OLAF — A General Modeling System to Evaluate and Optimize the Location of an Air Polluting Facility

Project Report

Jörg Fliege

Fachbereich Mathematik
University of Dortmund
44221 Dortmund
Germany
email fliege@math.uni-dortmund.de

October 25, 1999

Acknowledgments This work was sponsored by the Region Bruxelles Capital under a "Research in Brussels" grant.

I would hereby like to express my gratitude to Prof. Dr. F. Plastria from the Vrije Universiteit Brussel for the kind hospitality, his encouragement, and the many fruitful discussions I enjoyed with him during my stay at the Vakgroep Beleidsinformatica. Moreover, I would like to thank my coworkers and colleagues of the Faculteit van de Economische, Sociale en Politieke Wetenschappen for their help.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Finding a location for an air polluting facility is usually a source of heated discussions among local inhabitants and decision makers, since the effects of polluted air on human health and the environment can be quite severe.

We propose a general modeling and optimization system for the effects of a low toxic air pollutant on the inhabitants of different land-based ecosystems, including health effects on humans or other beings. Health effects are quantified on the basis of detailed cytodynamic effects in different organs and tissues. The software package OLAF (*O*ptimal *L*ocating *A*ir *P*olluting *F*acilities) developed on the basis of this model does not only evaluate quantitatively the effects of a proposed decision, but uses an efficient optimization technique to optimize the location of the polluting facility with respect to the adverse effects of the pollution on the inhabitants of the modeled region.

Up to now, the decision process with respect to environmental issues and health concerns was mainly based on simulations of a small number of different decisions. However, the need for an automated optimization process incorporating detailed estimates of effects of the pollution has recently been acknowledged by the European Environmental Agency [27, 4]. Unfortunately, up to now only few attempts have been made to integrate the decision process into a general optimization procedure, thereby automatically minimizing a measure of risk. While some of the models developed with respect to this are rather simplistic [21, 20, 1], others use stochastic optimization techniques (see [24], [25] and the references therein), methods which are known for their notoriously slow behavior.

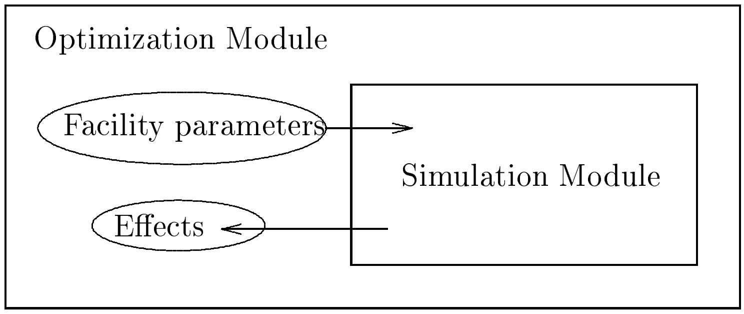

Figure 1.1: General structure of the developed software.

1.2 Overview

The general strategy of the task is clear: after a simulation module has been developed for simulating the environmental effects and the health effects for a given choice of parameters which characterize the polluting facility, an optimization module to be developed can handle the task of finding optimal choices for the describing parameter values (compare Figure 1.1). Since the cause-effect relationships of pollution and human health are quite complicated and dynamic, it has to be expected that only in this way an optimal strategy for pollution control can be developed. Of course, all parts of the cause-effect chain need to be considered.

However, it is quite obvious that a more detailed consideration of the task at hand is necessary. To this end, let us first focus on two main questions.

1.2.1 Two Main Questions

1. What are the parameters of the facility that can be controlled by the decision makers?

2. How can the effects on the environment and on the health of certain individuals be quantified?

While there are no definite answers to these questions, we can give some recommendations, thereby providing at least partial answers.

1. Since we are concerned with locating a *new* facility, we can safely assume that the question "Must the facility be built anyway?" has already been answered positively. It remains to find an acceptable location for the facility. Therefore, the actual location is represented by two (real-valued) unknowns or parameters which have to be optimized. Since we are assuming that the main environmental effect of the facility is airborne pollution, we consider the height of the point of entry of the pollutant

into the atmosphere (e. g. the height of the smokestack) as an additional unknown.

We assume that production-dependent considerations have already fixed the amount of pollutant released in a given time period, and we assume that the amount released is constant in time. As a consequence, we are naturally led to the conclusion that all the physical and toxicological characteristics of the pollutant are prespecified. This leaves us with the width of the opening of the "smokestack" (or, alternatively, with the speed of the exhaust gas) as one additional decision variable. For more details, we refer to Chapter 5.

At the present moment, environmental policy issues like the choice of a specific washing system for the exhaust air are not of concern for us. Adding them to the optimization process is certainly possible, but the discussion of the optimal use of these technologies goes beyond the scope of this work.

2. On an individual level, a well-established measure for carcinogenic effects within a certain organ or tissue is the number of malignant cells appearing within a certain time period. Likewise, a measure for cytotoxic effects is the number of cells which have died during a fixed time period. Note, however, that detrimental effects due to toxicity not leading to the death of certain cells can not be accounted for in this scheme. Since we are dealing with low-toxic doses, actual outbreaks of symptoms are not an issue here.

   Taking now the position of a postulated central authority (i. e. a governmental health care official), *the total sum of all health effects over all individuals has to be minimized.* The same analysis holds with respect to environmental issues.

   More details with respect to the evaluation of the health effects can be found in Chapter 5.

The refined structure of the system is depicted in Figure 1.2.

1.2.2 The Submodules

Having given a more detailed account on the decision variables and the quantification of the effects to be minimized, we can now turn our attention to the simulation of the fate of a pollutant through the biosphere. It was chosen to develop a general modeling system to simulate the following subsystems of the environment and the fate of the pollutant in them:

1. atmosphere,

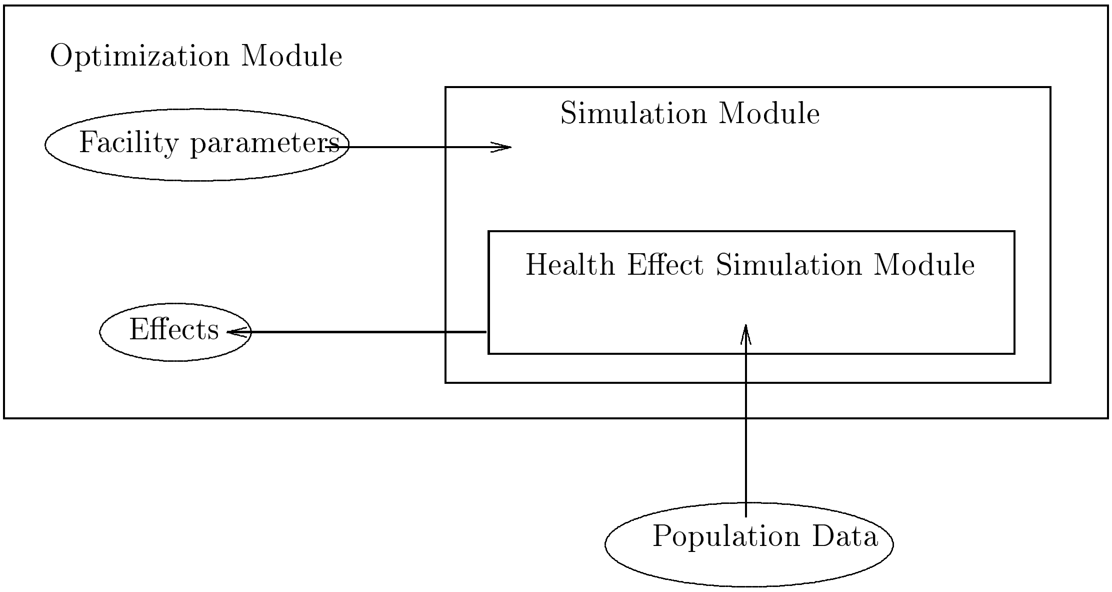

Figure 1.2: General structure of the software developed including health effects.

2. land-based ecosystems,
3. individuals, and
4. individual organs.

Of course, this list of subsystems is not exhaustive. We deliberately skipped the geosphere, the aquatic distribution of pollutants, the corresponding water-based ecosystems, the simulation of the effects of a pollutant on cells, etc.

Note that the meteorological conditions of the atmosphere are independent of the fate of the pollutant and of the location of the facility, but not vice versa. It makes therefore sense to calculate the meteorological variables needed over the time period of interest beforehand and use this data in subsequent simulation runs inside the optimization module.

The discussion above leads in a natural way to a breakdown of the software system into the following submodels and modules.

1. The *meteorological preprocessor*, which uses observational meteorological data to compute all the meteorological parameters needed for the time period of interest in the spatial domain considered,
2. the *air dispersion model*, in which the distribution of the pollutant in the atmosphere is simulated,

3. the *ecosystem dispersion model*, which simulates the fate of the pollutant in the ecosystems considered, through different trophic levels, through food chains, etc.,

4. the *chemokinetic model*, in which the concentration of the pollutant in different organs of different individuals, living in the abovementioned ecosystems, is traced,

5. the *cytodynamic multistage model*, which simulates the effects of the pollution on different cell populations in the organs of the individual life forms considered,

6. the *effect quantification module*, agglomerating the cytodynamic effects on the individuals to an effect at the population level, and finally

7. the *optimization module*, which minimizes this effect by optimizing the decision variables specified in Section 1.2.1.

The corresponding data flow between the submodules is depicted in Figure 1.3. In the following chapters, each of these parts is described in detail.

