10

The expected type of weather, as forecast by the Royal Dutch Meteorological Institute (KNMI), is expressed by a single score representing the average of two scores, one for the amount of sunshine and/or degree of cloudiness (left) and the other for the intensity and/or duration of precipitation (right).

Table 2
Scheme for formulation of the relationship between weather-forecast score and index of complaints

No. pollen grains	No	Low; rising	High	Falling		
Period	April 27—May 7	May 8—May 31	June 1—June 25	June 26—July 20	Section of index scale	Weather conditions for hay fever pat.
Score	1,2,3,4,5,6 $\frac{t \leq 17°}{7,8,9}$	1,2,3,4,5 $\frac{t \leq 15°}{6}$	1,2 $\frac{t \leq 15°}{3,4}$	1,2,3,4 $\frac{t \leq 17°}{5,6}$	0-30	favourable
for	$\frac{t > 17°}{7,8,9}$	$\frac{t > 15°}{6}$ 7 $\frac{t \leq 20°}{8,9}$	$\frac{t > 15°}{3,4}$ $\frac{t \leq 20°}{5}$ $\frac{t \leq 15°}{6}$	$\frac{t > 17°}{5,6}$ 7	21-60	not very favourable
weather	10	$\frac{t > 20°}{8,9}$ 10	$\frac{t > 20°}{5}$ $\frac{t > 15°}{6}$ 7,8,9,10	8,9,10	> 50	unfavourable

The weather forecast score (derived from Table 1) located in the actual period and (where indicated) related to the expected maximum day time temperature, determines the appropriate section of the scale of indices of complaints (right). The three sections of the index scale correspond with the expected effects of weather conditions for the complaints of hay fever sufferers (far right) in the terms used in the message given in the daily radio broadcast.

Appendix B

Map of The Netherlands

Figure B.1:

Appendix C

Weather data in The Netherlands

(see next page)

City	Wind direction	Wind strength[m/s]	Precipitation duration [hour]	Precipitation amount [mm]
De Bilt	171	3.2	1,69	2.61
De Kooy	158	5.5	1,18	1.34
Eelde	179	3.9	1.16	1.33
Eindhoven	170	3.9	1.51	2.42
Leeuwarden	167	4.6	0.94	1.43
Maastricht	154	4.2	1.30	1.50
Rotterdam	164	4.2	1,58	2.29
Schiphol	165	4.9	1,36	1.75
Twenthe	171	3.4	1.57	2.58
Vlissingen	186	5.9	1.32	1.26
Average	168	4.4	1.36	1.86

Table C.1: Averages calculated during the Birch blooming season in 2007. Source KNMI.

Appendix D

Chimere code dissection

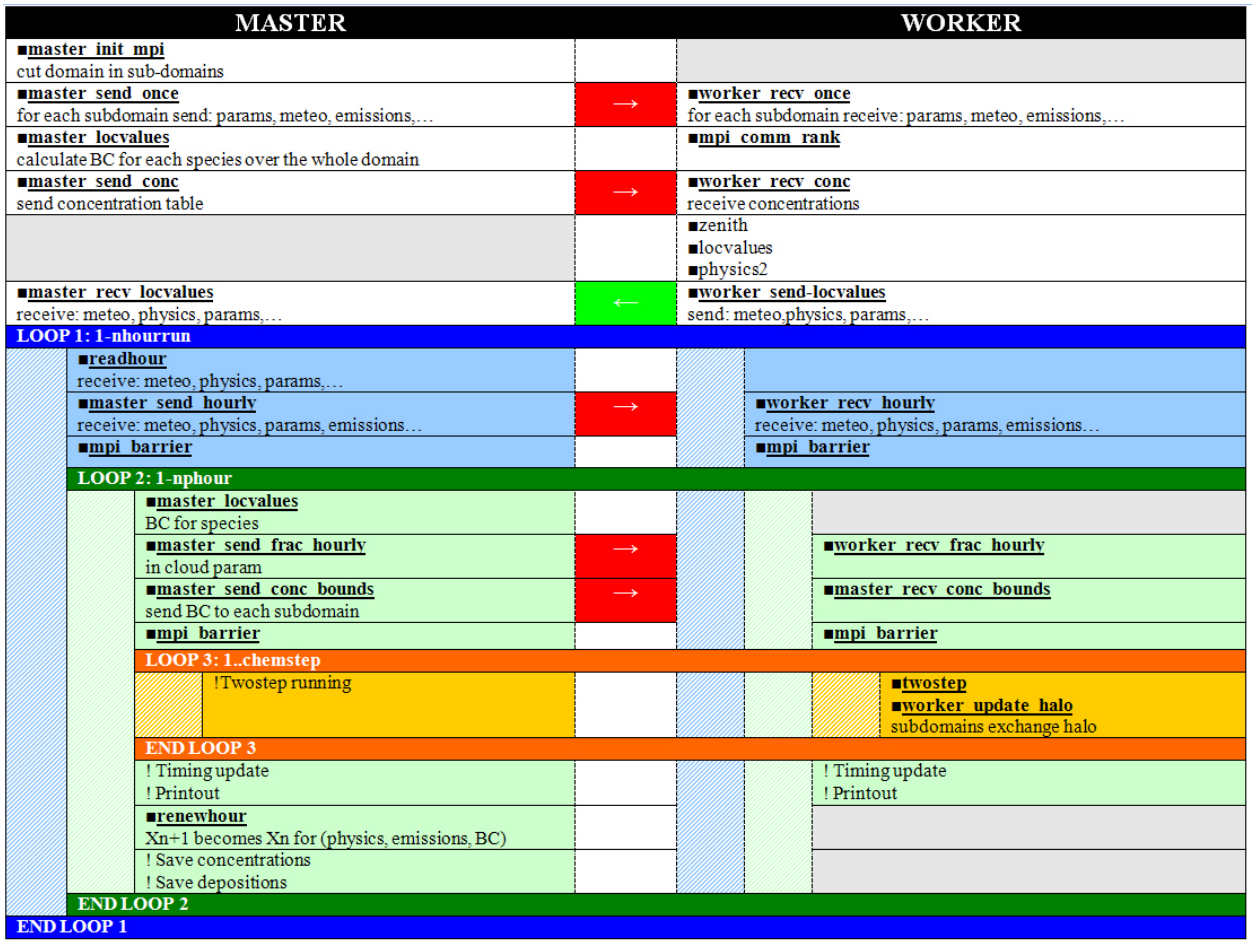

Figure D.1: Communication between the master process and the worker processes.