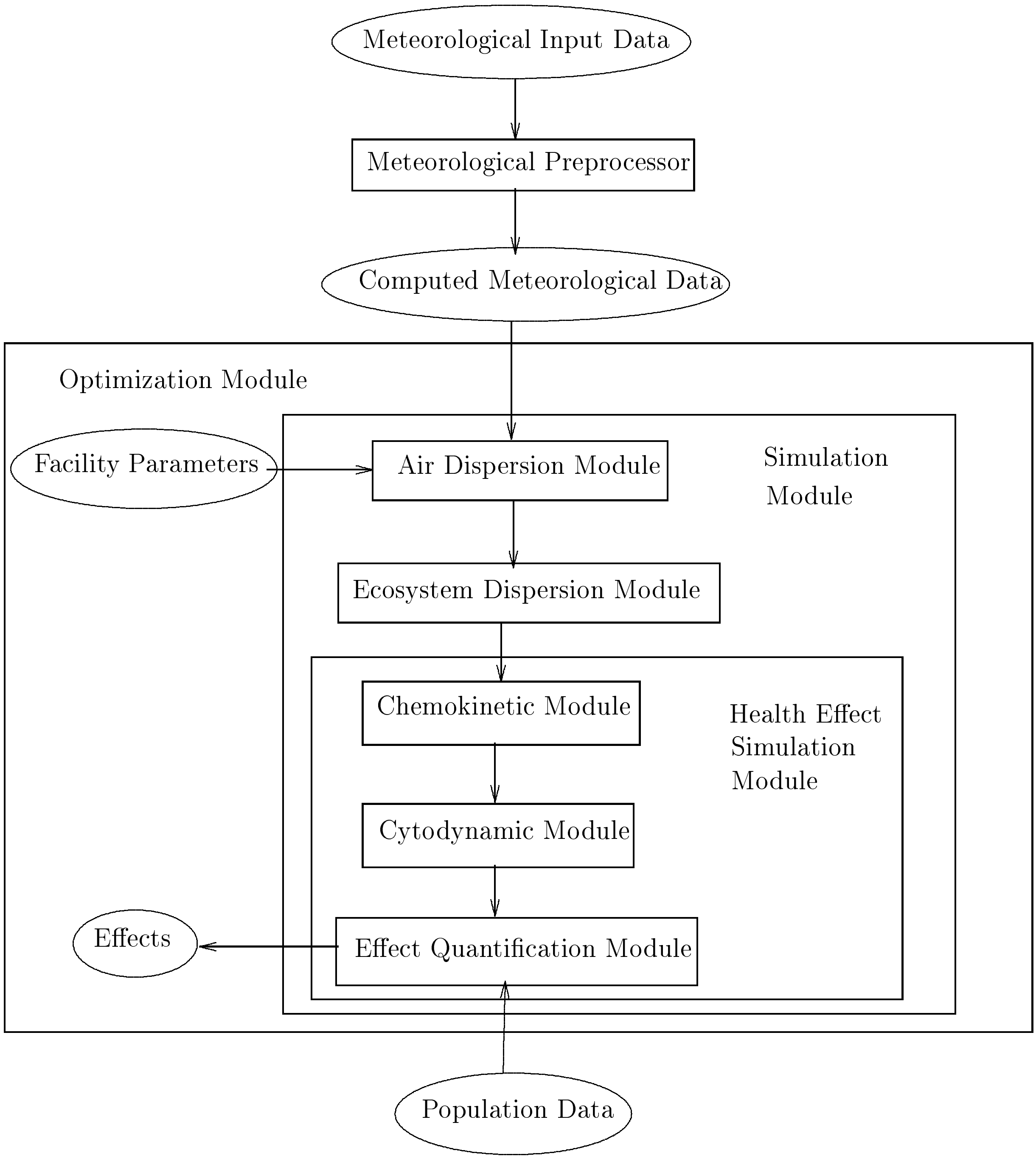

Figure 1.3: Submodules and data flow.

Chapter 2

Meteorology

2.1 The Meteorological Preprocessor

A meteorological preprocessor takes observational meteorological data from a given time period and a given spatial domain and computes out of this sparse data set all the meteorological information necessary to calculate pollutant concentrations in the air later on.

At the present moment, the preprocessor in use is a modified version of the MESOPAC II meteorological preprocessor [33, 39], developed for the Environmental Protection Agency of the U.S.A. This preprocessor interpolates given meteorological data in time and space and produces a gridded field of data in two vertical layers of the atmosphere. The grid dimensions of the rectangular domain of interest have to be specified by the user of the preprocessor by way of a control file. This grid is the *computational grid*, whose grid points represent all the receptor locations of the pollutants. Each of these receptor locations contains an ecosystem and some individuals whose response on the pollution is modeled in other modules later on.

The following data is necessary for a prespecified number of meteorological surface stations within a prespecified time interval for which the preprocessor has to compute data: hourly data with respect to cloud ceiling height, amount and type of precipitation in CD-144 coding, wind direction (in degree), wind speed, surface pressure, dry bulb temperature, relative humidity, and opaque cloud cover (in %). Of course, the latitude and longitude of the station providing the data has to be specified, too. Moreover, additional data from upper air stations providing rawinsonde data can be added to the data set: twice daily data with respect to soundings at different levels. At each level, data with respect to height, temperature, wind direction (in degree), and wind speed has to be given. Additionally, hourly precipitation data from surface stations can be added to the input. In any case, so called "land use data" has to be provided to allow for the computation of the friction velocity and the wind speed at the top of the surface layer. This land use data takes the form of a categorization of each surface grid point of the computational field specified according to twelve different categories.

The output of the MESOPAC II preprocessor consists of a binary file with the following data: at each grid point of the computational field, hourly data with respect to the lower-level wind field, the upper level wind field, mixing height, friction velocity, convective velocity scale, Monin-Obukhov length, PGT stability class, precipitation rate, surface air density, air temperature, solar radiation, relative humidity, and type of precipitation.

More details with respect to the preprocessor can be found in the user's manual [39].

Since the meteorological data set provided by the preprocessor is independent of the parameters of the pollutant and its source, the preprocessor needs to be called only once during an optimization run. Afterwards, the necessary data will be available in a file. This strategy has been followed for the design of the code. Subsequent computational tests showed significant savings in computation time.

Note that the preprocessor in use is diagnostic instead of prognostic, i. e. the input data consists of measurements, which are then interpolated according to a technique based on Voronoi-diagrams. In contrast to this, prognostic models are used, e. g., in forecasting and simulate the variables under consideration according to the underlying physical laws. It has to be expected that prognostic preprocessors are able to deliver more accurate data. On the other hand, diagnostic codes seem to be more efficient with respect to computation time. In any case, the diagnostic model used here can be easily replaced by any other kind of code, as long as the output data format is the same. The choice of the specific preprocessor used here was made due to the simplicity and the robustness of the approach.

2.2 The Air Dispersion Model

The air dispersion model takes as input the data computed by the meteorological preprocessor, the locational characteristics of the polluting facility, and the physical and chemical characteristics of the pollutant, and computes out of this given data the concentration of the pollutant in the air at ground level within the time window considered.

Let $\Omega \subset \mathbb{R}^3$ be the domain of interest. For a point $x \in \Omega$ and a time $t > 0$, we are interested in the pollutant concentration $c(x,t)$ at the point x at time t. We assume that the pollutant concentration at time $t = 0$ is known everywhere. As additional data, let a wind field $u : \Omega \times [0, +\infty[\longrightarrow \mathbb{R}^3$, a source term $I : \Omega \times [0, +\infty[\longrightarrow \mathbb{R}$ and a depletion term $D : \Omega \times [0, +\infty[\longrightarrow \mathbb{R}$ be given. The source term represents an input of pollutant into the atmosphere by way of, e. g., a polluting facility, while the depletion term represents deposition of the pollutant on the ground. The advection-diffusion equation governing the pollutant concentration in Ω is then given by

$$\frac{\partial c}{\partial t} = -\nabla \cdot (cu) + \nabla \cdot (\Sigma \nabla c) + I - D \tag{2.1}$$

(see, e. g., [14, 30]; arguments in the formula above have been omitted). Here, Σ is the diffusion matrix. Note that the formulation above uses Cartesian coordinates, which means that we assume the earth is flat. If several different pollutants are considered, the quantity $c(x,t)$ is a vector in the $\mathbb{R}^n$, where n is the number of pollutants. The formulation above then still applies, and equation (2.1) has to be interpreted as an equation in $\mathbb{R}^n$. However, chemical reactions between the different pollutants have then to be taken into account also, and an additional reaction term $R(c(x,t)) \in \mathbb{R}^n$, which is usually modeled as a quadratic function in c, has to be added to the formulation above.

The amount of literature concerning solution methods to solve (2.1) is monumental, see, e. g. the review article [32], which lists more than 100 different algorithms. We will make use here of a particular scheme which exploits the fact that (2.1) has an analytic solution in the special case $u(x,t) = \text{const}$ for all (x,t), $D \equiv 0$, I a Dirac functional of the form $I(z,0) = m > 0$ and $I(x,t) = 0$ for all $(x,t) \neq (z,0)$, and $c(x,0) = 0$ for all $x \neq z$. This means that the only polluting source in Ω is at the point $z \in \Omega$, and it injects once an amount m of pollutant into the atmosphere. Assuming, additionally, $u \equiv 0$, we get the solution

$$c(x_1,x_2,x_3,t) = \frac{m}{\sqrt{\pi \sigma_x \sigma_y \sigma_z t^3}} \exp\left(-\frac{(x_1-z_1)^2}{4\sigma_x t} - \frac{(x_2-z_2)^2}{4\sigma_y t} - \frac{(x_3-z_3)^2}{4\sigma_z t} \right), \tag{2.2}$$

which can be interpreted as a Gaussian "puff" in three-dimensional space around the origin. Here, σ_x, σ_y, and σ_z are the dispersion parameters. Other types of solutions exist under similar conditions [39, Chapter 6], [29, 37].

The basic strategy is now to discretize the given nonhomogeneous wind field and approximate it by a piecewise constant function. In each region of continuity of the approximation, the concentration function c can then be discretized according to (2.2) into a finite number of puffs, which are emitted at the pollutant point at discrete time steps. These puffs are then advected through the (original) wind field according to the trajectory that the movement of the puff center defines in the field. The trajectory path $x(t)$ ($t \geq t_0$, $t_0 \geq 0$ the emitter time) of a puff center is computed by solving the nonlinear ordinary differential equation

$$x'(t) = u(x(t),t), \quad t \geq t_0, \qquad x(t_0) = z. \tag{2.3}$$

Since the wind field is computed by the preprocessor at the predefined grid points only, it has to be interpolated, which is usually done bilinearly in time and space. The dispersion parameters, which are time- and space-dependent, are updated for each puff separately. Note that this dispersion model is mass-conservative, a feature of high importance in air pollution modeling [30, 41].

The air dispersion code used in OLAF is a modified version of the MESOPUFF II puff superposition code [33, 39, 6]. The MESOPUFF II code is able to simulate the dispersion of several different pollutants on a regional scale, i. e. on a scale of ca. 1000km in west-east direction and ca. 600km in north-south direction. This makes the package suitable for medium to long-range transportation simulation. The ordinary differential equation (2.3) which has to be solved to advect the puffs through the wind field is solved with a Runge-Kutta-formula of order 2 given by the Butcher scheme [11]

$$\begin{array}{c|cc} 0 & 0 & \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2\ . \end{array}$$

The dispersion parameters are recalculated after each time step according to empirical update formulae. If the puff center has traveled less than 100km, they take the familiar form $\sigma = ad^b$ (cmp. [38, 42]), where a, b are coefficients calculated according to the stability of the atmosphere at the present point and d is the total distance the puff center has traveled. For larger distances, other update formulae are used [13].

Simple atmospheric chemistry as well as wet and dry deposition processes are handled, too. Deposition processes are handled in a mass-conservative way, again. All the deposition takes place on the predefined Cartesian grid. Moreover, several pollutant sources including polluting areas can be present in the computational domain. In this way, a background pollution of the atmosphere can be modeled. After the dispersion model has been run, the concentration data computed can be found in a binary file. If fluxes to the ground are calculated, too, the corresponding gridded data is written to additional binary files.

As it is the case for the preprocessor, the air dispersion code can be replaced by an arbitrary code capable of computing approximations to the desired values as long as the format of the input and the output stays the same. Otherwise, some coding effort to provide the corresponding translation routines is needed.

More details with respect to the dispersion model can be found in the user's manual [39] and in an application protocol [6].

Chapter 3

Ecology and Chemokinetics

3.1 Compartment Models

In this and the following section, we describe the standard compartment model, some nonstandard extensions to it, the underlying mathematics and the corresponding numerical treatment. In Section 3.2–3.5 we show what kind of natural processes can be described by compartment models, and we give some examples for their use.

3.1.1 The Standard Model

Both the ecosystem dispersion model and the chemokinetic model are described in this chapter, since the modeling paradigm used for the two submodels is the same.

The ecosystem dispersion model takes as input the time- and space-dependent air concentration of the pollutant at the predefined grid points and follows the fate of the pollutant across the different trophic levels of the ecosystems considered. Each grid point of the air dispersion model represents one local food web, through which the trace of the pollutant is followed.

Note that population dynamics are not accounted for, since there does not seem to be a general mechanism for population dynamics, valid for all kinds of interacting populations [10, 28].

The chemokinetic model takes as input the time- and space-dependend exposure of an individual to the pollutant under consideration and computes the uptake and the distribution of the chemical in the different body tissues and organs. Several different types of individuals can be simulated by specifying the corresponding parameters. The metabolism of the pollutant under consideration is modeled by Michaelis-Menten kinetics.

Both codes simulating the processes described above are based on a compartment model of the system to be simulated [19, 37] (also called continuously stirred tank reactor (CSTR) model). These systems are of the following type, and one of these systems is set up for each grid point of the computational grid of the atmospheric dispersion model. The modeler divides the total system under consideration (i. e. the ecosphere and the geosphere) into $n \geq 1$ different *compartments*. The total mass of pollutant over time in a compartment k, $k = 1, \ldots, n$, is then given by a function $m_k : [0, +\infty[\longrightarrow \mathrm{IR}$, which has to be computed. Again, note that we have $m_k(t) = m_k(x, t)$ where $x \in \Omega$ is a variable denoting the point in space where the compartment system resides. In what follows, we skip this dependency for ease of notation. In the compartment model implemented in OLAF, compartment no. 1 is always the "air compartment", and the total mass in it is represented by the pollutant concentration calculated by the MESOPUFF II model (see Section 2.2). As a consequence, the function m_1 can be considered as an external variable which does not need to be calculated here any more. By the first law of thermodynamics, the change of mass in compartment no. k ($k \neq 1$), $\partial m_k(t)/\partial t$, is the difference between the inflow and the outflow to other compartments. The inflow from compartment no. $j \neq k$ to compartment no. k is proportional to the mass inside compartment j, while the outflow out of compartment k to compartment no. i is proportional to the mass in compartment k. As a consequence, the following ordinary differential equation holds for m_k:

$$m_k'(t) = \sum_{\substack{j=1 \\ j \neq k}}^{n} \lambda_{j,k}(t) m_j(t) - \left(\sum_{\substack{i=1 \\ i \neq k}}^{n} \lambda_{k,i}(t) \right) m_k(t),$$

where $\lambda_{j,k}(t) \geq 0$ is the time-dependent proportionality constant for the flow from compartment j to compartment k, and $\lambda_{k,i}(t)$ is the time-dependent proportionality constant for the flow from compartment k to compartment i ($i \neq k \neq j$). These parameters are dimensionless. The differentiation symbol denotes differentiation with respect to t. Again, we have suppressed in the notation used here any dependency on the spatial grid point considered. In what follows, we assume without loss of generality $\sum_{i=1, i \neq k}^{n} \lambda_{i,k}(t) > 0$ for $k = 2, \ldots, n$ and all $t \geq 0$, i. e. at any time there is at least one inflow into each non-air compartment considered. Define now the matrix $\Lambda(t) \in \mathrm{IR}^{n \times n}$ by using, as usual, $\lambda_{k,i}(t)$ as the entry in row k and column i and define the diagonal elements by

$$\lambda_{k,k}(t) := - \sum_{\substack{i=1 \\ i \neq k}}^{n} \lambda_{k,i}(t). \tag{3.1}$$

When using the vector notation $m(t) = (m_1(t), \ldots, m_n(t))^\top$, we see that the equations

$$m'(t) = \Lambda^\top(t) m(t), \quad t \geq 0, \tag{3.2}$$

hold, where the starting condition $m(0)$ depends on the spatial location, i. e. the grid point of the computational grid used. Note that the function m_1 is given and does not have to be computed.

In the current version of the OLAF system, a back flow from an arbitrary compartment to the air compartment is not handled, since it has to be expected that the quantities involved flowing back are too small to be tracked numerically once they are added to the large amount of mass in the air compartment. As a consequence, it is possible to use values $\lambda_{k,1} > 0$ for some $k = 2, \ldots, n$ to model a release of pollutant from a compartment to the air. But, while the mass-balance of the compartments $k = 2, \ldots, n$ is correctly computed (i. e. a certain mass is deducted), the corresponding mass-balance of the air compartment is not taken into account. Moreover, instead of the mass of the air compartment, the concentration in it is computed by MESOPUFF. This means that instead of dimensionless scalars $\lambda_{1,k}(t)$ $(k = 2, \ldots, n)$ values $\hat{\lambda}_{1,k}(t)$ with dimension m^3 have to be used. Now let $c(t)$ be the concentration of pollutant in the air at the time $t \geq 0$. (Again, we suppress the dependence on the spatial scale.) We then have to solve the system

$$\begin{pmatrix} m_2'(t) \\ \vdots \\ m_n'(t) \end{pmatrix} = \begin{pmatrix} \lambda_{2,2}(t) & \cdots & \lambda_{n,2}(t) \\ \vdots & \ddots & \vdots \\ \lambda_{2,n}(t) & \cdots & \lambda_{n,n}(t) \end{pmatrix} \begin{pmatrix} m_2(t) \\ \vdots \\ m_n(t) \end{pmatrix} + c(t) \begin{pmatrix} \hat{\lambda}_{1,2}(t) \\ \vdots \\ \hat{\lambda}_{1,n}(t) \end{pmatrix}, \quad (3.3)$$

which is a linear inhomogeneous system of ordinary differential equations of dimension $n - 1$.