Appendix E

Pollen code - Created files

E.1 Emission file: diagpol.f90

```
!=========================================================================
! DIAGPOL
! version for Birch pollen based on heatsum model (compatible with WUR)
!
! by Jerome Hilaire
!
! INPUTS:
!  - pollen behaviour
!  - pollen properties
!  - tree properties
!  - heatsum model
!  - land-use map
!  - vegetation map(s)                          [trees/cell]
!  - Temperature (meteo file [temp2m/temp3D]) [degC]
! OUTPUT:
!  - heatsum           [degD]
!  - pollen emission   [molec/s]
! SUMMARY:
! 1. Initialization
! 2. Calculate Heatsum
! 3. Choose vegetation map
! 4. Initialize NETCDF file
! 5. Loop over hourly emissions and shape release
! 6. Finish program
!=========================================================================

program diagpol
  use chimere_consts
  use chimere_params
```

```
  use diagpol_common
  use diagpol_maths
  use diagpol_science
  use netcdf
#ifdef IFORT
  use ifport
  use ifposix
#endif

  implicit none

#define NCERR(lnum) if(ncstat/=NF90_NOERR) &
                                call nc_err(ncstat,lnum,'diagpol.f90')

  !***************************************************************
  ! Input parameters
  integer :: idss                       ! simulation start date
  integer :: ids                        ! start date, chimere format
  integer :: ide                        ! end date, chimere format
  integer :: nhourrun                   ! Model run duration
  integer :: ksens
  integer :: hs_type                    ! Type of heatsum
  character(len=32)  :: hs_model       ! Type of model
  character(len=132) :: tempdir,     & ! Directory of temperature files
                        fniEMISP,    & ! Input pollen emission species
                        fniPOLBE,    & ! Input pollen behaviours file
                        fniPOLPROP,  & ! Input pollen properties file
                        fniTREEPROP, & ! Input tree properties file
                        fniHSMOD,    & ! Input heatsum model file
                        fniLDU,      & ! Input LDU file
                        fniVEG,      & ! Input vegetation file
                        fniTEMP,     & ! Input temperature file
                        fnoHS,       & ! Output heatsum file
                        fnoPOL         ! Output emission file


  namelist /args/ idss,          &
                  ids,           &
                  ide,           &
                  nhourrun,      &
                  hs_type,       &
                  hs_model,      &
                  tempdir,       &
                  fniEMISP,      &
                  fniPOLBE,      &
```

```
                    fniPOLPROP, &
                    fniTREEPROP,&
                    fniHSMOD,   &
                    fniLDU,     &
                    fniVEG,     &
                    fniTEMP,    &
                    fnoHS,      &
                    fnoPOL

! LOCAL CONSTANTS
integer,parameter :: datestrlen = 19
character(len=*),parameter :: fniARGS = 'diagpol.nml'

! CONSTANT NETCDF attributes
character(len=*),parameter :: title='CHIMERE SUITE'
character(len=*),parameter :: subtitle='Pollen emissions file'
character(len=*),parameter :: subtitle_hs='Heat sum file'
character(len=*),parameter :: generator='Generated by diagpol'
character(len=*),parameter :: conventions=''

! LOCAL VARIABLES
! Logical
logical :: isin
! Indexes
integer :: ihsnbp              ! HS model index
integer :: inbp, jnbp          ! species indexes
integer :: ihoriz              ! nhoriz index    (nhoriz=nzonal*nmerid)
integer :: izo                 ! zonal index    (west-east)
integer :: ime                 ! merid index    (south-north)
integer :: ntype               ! LDU index
integer :: itim, itim2         ! time loop index
integer :: nz_captor,nm_captor ! CAPTOR position
! Dates
integer :: iye,imo,idy,iho
! Inputs
integer :: ifn_args
integer :: ifnEMISP, ifnPOLBE ,ifnPOLPROP , &
           ifnTREEPROP ,ifnHSMOD, ifnLDU, ifnMAP
integer :: ida,ida0                     ! dates, CHIMERE format
integer :: ierr
character(len=32)  :: usrname,hname,systime
character(len=255) :: cwd

! Initialization
integer :: nshifthour
```

```
integer :: sigma_gaussian

! For pollen behaviour
character(len=132) :: stringbuf ! Buffer for string

! For Heat sum
integer                                 :: days_inbetween, current_date
real,dimension(nzonal,nmerid,ntimes)    :: temp
real                                    :: temp_mean
character(len=1024)                     :: pathname

! Interpolated map
real,dimension(nzonal,nmerid) :: datamap
real(kind=8)                  :: coeff

! For land-use estimation
real(kind=8),allocatable,dimension(:,:)     :: xread
real(kind=8),dimension(9)                   :: tmpxread

! Emissions
real(kind=8),allocatable,dimension(:,:,:)   :: tmpbuf
real(kind=8),allocatable,dimension(:,:,:)   :: tmpbuf2
character(len=dlen)                         :: datebuf
real(kind=8)                                :: releasing_shape


character(len=1024) :: history

! Functions
character(len=datestrlen) :: numeric2mm5date
integer                   :: mm5date2numeric
integer                   :: interdat
!*****************************************************************

!-----------------------------------------------------------------
! 1. Initializations
!
allocate(xread(nhoriz,9))
allocate(tmpbuf(nemisp,nzonal,nmerid))
allocate(tmpbuf2(nemisp,nzonal,nmerid))
!
!
! 1.1 Set parameter
nshifthour     = 3            ! For parabola shape
sigma_gaussian = 1            ! For gaussian shape
```

```
nz_captor      = 32 != mod(nsho,nzonal)
nm_captor      = 36 !nsho/nzonal+1
!
!
! 1.2 Read input arguments from namelist
call opfi(ifn_args,fniARGS,'f','o')
read(ifn_args,nml=args,iostat=ierr)
if(ierr/=0) then
   print *,'diagpol: problem reading arguments namelist'
   stop
end if
close (ifn_args)
call get_system(usrname,hname,systime,cwd)
!
!
! 1.3  Reading Land Use map
call opfi(ifnLDU,fniLDU,'f','o')
do ihoriz=1,nhoriz
  read(ifnLDU,*)(xread(ihoriz,ntype),ntype=1,9)
enddo
close(ifnLDU)
!
!
! 1.4 Reading name of pollen species
call opfi(ifnEMISP,fniEMISP,'f','o')
do inbp=1,nemisp
  read(ifnEMISP,*) emisp_name(inbp)
enddo
close(ifnEMISP)
!
!
! 1.5 Reading and setting parameters
Write(*,*) ' - Loading pollen properties...'
!
! 1.5.1 Pollen properties
call opfi(ifnPOLPROP,fniPOLPROP,'f','o')
do inbp=1,12
  read(ifnPOLPROP,*)
end do
do inbp=1,nemisp
  read(ifnPOLPROP,*)  pollenprop_name(inbp),pollenprop_diamvar(inbp),&
                  pollenprop_diammean(inbp),pollenprop_density(inbp),&
                  pollenprop_molar(inbp)
  do jnbp=1,nemisp
    if (emisp_name(jnbp) .eq. pollenprop_name(inbp)) then
```

```
      isin = .TRUE.
      exit
   end if
  end do
  if (.not.isin) print *,'diagpol: Problem with pollen property file.&
                   Unknown pollen specie ',trim(pollenprop_name(inbp))
enddo
close(ifnPOLPROP)
!
! 1.5.2 Tree properties
Write(*,*) ' - Loading tree properties...'
call opfi(ifnTREEPROP,fniTREEPROP,'f','o')
do inbp=1,12
  read(ifnTREEPROP,*)
end do
do inbp=1,nemisp
  read(ifnTREEPROP,*) treeprop_name(inbp),treeprop_density(inbp),&
                      treeprop_nbgrains(inbp)
  do jnbp=1,nemisp
    if (emisp_name(jnbp) .eq. treeprop_name(inbp)) then
      isin = .TRUE.
      exit
   end if
  end do
  if (.not.isin) print *,'diagpol: Problem with tree property file. &
                   Unknown pollen specie ',trim(treeprop_name(inbp))
enddo
close(ifnTREEPROP)
!
! 1.5.3 Calculating masses, number of molecules, etc...
do inbp=1,nemisp
  pollen_grain_mass(inbp)    = 4d0/3d0*pi                          &
                             *(pollenprop_diammean(inbp)/2d0)**3 &
                             *pollenprop_density(inbp)   ! mass of a grain
  pollen_grain_molec(inbp)   = pollen_grain_mass(inbp)           &
                             /pollenprop_molar(inbp)*an  ! nb of molecules
  pollen_grain_factor(inbp)  = treeprop_density(inbp)            &
                             *treeprop_nbgrains(inbp)            &
                             *pollen_grain_molec(inbp)   ! nb of molecules
enddo
!
!
! 1.6 Read pollen behaviours
Write(*,*) ' - Loading pollen behaviours...'
call opfi(ifnPOLBE,fniPOLBE,'f','o')
```

```
 do inbp=1,nemisp
  read(ifnPOLBE,*) pollenpobe_name(inbp),stringbuf ! Get the name
  do jnbp=1,nemisp
   if (emisp_name(jnbp) .eq. pollenpobe_name(inbp)) then
    isin = .TRUE.
    exit
   end if
  end do
   if (isin) then
    write(*,*) '  - specie: ',trim(pollenpobe_name(inbp)),' USED'
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%total_amount
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%total_amount_unty
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%low_level_amt
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%high_level_amt
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%coterm30y
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%hs_sdate
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%hs_startend_diff
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%hs_thresh
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%temp_cutoff
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%precip_thresh
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%wind_max_scale
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%wind_sat
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%humid_low_thresh
    read(ifnPOLBE,*) stringbuf,pobe(inbp)%humid_high_thresh
    read(ifnPOLBE,*) ! Comment
   else
    print *,'diagpol: Unknown pollen specie ',trim(pollenpobe_name(inbp))
    stop
   end if
 end do
close(ifnPOLBE)
!
!
! 1.7 Load Heatsum models
Write(*,*) ' - Loading heatsum models...'
! Read
 call opfi(ifnHSMOD,fniHSMOD,'f','o')
  read(ifnHSMOD,*)
  read(ifnHSMOD,*)
  do inbp=1,nemisp
    read(ifnHSMOD,*) pollenhsm_name(inbp)
    do jnbp=1,nemisp
      if (emisp_name(jnbp) .eq. pollenhsm_name(inbp)) then
        isin = .TRUE.
        exit
```

```
      end if
    end do
    if (isin) then
      do ihsnbp=1,hsnbp
        read(ifnHSMOD,*) hsm_y(ihsnbp,inbp), hsm_x(ihsnbp,inbp)
      end do
    else
      print *,'diagpol: Unknown pollen specie ',pollenhsm_name(inbp)
      stop
    endif
  end do
 close(ifnHSMOD)
call heatsum_linear


!-----------------------------------------------------------------
! 2. Calculate Heat sum if HS.nc does not exist yet
!
! 2.1 Minimum starting date of pollen season
min_start_ps_date =pobe(1)%hs_sdate
if (nemisp .ge. 2) then
  do inbp=2,nemisp
    if (pobe(inbp)%hs_sdate .lt. min_start_ps_date) then
      min_start_ps_date =pobe(inbp)%hs_sdate
    endif
  enddo
endif
!
!
! 2.2 Checking temperature file
write(pathname,'(a,"/meteo.",i8.8,".nc")') trim(tempdir),min_start_ps_date
open(ifn_args,file=pathname,iostat=ierr)
  if(ierr/=0) stop ' *** diagpol : Problem with temperature files. &
                   Check dates or check file is existing.'
close(ifn_args)
!
!
! 2.3 Create Heat sum file
days_inbetween = interdat(min_start_ps_date,ide*100)/24 +1
if (ids/100.eq.idss) then
 write(*,*) "First time, Create HS file"
 call create_heatsum_file
 hs_prev = 0e0
else
 write(*,*) "Open HS file"
```

```
   call open_heatsum_file
   call get_previous_heatsum
  endif
  !
  !
  ! 2.4 Calculate Heat sum
  if (hs_type.eq.1) then
   if (ids/100.eq.idss) then
    do itim=1,days_inbetween
     call reldat(min_start_ps_date,(itim-1)*24,current_date)
     write(pathname,'(a,"/meteo.",i8.8,".nc")') trim(tempdir), &
                                          current_date/100
     ncstat=nf90_open(pathname,NF90_NOWRITE,temp_ncid)
     NCERR(__LINE__)
     call get_temperature
     ncstat=nf90_close(temp_ncid)
     NCERR(__LINE__)
     do inbp=1,nemisp
       call calculate_heatsum(temp-273.15,hs_prev,hs_calc)
     enddo
     ! DISPLAY FOR CAPTOR-----------------------------------------
     temp_mean = 0e0
     do itim2=1,ntimes
      temp_mean = temp_mean + temp(nz_captor,nm_captor,itim2)
     enddo
     temp_mean = temp_mean/24 - 273.15
     if (hs_calc(1,nz_captor,nm_captor).lt.pobe(1)%hs_thresh) then
      write(*,'(i8.8,a,f5.2,a,f7.2,a)') current_date/100,      &
            ' at CAPTOR T: ',temp_mean,'Deg C - HS: ',         &
            hs_calc(1,nz_captor,nm_captor),'Deg D'
     else
      write(*,'(i8.8,a,f5.2,a,f7.2,a)') current_date/100,      &
            ' at CAPTOR T: ',temp_mean,'Deg C - HS: ',         &
            hs_calc(1,nz_captor,nm_captor),'Deg D (Emission!)'
     endif
     ! END OF DISPLAY FOR CAPTOR ----------------------------------
     call write_heatsum_file
     hs_prev = hs_calc
    enddo
   else
!    days_inbetween = interdat(ids*100,ide*100)/24
!    do itim=1,days_inbetween
!     call reldat(ids*100,(itim-1)*24,current_date)
     current_date = ide*100
     write(pathname,'(a,"/meteo.",i8.8,".nc")') trim(tempdir), &
```

```
                                                 current_date/100
      ncstat=nf90_open(pathname,NF90_NOWRITE,temp_ncid)
      NCERR(__LINE__)
      call get_temperature
      ncstat=nf90_close(temp_ncid)
      NCERR(__LINE__)
      do inbp=1,nemisp
        call calculate_heatsum(temp-273.15,hs_prev,hs_calc)
      enddo
      ! DISPLAY FOR CAPTOR ----------------------------------------
      temp_mean = 0e0
      do itim2=1,ntimes
       temp_mean = temp_mean + temp(nz_captor,nm_captor,itim2)
      enddo
      temp_mean = temp_mean/24 - 273.15
      if (hs_calc(1,nz_captor,nm_captor).lt.pobe(1)%hs_thresh) then
       write(*,'(i8.8,a,f5.2,a,f7.2,a)') current_date/100,       &
              ' at CAPTOR T: ',temp_mean,'Deg C - HS: ',         &
              hs_calc(1,nz_captor,nm_captor),'Deg D'
      else
       write(*,'(i8.8,a,f5.2,a,f7.2,a)') current_date/100,
              ' at CAPTOR T: ',temp_mean,'Deg C - HS: ',         &
              hs_calc(1,nz_captor,nm_captor),'Deg D (Emission!)' &
      endif
      ! END OF DISPLAY FOR CAPTOR ---------------------------------
      call write_heatsum_file
      hs_prev = hs_calc
!      enddo
    endif
   elseif (hs_type.eq.2) then
    ! Not implemented yet
   else
    write(*,*) "diagpol: Unknown type of heat sum."
    stop
   endif
   !-------------------------------------------------------------------
   ! 3. Chose vegetation map:
   !
   ! 3.1 Reading birch map
   ! Write(*,*) 'Loading vegetation maps...'
   ! do inbp=1,nemisp
   !  write(pathname,'(a,"/map.",a,".dat")') trim(fniVEG), &
                                  trim(pollenprop_name(inbp))
   !  call opfi(ifnMAP,pathname,'u','o')
   !   read(ifnMAP) datamap
```

```
  !  close(ifnMAP)
  !  tmpbuf(1,:,:) = dble(datamap)                                    )
  ! enddo
  !
  !
  ! 3.2 Generation of fake Birch tree map using LDU
!   Write(*,*) 'Generating fake vegetation maps...'
!   tmpbuf = 1
!   do inbp=1,nemisp
!     do ime=1,nmerid
!       do izo=1,nzonal
!         ihoriz=izo+(ime-1)*nzonal
!         tmpxread = xread(ihoriz,:)
!         if ((tmpxread(3) .gt. 0d0) .OR. (tmpxread(4) .gt. 0d0) &
              .OR. (tmpxread(9) .gt. 0d0)) then
!           coeff = 0d0
!         else
!           coeff = 0d0
!           Pourcentage of trees or plants in the area
            ! Agricultural land/crops
!           if (tmpxread(1) .gt. 0d0) coeff = coeff + 0.05d0*tmpxread(1)
            ! Grass land
!           if (tmpxread(2) .gt. 0d0) coeff = coeff + 0.04d0*tmpxread(2)
            ! Urban
!           if (tmpxread(5) .gt. 0d0) coeff = coeff + 0.10d0*tmpxread(5)
            ! Shrubs
!           if (tmpxread(6) .gt. 0d0) coeff = coeff + 0.02d0*tmpxread(6)
            ! Needleaf forest
!           if (tmpxread(7) .gt. 0d0) coeff = coeff + 0.05d0*tmpxread(7)
            ! Broadleaf forest
!           if (tmpxread(8) .gt. 0d0) coeff = coeff + 0.20d0*tmpxread(8)
!         end if
!         tmpbuf(inbp,izo,ime) = tmpbuf(inbp,izo,ime) * coeff
!       end do
!     end do
!   end do
  !
  !
  ! 3.3 Play with cells in The Netherlands
  ! Leiden coordinates (32,36)
  tmpbuf = 0
  tmpbuf(1,32,36) = 1     ! "One tree" in The Netherlands 32,36
  !tmpbuf(1,31,36) = 1 ! "One cell of trees" in The Netherlands
```

```
  !---------------------------------------------------------------
  ! 4. Initialize NETCDF emission file
  !
  call create_emission_file


  !---------------------------------------------------------------
  ! 5. Loop of hourly emissions
  !
  Write(*,*) 'Calculating emissions...'
  call get_heatsum
  do itim=1,nhourrun
    !
    ! 5.1 Initialization-----------------------
    tmpbuf2 = 0.
    do inbp=1,nemisp
      do ime=1,nmerid
        do izo=1,nzonal

          polperc = calculate_emission(hs_loop(inbp,izo,ime,1) &
                                      ,hs_loop(inbp,izo,ime,2) &
                                      ,inbp,izo,ime)

          qty_pollen_in_cell = polperc * tmpbuf(inbp,izo,ime) *&
                               pollen_grain_factor(inbp)
      !
      !
      ! 5.2 Set releasing shape-------------------
      ! Each tree release an certain amount of grains
      releasing_shape =  1                          &
                     * - qty_pollen_in_cell/288d0  &
                     *  dble((itim-7)*(itim-19))
!                    *
!                    * rand()
      if (releasing_shape .lt. 0.) releasing_shape = 0d0
      !
      !
      ! 5.3 Calculate emissions-------------------
        if (hs_type.eq.1) then
          tmpbuf2(inbp,izo,ime) = releasing_shape / dble(3600*24)
        elseif (hs_type.eq.2) then
          ! Not implemented yet
        else
          write(*,*) "diagpol: Unknown type of heat sum."
          stop
```

```
        endif
        end do
      end do
    enddo
    !
    !
    ! 5.4 Writing to NETCDF file----------------
    ! Write Times variable
    ncstat=nf90_put_var(                              &