Mass Conservation and Nonnegativity

A simple observation based on (3.1) is

$$\sum_{k=1}^{n} m_k'(t) = 0 \text{ for all } t \geq 0,$$

which means that the model is mass-conservative, i. e.

$$\sum_{k=1}^{n} m_k(t) = \sum_{k=1}^{n} m_k(0) = \text{const for all } t \geq 0. \quad (3.4)$$

Let $\hat{m}_k(t)$ be a numerically computed approximation to $m_k(t)$ $(k = 2, \ldots, n)$. In view of property (3.4), it makes sense to impose either mass conservation (in time) on the computed approximation or nonnegativity, i. e. the restriction $\hat{m}_k(t) \geq 0$ for all k, or both. Mass conservation is a natural concern for environmental simulations [30, 41]. If nonnegativity does not hold, physically meaningless mass values are calculated for a time t, and this error might propagate in later time steps. Moreover, the cytodynamic module in use might not necessarily be able to handle those erroneous values. In such a case, a filtering technique has to be used to "redistribute" negative masses such that nonnegativity or mass-consistency holds [40]. Consequently, implementations based on such techniques are not very transparent. Numerical issues with respect to nonnegativity and mass conservation will be discussed in detail in the next subsection. First, let us focus on the nonnegativity of the exact solution of (3.3).

Denote by

$$A(t) := \begin{pmatrix} \lambda_{2,2}(t) & \cdots & \lambda_{n,2}(t) \\ \vdots & \ddots & \vdots \\ \lambda_{2,n}(t) & \cdots & \lambda_{n,n}(t) \end{pmatrix}$$

the system matrix of (3.3), which is also the Jacobian of the right hand side with respect to m. The nonnegativity of the solution $(m_2, \ldots, m_n)^\top$ of (3.3) under quite weak conditions can now be seen as follows. Let A be piecewise uniform continuous and c as well as all $\hat{\lambda}_{1,k}$ $(k = 2, \ldots, n)$ be piecewise continuous. Moreover, let those functions be right-sided continuous everywhere. Interpolate A by a piecewise constant function $\tilde{A}$ on a grid $0 =: t_0, t_1, t_2, \ldots$ containing the points of discontinuity of both A, c, and $\hat{\lambda}_{1,k}$ $(k = 2, \ldots, n)$ and such that $\tilde{A}(t_i + h) = \tilde{A}(t_i) = A(t_i)$ for all $h \in [0, t_{i+1} - t_i[$ $(i = 0, 1, 2, \ldots)$. If A is replaced by $\tilde{A}$ in (3.3), the corresponding solution $\tilde{m}$ on the interval $[t_i, t_{i+1}[$ can be written as

$$\begin{pmatrix} \tilde{m}_2(t) \\ \vdots \\ \tilde{m}_n(t) \end{pmatrix} = \mathrm{e}^{(t-t_i)A(t_i)} \begin{pmatrix} m_2(t_i) \\ \vdots \\ m_n(t_i) \end{pmatrix} + \int_{t_i}^{t} \mathrm{e}^{(s-t_i)A(t_i)} c(s) \begin{pmatrix} \hat{\lambda}_{1,2}(s) \\ \vdots \\ \hat{\lambda}_{1,n}(s) \end{pmatrix} \mathrm{d}s$$

$(i = 0, 1, 2, \ldots)$, where, as usually, the matrix exponential function is given by

$$\mathrm{e}^M := \sum_{i=0}^{\infty} \frac{1}{i!} M^i$$

for arbitrary square matrices M. Since the nondiagonal elements of A are nonnegative, all the elements of $\mathrm{e}^{(t-t_i)A(t_i)}$ are nonnegative for all $t \geq t_i$. If $\hat{\lambda}_{1,k}(t_i), \tilde{m}_k(t_i) \geq 0$ $(k = 2, \ldots, n,\ i = 0, 1, 2, \ldots)$ holds, the nonnegativity of the functions $\tilde{m}_k$ $(k = 2, \ldots, n)$ for $t \geq t_i$ follows if, in addition, c is nonnegative. But $\tilde{m}$ converges uniformly to the solution m of (3.3) since $\tilde{A}$ and $\tilde{c}$ converge uniformly to A and c when the grid is refined.

Numerics

Note that all entries in the system matrix of (3.3) except those on the diagonal are nonnegative, while the diagonal entries are nonpositive. Moreover, the inhomogeneous part is nonnegative (or, more precisely, strictly positive for compartment $k \neq 1$) if and only if the air compartment contains a positive pollutant mass and there is an exchange between the air compartment and compartment k. Consider for the moment the case that the inhomogeneous part is zero, i. e. $m_1 \equiv 0$ or $\hat{\lambda}_{1,2} \equiv \cdots \equiv \hat{\lambda}_{1,n} \equiv 0$, and that no backflow to the air compartment takes place, i. e. $\lambda_{2,1} \equiv \cdots \equiv \lambda_{n,1} \equiv 0$. As a consequence, all column sums of the matrix $A(t)$ are zero for all t, and any consistent multistep or Runge-Kutta method for solving (3.3) will lead to mass consistency, regardless what step size is used and no matter if the method is explicit or implicit.

Due to (3.1) and the theorem of Gershgorin, the eigenvalues of $A(t)$ have only nonpositive real parts (for all $t \geq 0$), and the only possible eigenvalue with real part equal to zero is 0. This makes the system stiff, while a lower bound for the real part of the eigenvalues at time t is

$$\min\{\lambda_{k,k}(t) \mid k = 2, \ldots, n\} = \min\left\{ -\sum_{\substack{i=1 \\ i \neq k}}^{n} \lambda_{k,i}(t) \,\middle|\, k = 2, \ldots, n \right\} < 0.$$

At first glance an implicit numerical solver should be used to solve (3.3). But implicit methods require the solution of a linear system in $n-1$ variables per time step. In contrast to this, an explicit solver just needs one matrix-vector multiplication to compute the next approximation. Moreover, the accuracy demand in the application discussed here is rather moderate, since the inhomogeneous part of (3.3) cannot be calculated with high accuracy. As a consequence, we use Euler's method in the present implementation of the compartment model within OLAF. Nonnegativity of the iterates is achieved by the following step size control. In the homogeneous case, we have

$$m^{t+1} := m^t + hA^t m^t = \begin{pmatrix} 1 + h\lambda_{2,2}^t & \cdots & h\lambda_{n,2}^t \\ \vdots & \ddots & \vdots \\ h\lambda_{2,n}^t & \cdots & 1 + h\lambda_{n,n}^t \end{pmatrix} m^t$$

as the iteration procedure. Here, the superscripts denote the time steps for the approximations $m^t, m^{t+1} \in \mathbb{R}^{n-1}$ that we compute with step length $h > 0$. Given a (componentwise) nonnegative vector m^t, the nonnegativity of m^{t+1} can be assured if $1 + h\lambda_{k,k}^t \geq 0$ for $k = 2, \ldots, n$. The step size chosen by OLAF is then

$$h := \min_{k=2}^{n} -\frac{1}{\lambda_{k,k}^t} = \min_{k=2}^{n} \frac{1}{\sum_{i=2, i\neq k}^{n} \lambda_{k,i}^t}, \tag{3.5}$$

the largest one possible under this constraint. Note that larger step sizes can also be allowed, e. g. if

$$\varrho_k^t := \sum_{\substack{i=2 \\ i \neq k}}^{n} \lambda_{i,k}^t m_i^t - m_k \sum_{\substack{i=1 \\ i \neq k}}^{n} \lambda_{k,i}^t < 0$$

for a $k \geq 2$, $k \leq n$, then use a step size h with

$$h \leq \frac{-m_k^t}{\varrho_k^t}. \tag{3.6}$$

This rule depends then on the values of m^t. Even larger steps can be made if the inhomogeneous term is used in the above estimate:

$$h \leq \frac{-m_k^t - \lambda_{1,k} c^t}{\varrho_k^t} \tag{3.7}$$

for all $k = 2, \ldots, n$ with $\varrho_k < 0$.

With the implicit Euler method,

$$m^{t+1} = m^t + hA^{t+1} m^{t+1},$$

i. e.

$$m^{t+1} = (I - hA^{t+1})^{-1} m^t \tag{3.8}$$

one may use arbitrary step sizes $h > 0$ without violating the nonnegativity of m^{t+1}. This can be seen in the following way. The unique solution $\hat{m}$ of the homogeneous version of (3.3),

$$\hat{m}'(t) = A(t)\hat{m}(t), \quad t \geq 0, \qquad \hat{m}(0) = m_0 \geq 0 \tag{3.9}$$

is nonnegative. Now write $A(t) = N(t) - D(t)$ with the positive diagonal matrix $D(t) := \operatorname{diag}(\lambda_{2,2}(t), \ldots, \lambda_{n,n}(t))$ and define the operator

$$P(t, y) := \begin{pmatrix} \max\{(N(t)y)_1, 0\} \\ \vdots \\ \max\{(N(t)y)_n, 0\} \end{pmatrix}$$

for arbitrary vectors $y \in \mathrm{IR}^n$ and $t \geq 0$. Note that $P(t, y) \geq 0$ for all y and all $t \geq 0$. With this, the system of differential equations

$$\tilde{m}'(t) = P(t, \tilde{m}(t)) - D(t)\tilde{m}(t), \quad t \geq 0, \qquad \tilde{m}(0) = \hat{m}(0) \geq 0 \tag{3.10}$$

has a unique solution $\tilde{m}$. But then $\tilde{m} = \hat{m}$, and consequentially $P(t, \tilde{m}(t)) - D(t)\hat{m}(t) = A(t)\hat{m}(t)$ for all $t \geq 0$. Therefore, employing the implicit Euler method on (3.10) leads to the same approximations $\tilde{m}^t$ as if the method has been used to solve (3.9). But then

$$\tilde{m}^{t+1} + hD^{t+1}\tilde{m}^{t+1} = \tilde{m}^t + P(t + h, \tilde{m}^{t+1}),$$

and when the right hand side is nonnegative (by induction), $\tilde{m}^{t+1}$ will also be nonnegative. Note that this reasoning works only if the system of linear equations (3.8) is solved exactly, i. e. with a direct method and not an iterative one.