         emis_ncid,                                   &
         timesvarid,                                  &
         numeric2mm5date(ida),                        &
         start=(/1,          itim/),                  &
         count=(/datestrlen,1   /)                    &
         )
    NCERR(__LINE__)
    ! Write pollenemission variable
    ncstat=nf90_put_var(                              &
         emis_ncid,                                   &
         emisvarid,                                   &
         tmpbuf2,                                     &
         start=(/1,1,     1,       itim/),          &
         count=(/nemisp,nzonal,nmerid,1   /)          &
         )
    NCERR(__LINE__)
!*******************************************************************
     ! DISPLAY FOR CAPTOR ----------------------------------------
     write(*,'(i10.10,a,E8.2,a)') ida,' at CAPTOR - Emission: ', &
                          tmpbuf2(1,nz_captor,nm_captor),' molec/s'
     ! END OF DISPLAY FOR CAPTOR ---------------------------------
    call reldat(ida,1,ida0)
    ida = ida0
  enddo !itim=1,nhourrun


  !-----------------------------------------------------------------
  ! 6. Finishing program
  ncstat=nf90_close(emis_ncid)
  NCERR(__LINE__)
  ncstat=nf90_close(hs_ncid)
  NCERR(__LINE__)
  deallocate(xread)
  deallocate(tmpbuf)
  deallocate(tmpbuf2)
```

```
contains

  subroutine create_heatsum_file
    ! open output file
    ncstat=nf90_create(fnoHS,NF90_CLOBBER,hs_ncid)
    NCERR(__LINE__)
   ! create dimensions in output file
    ncstat=nf90_def_dim(hs_ncid,'Time', NF90_UNLIMITED,hs_timedimid)
    NCERR(__LINE__)
    ncstat=nf90_def_dim(hs_ncid,'DateStrLen', datestrlen,hs_datedimid)
    NCERR(__LINE__)
    ncstat=nf90_def_dim(hs_ncid,'west_east',nzonal,hs_wedimid)
    NCERR(__LINE__)
    ncstat=nf90_def_dim(hs_ncid,'south_north',nmerid,hs_sndimid)
    NCERR(__LINE__)
    ncstat=nf90_def_dim(hs_ncid,'pollen_species',nemisp,hs_speciesdimid)
    NCERR(__LINE__)
    ! create variables in output file
    ncstat=nf90_def_var(                                          &
         hs_ncid,                                                 &
         'Times',                                                 &
         NF90_CHAR,                                               &
         (/hs_datedimid,hs_timedimid/),                           &
          hs_timesvarid)
    NCERR(__LINE__)
    ncstat=nf90_def_var(                                          &
         hs_ncid,                                                 &
         'heatsum',                                               &
          NF90_FLOAT,                                             &
         (/hs_speciesdimid,hs_wedimid,hs_sndimid,hs_timedimid/), &
         hs_heatsum_varid)
    NCERR(__LINE__)

    ! create global attributes
    ncstat=nf90_put_att(hs_ncid,NF90_GLOBAL,'Title',title)
    NCERR(__LINE__)
    ncstat=nf90_put_att(hs_ncid,NF90_GLOBAL,'Sub-title',subtitle_hs)
    NCERR(__LINE__)
    ncstat=nf90_put_att(hs_ncid,NF90_GLOBAL,'Generating_process',generator)
    NCERR(__LINE__)
    ncstat=nf90_put_att(hs_ncid,NF90_GLOBAL,'Conventions',conventions)
    NCERR(__LINE__)
    ncstat=nf90_put_att(hs_ncid,NF90_GLOBAL,'Domain',domain)
    NCERR(__LINE__)
   ! Fill history
```

```
  call get_system(usrname,hname,systime,cwd)
  history=''
  call update_history(history,cwd,fnoHS,systime,usrname,hname,fniTEMP)
  ncstat=nf90_put_att(hs_ncid,NF90_GLOBAL,'history', &
      history(1:len_trim(history)))
  NCERR(__LINE__)
 ! End netCDF definition mode
  ncstat=nf90_enddef(hs_ncid)
  NCERR(__LINE__)
end subroutine create_heatsum_file

subroutine get_previous_heatsum
  implicit none
  if (hs_type.eq.1) then
    ncstat=nf90_get_var(    &
       hs_ncid,             &
       hs_heatsum_varid,    &
       hs_prev,             &
       (/1,1,1,days_inbetween-1/),(/nemisp,nzonal,nmerid,1/))
  elseif (hs_type.eq.2) then
    ! Not implemented yet
  else
    write(*,*) "diagpol: Unknown type of heat sum."
    stop
  endif
end subroutine get_previous_heatsum

subroutine get_temperature
 implicit none
 integer,allocatable,dimension(:) :: stvec
 integer,allocatable,dimension(:) :: cntvec
 allocate(stvec(3))
 allocate(cntvec(3))
 stvec=(/1,1,1/)
 cntvec=(/nzonal,nmerid,ntimes/)
 ! Get temperature from temperature file (file already opened)
  ncstat=nf90_inq_varid(temp_ncid,'tem2',temp_temp_varid)
  NCERR(__LINE__)
  ncstat=nf90_get_var(temp_ncid,temp_temp_varid,temp,stvec,cntvec)
  NCERR(__LINE__)
 deallocate(stvec)
 deallocate(cntvec)
end subroutine get_temperature

subroutine write_heatsum_file
```

```
 implicit none
 character(len=datestrlen) :: numeric2mm5date
 ! write Times variable
  ncstat=nf90_put_var(                                &
       hs_ncid,                                       &
       hs_timesvarid,                                 &
       numeric2mm5date(current_date),                 &
       start=(/1,         itim/),                     &
       count=(/datestrlen,1   /)                      &
       )
  NCERR(__LINE__)
 ! write heatsum variable
  ncstat=nf90_put_var(                                &
       hs_ncid,                                       &
       hs_heatsum_varid,                              &
       hs_calc,                                       &
       start=(/1,1,     1,       itim/),              &
       count=(/nemisp,nzonal,nmerid,1   /)            &
       )
  NCERR(__LINE__)
end subroutine write_heatsum_file

subroutine create_emission_file
  ida = ids
  ! open output file
  ncstat=nf90_create(fnoPOL,NF90_CLOBBER,emis_ncid) ; NCERR(__LINE__)
  ! create dimensions in output file
  ncstat=nf90_def_dim(emis_ncid,'Time',NF90_UNLIMITED,timedimid)
  NCERR(__LINE__)
  ncstat=nf90_def_dim(emis_ncid,'DateStrLen',datestrlen,datedimid)
  NCERR(__LINE__)
  ncstat=nf90_def_dim(emis_ncid,'west_east',nzonal,wedimid)
  NCERR(__LINE__)
  ncstat=nf90_def_dim(emis_ncid,'south_north',nmerid,sndimid)
  NCERR(__LINE__)
  ncstat=nf90_def_dim(emis_ncid,'pollen_species',nemisp,speciesdimid)
  NCERR(__LINE__)

  ! create variables in output file
  ncstat=nf90_def_var(                                  &
     emis_ncid,                                         &
     'Times',                                           &
     NF90_CHAR,                                         &
     (/datedimid,timedimid/),                           &
     timesvarid)
```

```
  NCERR(__LINE__)
  ncstat=nf90_def_var(                                &
     emis_ncid,                                       &
     'pemission',                                     &
     NF90_FLOAT,                                      &
     (/speciesdimid,wedimid,sndimid,timedimid/),      &
     emisvarid)
  NCERR(__LINE__)

  ! create global attributes
  ncstat=nf90_put_att(emis_ncid,NF90_GLOBAL,'Title',title)
  NCERR(__LINE__)
  ncstat=nf90_put_att(emis_ncid,NF90_GLOBAL,'Sub-title',subtitle)
  NCERR(__LINE__)
  ncstat=nf90_put_att(emis_ncid,NF90_GLOBAL,'Generating_process',&
                                                        generator)
  NCERR(__LINE__)
  ncstat=nf90_put_att(emis_ncid,NF90_GLOBAL,'Conventions',conventions)
  NCERR(__LINE__)
  ncstat=nf90_put_att(emis_ncid,NF90_GLOBAL,'Domain',domain)
  NCERR(__LINE__)

  history=''
  call update_history(history,cwd,fnoPOL,systime,usrname,hname,fniLDU)

  ncstat=nf90_put_att(emis_ncid,NF90_GLOBAL,'history', &
     history(1:len_trim(history)))
  NCERR(__LINE__)

  !! End netCDF definition mode
  ncstat=nf90_enddef(emis_ncid)
  NCERR(__LINE__)
end subroutine create_emission_file

subroutine open_heatsum_file
  implicit none
  ! open output file
  ncstat=nf90_open(fnoHS,NF90_WRITE,hs_ncid) ; NCERR(__LINE__)
  ! Checking dimensions consistency between file and chimere parameters
  ! dimensions IDs
  ncstat=nf90_inq_dimid(hs_ncid,'Time',hs_timedimid)
  NCERR(__LINE__)
  ncstat=nf90_inq_dimid(hs_ncid,'DateStrLen',hs_datedimid)
  NCERR(__LINE__)
  ncstat=nf90_inq_dimid(hs_ncid,'west_east',hs_wedimid)
```

```
    NCERR(__LINE__)
    ncstat=nf90_inq_dimid(hs_ncid,'south_north',hs_sndimid)
    NCERR(__LINE__)
    ncstat=nf90_inq_dimid(hs_ncid,'pollen_species',hs_speciesdimid)
    NCERR(__LINE__)
    ! Reading variable IDs
    ncstat=nf90_inq_varid(hs_ncid,'Times',hs_timesvarid)
    NCERR(__LINE__)
    ncstat=nf90_inq_varid(hs_ncid,'heatsum',hs_heatsum_varid)
    NCERR(__LINE__)
  end subroutine open_heatsum_file

  subroutine get_heatsum
    implicit none
    if (hs_type.eq.1) then
      ncstat=nf90_get_var(    &
         hs_ncid,             &
         hs_heatsum_varid,    &
         hs_loop,             &
         (/1,1,1,days_inbetween-1/),(/nemisp,nzonal,nmerid,2/))
    elseif (hs_type.eq.2) then
      ! Not implemented yet
    else
      write(*,*) "diagpol: Unknown type of heat sum."
      stop
    endif
  end subroutine get_heatsum

end program diagpol
```

E.2 Emission file: diagpol_maths.f90

```
module diagpol_maths

 use diagpol_common
 use chimere_params

 implicit none

contains
!*********************************************************************
! Create a picewise linear approximation for an heatsum model
!*********************************************************************
! Inputs  : hsm_x         - Vector of temperatures
```

```
!           hsm_y         - Vector of percentages
! Outputs : hs_linear     - Matrix of linear coefficients (a,b)
!                           in y=a.x+b
!*****************************************************************
subroutine heatsum_linear

 implicit none

 ! Inputs

 ! Outputs

 ! Local variables
 ! Loop index
 integer :: i, j, nbp_loop

!*****************************************************************
 ! Create the picewise linear interpolation
 nbp_loop = hsnbp-1
 do i=1,nemisp
  do j=1,nbp_loop
   hsm_linear(j,i,1) = (hsm_y(j+1,i)- hsm_y(j,i)) &
                       / (hsm_x(j+1,i)- hsm_x(j,i))
   hsm_linear(j,i,2) = hsm_y(j,i) - hsm_linear(j,i,1)*hsm_x(j,i)
  end do
 end do

end subroutine heatsum_linear


!*****************************************************************
! Calculate the percentage of emission during a time step
!*****************************************************************
! Inputs : x1, x2        - temperatures corresponding to t1 and t2 in dt
!                          with x1 < x2
! Output : integrate     - percentage of emission released
!*****************************************************************
real function calculate_emission (x1,x2,inbp,izo,ime)

 implicit none

 real :: a1, a2, b1, b2, x1, x2, y1, y2
 integer :: ix1,ix2,izo,ime
 integer :: inbp
 integer :: i, cnbp
```

```
 ! Get indexes
  ix1 = 0
  ix2 = 0
  cnbp = hsnbp-1
  do i=1,cnbp
   if (ix1 .ne. 0 .AND. ix2 .ne. 0) exit
   if (x1 .gt. hsm_x(i,inbp)   .AND. &
       x1 .lt. hsm_x(i+1,inbp) .OR.  &
       x1 .eq. hsm_x(i,inbp))  then
    ix1 = i
   end if
   if (x2 .gt. hsm_x(i,inbp)   .AND. &
       x2 .lt. hsm_x(i+1,inbp) .OR.  &
       x2 .eq. hsm_x(i,inbp))  then
    ix2 = i
   end if
  end do

  ! Calculate
  if (ix1 .eq. ix2) then
    a1 = hsm_linear(ix1,inbp,1)
    b1 = hsm_linear(ix1,inbp,2)

    y1 = a1*x1+b1
    y2 = a1*x2+b1

    calculate_emission = y2-y1
  else
    a1 = hsm_linear(ix1,inbp,1)
    b1 = hsm_linear(ix1,inbp,2)
    a2 = hsm_linear(ix2,inbp,1)
    b2 = hsm_linear(ix2,inbp,2)

    y1 = a1*x1+b1
    y2 = a2*x2+b2

    calculate_emission = y2-y1
  end if
end function calculate_emission

end module diagpol_maths
```