Therefore, a nonnegativity enforcing version of Euler's implicit method needs $O(n^3)$ floating point operations per time step, as compared to the $O(n^2)$ floating point operations for the explicit version. More precise estimates depend on the sparsity structure of the system matrix A.

Note, however, that computing an approximative solution of this system of ordinary differential equations is necessary for each point of the computational grid of the air dispersion code, and that any step size rule depending on m^t decouples the time grid of the different systems to be solved. This poses a problem when the actual objective function has to be evaluated, because each mass in each compartment is then also equipped with a time interval representing the length of stay of the mass inside the compartment. Due to this, further tests with step size rules like (3.6) or (3.7) have been postponed.

While atmospheric dispersion models usually use a rather large time scale, chemokinetic models do not. Typical time steps in atmospheric dispersion are one hour or several hours, while the scale needed in the chemokinetic model is, according to what is said above, only minutes. As a consequence, the concentration of the pollutant in the air as used in the ecosystem dispersion and the chemokinetic model is approximated by a piecewise constant function. While it is possible to compute out of this data a smooth approximation, it is not clear how mass balance can be achieved. On the other hand, nonsmooth right-hand sides in (3.3) pose serious numerical problems to the corresponding solver. Due to this, Euler's method is used only from one time step of the atmospheric dispersion model to the next. The last step size used in Euler's method is computed in such a way that the step taken coincides with the end point of the interval in which the current constant approximation of pollutant concentration in the air holds. The next step of Euler's method uses the new pollutant concentration from the next time step of the atmospheric dispersion model.

3.1.2 Metabolism

The metabolism of a pollutant due to microbial degradation is modeled according to Monod kinetics. Note that the Michaelis-Menten kinetics, widely used to model enzymatic reactions affecting the concentration of pollutants in different body tissues and organs, are completely analogous to the Monod kinetics, and we will therefore use the corresponding terms interchangeably. In a situation governed by Michaelis-Menten kinetics, the rate of change of the mass of pollutant in the compartment no. k ($2 \le k \le n$) is additionally affected by a term of the form

$$-\frac{v_k m_k(t)}{\delta_k + m_k(t)},$$

where $v_k \geq 0$ is the maximal induction level of the enzyme connecting to the pollutant and $\delta_k \geq 0$ is the dissociation constant of the enzyme. These two parameters have to be specified by the modeler. As a consequence, the system of ordinary differential equations that has to be solved is not (3.3), but

$$\begin{pmatrix} m_2'(t) \\ \vdots \\ m_n'(t) \end{pmatrix} = \begin{pmatrix} \lambda_{2,2} & \cdots & \lambda_{n,2} \\ \vdots & \ddots & \vdots \\ \lambda_{2,n} & \cdots & \lambda_{n,n} \end{pmatrix} \begin{pmatrix} m_2(t) \\ \vdots \\ m_n(t) \end{pmatrix} + c(t) \begin{pmatrix} \lambda_{1,2} \\ \vdots \\ \lambda_{1,n} \end{pmatrix} - \begin{pmatrix} v_2 m_2(t)/(\delta_2 + m_2(t)) \\ \vdots \\ v_n m_n(t)/(\delta_n + m_n(t)) \end{pmatrix}.$$

Of course, this additional nonlinearity destroys any mass-conservation properties that the model might had beforehand. Nevertheless, mass conservation with respect to the kinetics is an important feature of any model and code trying to analyze quantitatively the effects of pollutants on individuals, and we believe that software whose kinetic model is not mass conservative will have serious difficulties in predicting the aforementioned effects. In any case, metabolism and degradation will usually take place only in a few selected compartments, such that most of the nonlinear terms just introduced will vanish.

3.2 Ecosystems and Food Webs

Ecosystems and food webs can be described using a compartment model. In principle, each compartment represents one species or one type of species. Species which feed on each other are connected by an interchange flow modeling their predating pattern, as described in the last section.

Of course, the more detailed the ecosystem description is, the more detailed and more realistic the compartment model will be. For example, it is possible to divide a species into several distinct classes, each living spatially separated or equipped with distinctive anatomic features. Each of these classes would then be represented by a separate compartment.

To give the reader some inkling of what can be achieved with this kind of model, we describe some simple examples from the literature for ecosystems or parts of ecosystems modeled as compartment models.

Example 1 ([37]) The uptake of 'Seveso-dioxin' 2,3,7,8-TCDD via grass and cows and it's deposition in milk can be modeled according to the following simple food chain depicted in Figure 3.1: TCDD is deposited as particulate matter on the grass or washed onto the soil by wet deposition processes. Afterwards, grazing cows feed on the grass and take the pollutant up into their

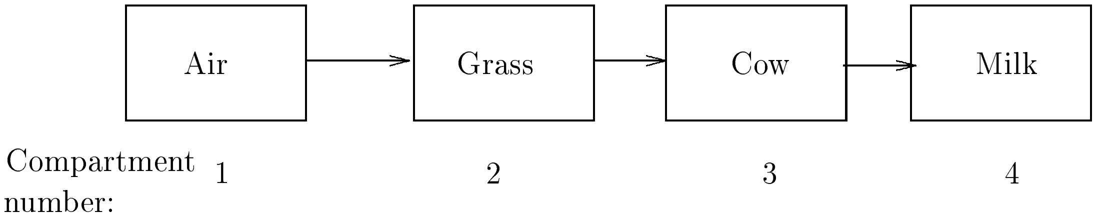

Figure 3.1: Simple TCDD food chain.

bodies. Once inside, it is mainly deposited in the milk of the cows. This last deposition process is not strictly a process on ecosystem level.

Example 2 ([18]) Uptake and distribution of DDT in an aquatic ecosystem can be modeled by the following simple model, involving eight compartments (cmp. Figure 3.2, page 26): air, suspended DDT, DDT in water phase, DDT in sediment, degraded DDT, DDT in the food chain, DDT in harvested fish, DDT in excretions. Obviously, the food chain compartment can be broken up into smaller compartments.

Example 3 ([37]) The exposure pathways of humans due to soil contamination of air pollutants can be described by a compartment model with nine compartments (cmp. [37, Chapter 7]): air, dust, soil, plants, ground water, surface water, livestock, fish, and man. The plant compartment can be further subdivided into leaves, fruits, and roots, or into different plant species.

Example 4 ([19]) Eutrophication by phosphor for aquatic ecosystems can be modelled with the following twelve compartments (cmp. [19, Chapter 2]): dissolved phosphor in epilimnion, algae in epilimnion, zooplancton in epilimnion, herbivorous fish in epilimnion, carnivorous fish in epilimnion, detritus in epilimnion, dissolved phosphor in hypolimnion, algae in hypolimnion, zooplancton in hypolimnion, herbivorous fish in hypolimnion, carnivorous fish in hypolimnion, and interstitial water.

3.3 Individual based Chemokinetic Models

The chemokinetic model used by OLAF is a compartment model. The modeler has to divide the body of a modeled individual into as many compartments as he or she wishes and define the pathways for the pollutant between the body compartments and the ecosystem compartments defined previously. Usually, each body compartment represents a specific organ or a specific tissue type. An extremely small example is depicted in Figure 3.3. A model of such a

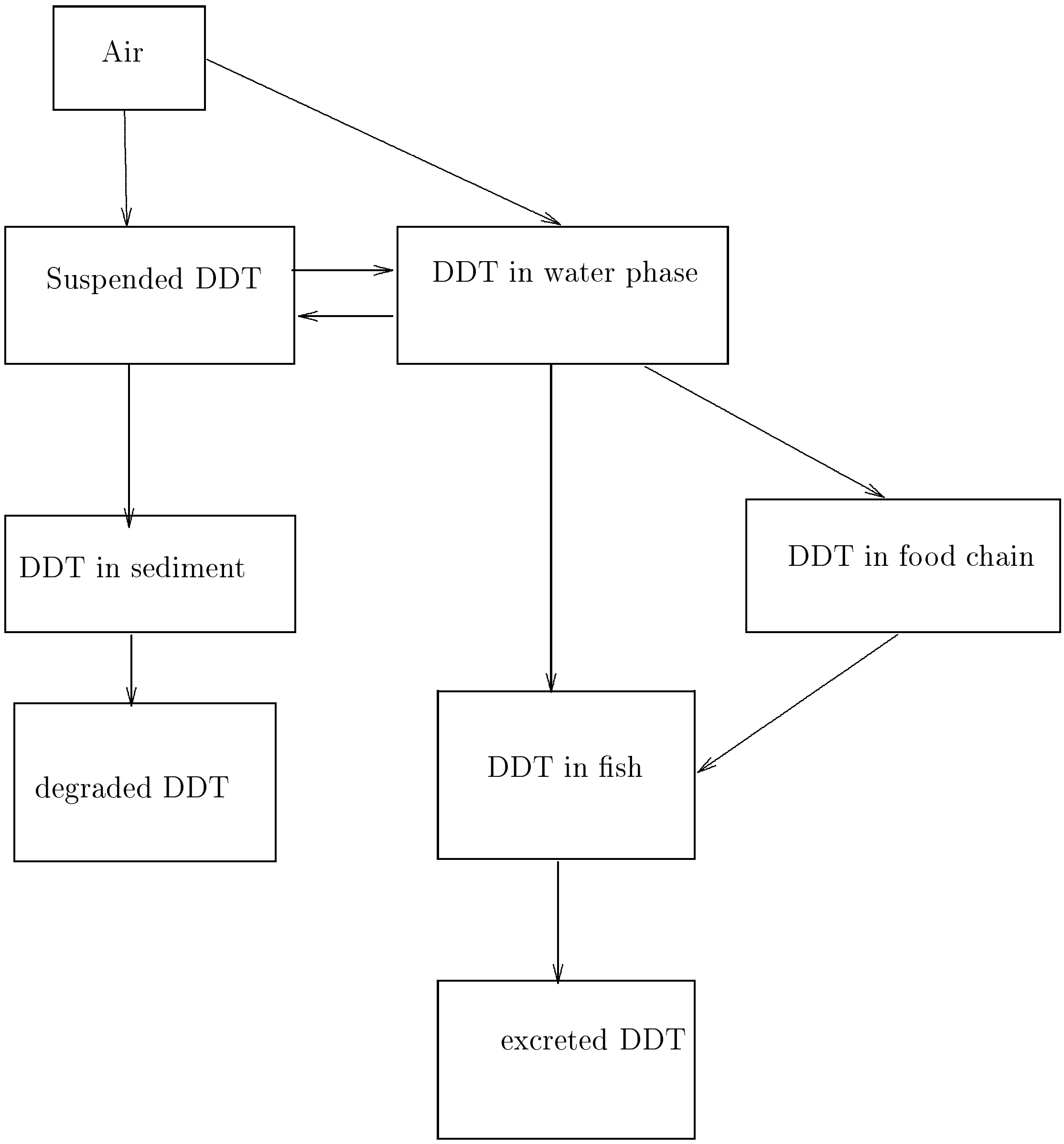

Figure 3.2: Simple DDT model for an aquatic ecosystem.

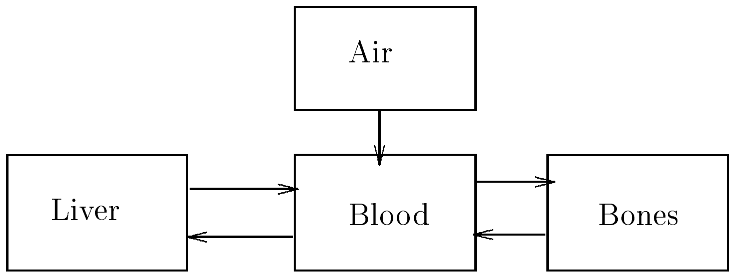

Figure 3.3: Simple model for the distribution of a pollutant in an animal.

simplicity, is, of course, only useful for didactical reasons. In toxicology, models of this type are also called physiologically based pharmacokinetic models (PBPK). The interconnection between ecosystem compartments and chemokinetic compartments can be seen in Example 1. Here, the grass and the cow compartment are, strictly speaking, parts of the ecosystem, while the interplay between the milk compartment and the cow compartment models the pollutant flow inside the body of a (typical) cow.

Example 5 ([19]) Cadmium transport in mammals can be described by a model illustrated in Figure 3.4 (after [19, Section 7.7]). Here, nine compartments are grouped around the central blood compartment, which serves as a distributor of the pollutant load through the body.

Example 6 ([5]) Transport of the degreasing agent trichlorethylene and the monomer 1,1-dichlorethylene (widely used in the packaging industry) in rats has recently been modeled by a compartment model depicted in Figure 3.5 [5]. The compartments in use here are the air compartment, the gas exchange tissue compartment (epithelial cells in the lung tissue as well as skin cells), rapidly perfused tissue (brain, muscles, kidneys, etc.), slowly perfused tissue (bones, guts, etc.), fat tissue, and the liver. Exchanges from the gas exchange tissue to the other in-body compartments is achieved by arterial blood, while the reverse pathway uses venal blood as the carrier.

Note that chemokinetic compartment models model the mass flow of the pollutant alone, leaving the mass flow of proteins, lipids etc. aside. As such, a compartment model is not necessarily inherently simplistic, but it has obvious limits. More detailed models, based on energy balance instead of mass balance exist (cmp, e. g., [12]), but they are inevitably nonlinear in the mass of the pollutant in different body tissues. Moreover, the right-hand side of most of these models is discontinuous in the energy integrator, a feature which makes these models susceptible to numerical difficulties.

Several different types of individuals can also be modeled by specifying the defining compartments for each of them. A typical example would be a

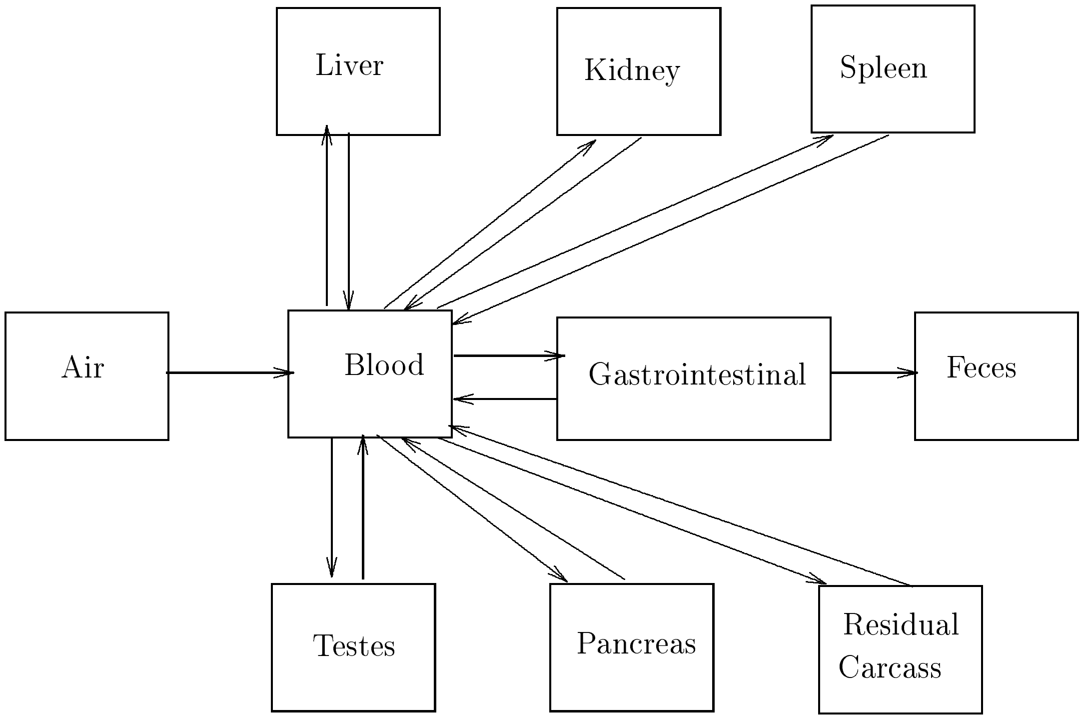

Figure 3.4: Compartment model for transport of cadmium in mammals.

structure as defined in Figure 3.6 (p. 30), in which three types of animals are considered: females, adolescents, and males. Here, the topological structure of each of the different animal classes is the same, since the simple model from Figure 3.3 is simply copied three times. The differences between the individuals considered are therefore hidden in the proportionality constants $\lambda_{i,k}$. However, the actual topology of the subgraphs defined might differ, too. For instance, it might make sense to use a more detailed model for the adolescent compartment group to get more information about how the pollutant affects those members of the population considered which are, for some reasons, probably more affected than others.

3.4 Age-Classes

Different age classes have already been mentioned at the end of the last section and in Figure 3.6. It is clear that any kind of compartment model can be "copied" several times to include multiple members of the same species. Usually, these new members then represent different genders, different age classes, etc. [35]. A multi-individual model spanning a significant part of the generation span of a given species should then also include a mass exchange of pollutant between neighboring age classes (thereby representing an aging

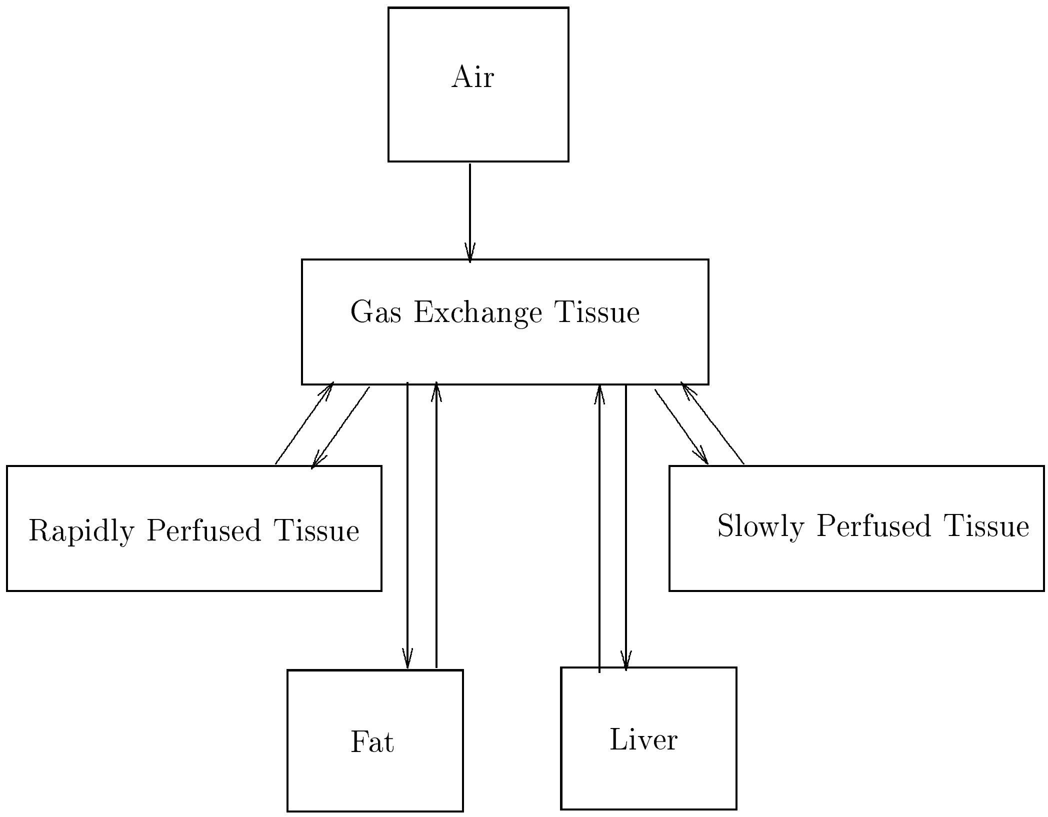

Figure 3.5: Compartment model for transport of trichlorethylene and 1,1-dichlorethylene in rats.