E.3 Emission file: diagpol_science.f90

```
module diagpol_science

  use diagpol_common
  use chimere_params
  use chimere_consts

contains

  !!*********************************************************************
  !! Update HeatSum
  subroutine calculate_heatsum(temp,hs_prev,hs_calc)
   implicit none

   ! Input variable
   real,dimension(nzonal,nmerid,ntimes),intent(in)    :: temp
   real,dimension(nemisp,nzonal,nmerid),intent(in)    :: hs_prev

   ! Output variable
   real,dimension(nemisp,nzonal,nmerid),intent(inout) :: hs_calc

   ! Local variables
   real,dimension(nzonal,nmerid) :: temp_tmp
   integer :: izo, ime, inbp, idt ! indexes

   do inbp=1,nemisp
     temp_tmp = 0e0
     do idt=1,ntimes
       temp_tmp = temp_tmp + temp(:,:,idt)
     enddo
     temp_tmp = temp_tmp/ntimes
     where(temp_tmp .gt. pobe(inbp)%temp_cutoff)
       hs_calc(inbp,:,:) = temp_tmp - pobe(inbp)%temp_cutoff
     elsewhere
       hs_calc(inbp,:,:) =0e0
     end where
   end do
   hs_calc = hs_calc + hs_prev
  end subroutine calculate_heatsum

end module diagpol_science
```