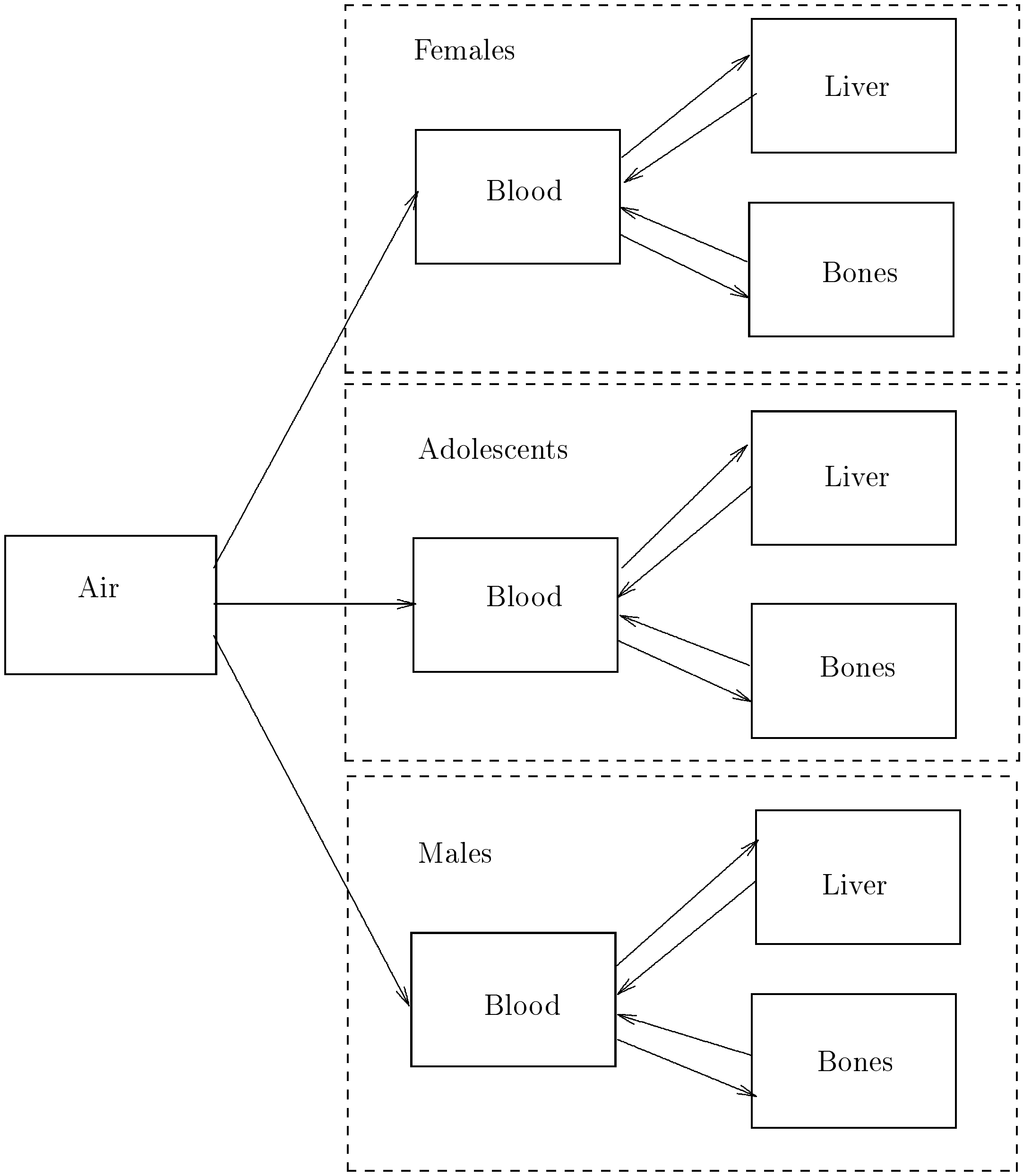

Figure 3.6: Simple model for the distribution of a pollutant in several types of animals.

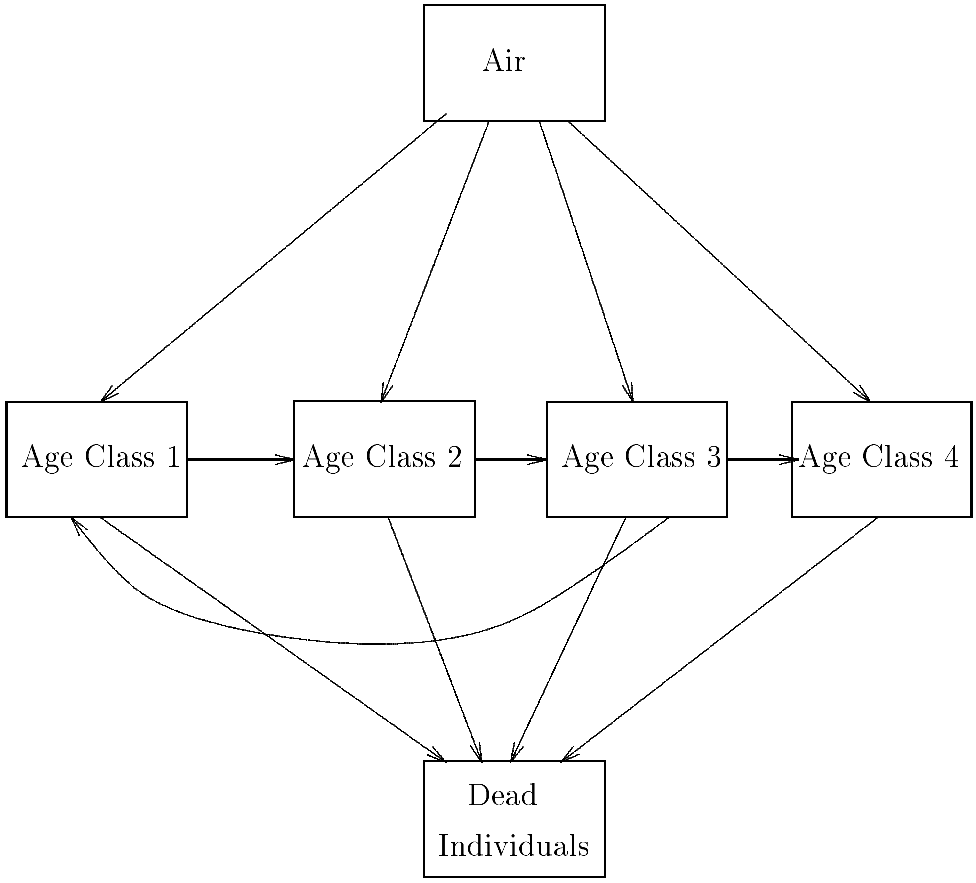

Figure 3.7: Simple age class model with four different age classes.

process of the individuals) and a mass exchange between parental individuals and newly born. A simple conceptual compartment model incorporating this idea is depicted in Figure 3.7. There, the population under consideration is divided into four different age classes. Age class 1 represent newly born and very young members of the population, age class 2 represents adolescents, while age class 3 is the only age class which is able to procreate. A pool for dead individuals has been added, too.

3.5 Finite-Difference Schemes are Compartment Models

It is perhaps surprising that discretization schemes aimed at solving partial differential equations have a direct connection with the model approach depicted in the last sections. However, when the different compartments are visualized as spatially different containers for the pollutant, it becomes clear that the resulting compartment model is nothing else than a spatial discretization of the underlying partial differential equation. In other words, when compartment models are used to discretize the spatial distribution of the pollutant, the method of lines (MOL) is employed to solve the corresponding partial differential equation. This can be seen in the following illustrative example.