E.4 Model file: deppol.f90

```
subroutine deppol

  !  Calculation of deposition velocities and rates
  !  This routine is called only every physical time step
  !  LOCAL VARIABLES
  !        ZNU  : Kinematic viscosity
  !        ZNUD : Dynamic viscosity
  !  INPUTS :
  !            PRESLOC    : Current Pressure
  !            TEMPLOC    : Current Temperature
  !            USTALOC  : Current value of U*
  !            AERRLOC  : Current value of aerodynamic resistance
  !            HLAYLOC  : Heights of model layer tops
  !            DLAND    : Land use type fractions
  !  OUTPUTS:  VAERO  : Deposition rates for particles
  !            VSED  :  Sedimentation velocity


  use chimere_consts
  use worker_params
  use worker_common

  implicit none

  ! local variables
  integer :: imon
  integer :: izo,ime,ivert,nd

  real(kind=8) :: znu,znud,zg,Ra,dd,c,Db
  real(kind=8) :: Rb,zvs,Sc,St
  real(kind=8) :: ustar

  zg=gravit*100.                ! in cm/s2

  !  Month

  imon = imonth(ihourrun)

  !  Loop on Horizontal cells
  ! According Mikhail Sofiev

  do nd=1,ms
     dd=1.d+02*sqrt(ds(nd)*ds(nd+1))
```

```
      do ivert=1,nverti
         do ime=1,nmerid
            do izo=1,nzonal
               call setgas(temploc(izo,ime,ivert),presloc(izo,ime,ivert),&
                           gasmw,denair,viscos,freemp)
               znud = 1d+01*viscos
               znu = (znud/denair)*1d+03
               !c=(1d0+((2.d+02*freemp/dd)*(1.257d0+4d-01*&
               !     exp(-0.55*dd/(1.d+02*freemp)))))
               vsedpol(nd,izo,ime,ivert)=(rhoini/1d+03)*(dd**2)          &
                                         *zg/(1.8d+01*znud)!*c
            end do
         end do
      end do
   end do

   return
end subroutine deppol
```

Appendix F

Chimere efficiency