Example 7 Consider a one-dimensional time-dependent flow field $u : \mathrm{IR} \times \mathrm{IR} \longrightarrow \mathrm{IR}$, which can serve as a simple model for the water movement in a (one-dimensional) river. For a pollutant concentration $c(x,t) \geq 0$ at the point $x \in \mathrm{IR}$ at the time t the standard mass balance equation

$$\frac{\partial c(x,t)}{\partial t} = -\frac{\partial}{\partial x}(u(x,t)c(x,t)) + \left(\frac{\partial}{\partial x}\sigma\frac{\partial c(x,t)}{\partial x}\right)$$

holds (cmp. the advection-diffusion equation (2.1)). Here, σ is the (longitudinal) dispersion coefficient. We do not consider sources or sinks for the pollutant. A simple finite difference discretization in space around the point $x = x_k$ with step size $h = x_{k+1} - x_k$ leads to

$$\begin{aligned}
\frac{\partial c(x_k,t)}{\partial t} &= -\frac{1}{2h}(u(x_{k+1},t)c(x_{k+1},t) - u(x_{k-1},t)c(x_{k-1},t)) \\
&\quad +\frac{\sigma}{h^2}(c(x_{k+1},t) + c(x_{k-1},t)) \\
&= \left(-\frac{u(x_{k+1},t)}{2h} + \frac{\sigma}{h^2}\right)c(x_{k+1},t) + \left(\frac{u(x_{k-1},t)}{2h} + \frac{\sigma}{h^2}\right)c(x_{k-1},t).
\end{aligned}$$

After multiplying this chain of equations with the volume h of each piece of river, we can interpret $hc(x_k,t)$ as the pollutant mass in a compartment representing a (small) spatial volume of river. Note that when $u \equiv 0$ we have that the flow between the compartments is proportional to the mass inside them, as in the standard model. However, in the general case the flow u represents an active "pump" between the different compartments, possibly destroying this property. Consequentially, the resemblance between finite difference formulations and compartment models ends here. Moreover, the proportionality constants used to quantify the flow between different compartments are time-dependent if and only if the flow u is time-dependent.

The last example can easily be generalized to two- or three-dimensional spatial domains.

Inflow and outflow to and from compartment no. k occur only with respect to the compartments spatially neighboring k, and the system matrix in (3.3) is exactly the standard finite-difference matrix. As a consequence, we regain the well-known method of lines: the space is discretized into a finite number of points or areas or volumes (compartments), while time is treated continuously.

It has therefore to be stressed that the simple considerations with respect to the numerical solution of (3.3) made in Section 3.1 are only valid for a small number of compartments not representing a full spatial discretization of a certain volume. Finite-difference methods are known to be prone to numerical difficulties of several types, and the reader is urged to consult the literature before modeling spatial domains with compartment models of a fine spatial resolution.

Chapter 4

Toxicology

4.1 The Cytodynamic Multistage Model

For each tissue and organ modeled in the chemokinetic model, the cytodynamic model calculates quantitatively the carcinogenic and/or cytotoxic effects of the pollutant in the corresponding tissue or organ. The approach adopted is an extension of the Armitage-Doll model of carcinogenesis and follows the cytodynamic model as outlined, for example, in [7, 3]. Note that this type of model is able to handle hormetic behavior of a tissue to an outside response, a feature of strong interests to toxicologists [2]. The modeler has to specify the number of different cell populations per compartment that should be modeled. The dynamic behavior of each population is then governed by the pollutant concentration inside the compartment (by a toxic or a mutagenic effect) and by the population numbers of the other cell types. The governing ordinary differential equations, however, can not be classified according to some simple criterion. As a consequence, the user must specify the right hand side of the equations explicitly in the source code by modifying the subroutine `CMPCLLR` (file `compartment.f`) accordingly.

The solver routine for the nonlinear system of ordinary differential equations is fully integrated into the routine for solving the chemokinetical equations described in the last chapter.

We describe some simple examples for the type of cytodynamic activity which can be modeled by this approach.

Example 8 Let us consider compartment no. k, representing an organ. The standard Armitage-Doll model of carcinogenesis starts with $r = r(k) \geq 3$ different cell types in an organ (cmp. Figure 4.1). Cell type no. 1 corresponds to healthy cells, cell type no. $r(k)$ corresponds to malignant cells. Cell types no. i ($i = 2, \ldots, r(k) - 1$) correspond to intermediate cells which exist only due to mutations. More precisely, at time $t \geq 0$ a portion of $\mu_{i,k}$ ($0 \leq \mu_{i,k} \leq 1$) cells

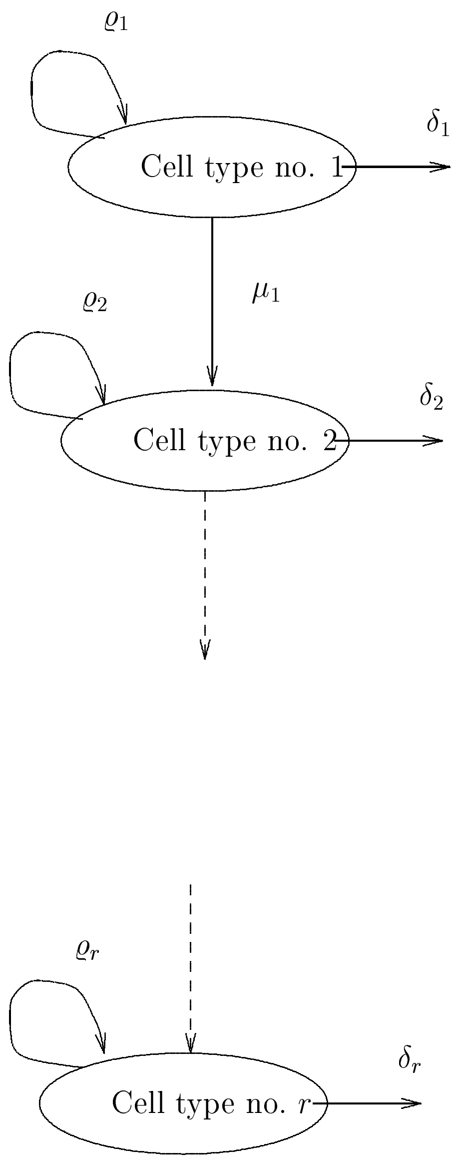

Figure 4.1: Standard Armitage-Doll model of carcinogenesis.

of type i ($i = 1, \ldots, r(k) - 1$) mutate during cell division and are afterwards cells of type $i + 1$. Of course, $\mu_{i,k} = \mu_{i,k}(m_k(t))$, i. e. the mutation rate depends on the concentration of pollutant in the organ. It is also sensible to assume that $\mu_{i,k}(\cdot)$ is monotone nondecreasing. Moreover, each cell type has a death rate $\delta = \delta_{i,k}(m_k(t)) \in [0, 1]$, again monotically nondecreasing in the pollutant concentration, and a reproduction rate $\rho = \rho_{i,k}(m_k(t)) \in [0, 1]$. We assume that the mutation rate is already accounted for in the values δ and ρ, i. e. $\delta_{i,k}(m_k(t))$ represents the total death rate while $\rho_{i,k}(m_k(t))$ represents the net reproduction rate. Let $p_{i,k}(t)$ be the population size (i. e. number of individuals) of type no. i at time $t \geq 0$. The complete system of ordinary differential equations is then a linear homogeneous non-autonomous system of size $r(k)$ with upper tridiagonal matrix:

$$\begin{pmatrix} p'_{1,k}(t) \\ p'_{2,k}(t) \\ \vdots \\ p'_{r-1,k}(t) \\ p'_{r,k}(t) \end{pmatrix} = \begin{pmatrix} a_{1,1}(t) & a_{1,2}(t) & & & \mathbf{0} \\ & a_{2,2}(t) & a_{2,3}(t) & & \\ & & \ddots & \ddots & \\ & & & a_{r-1,r-1}(t) & a_{r-1,r}(t) \\ \mathbf{0} & & & & a_{r,r}(t) \end{pmatrix} \begin{pmatrix} p_{1,k}(t) \\ p_{2,k}(t) \\ \vdots \\ p_{r-1,k}(t) \\ p_{r,k}(t) \end{pmatrix},$$

$t \geq 0$, $(p_{1,k}(0), \ldots, p_{r(k),k}(0))^\top = (p_{1,k,0}, \ldots, p_{r(k),k,0})^\top \geq 0$ given, $r := r(k)$ and

$$a_{i,i} := a_{i,i}^{(k)}(t) = -\delta_{i,k}(m_k(t)) + \rho_{i,k}(m_k(t))$$

($i = 1, \ldots, r(k)$) as well as

$$a_{i,i+1} := a_{i,i+1}^{(k)}(t) = \mu_{i+1,k}(m_k(t))$$

($i = 1, \ldots, r(k) - 1$).

Example 9 ([3]) A more realistic model, fully capable of simulating hormetic effects, is described in [3], see Figure 6.8 (p. 59). Here, stem cells can mutate to intermediate cells, which can then mutate into malignant cells. But stem cells are replenished by reservoir cells, which can mutate into a different class of intermediate cells. This second class of intermediate cells is then able to mutate to malignant cells. Adding a pool of reproductively dead cells, the model uses seven different cell types.

Example 10 A homeostatic model of cytotoxicology found in several textbooks is the following. There are two different types of cells in compartment no. k, reproducing cells with population size $p_{1,k}(t)$ at time $t \geq 0$ and reproductively dead cells with population size $p_{2,k}(t)$ at time $t \geq 0$. The latter are usually the actual 'working' cells in an organ, while the cells of the first type are the only ones considered who divide. The death rates of the populations are denoted by $\delta_{i,k}(m_k(t))$ $(i = 1, 2)$, while the cell division rate of population no. 1 is given by a function $\lambda_k(p_{2,k}(t))$, where $\lambda_k : [0, +\infty[\longrightarrow [0, 1]$ is a monotonically nonincreasing function. As a consequence, a larger part of the reproducing cell population divides the smaller the population of working cells is, i. e. the more the organ under consideration is harmed. Since cells divide into two cells, the number of newly created cells at time $t \geq 0$ is $2\lambda_k(p_{2,k}(t))p_{1,k}(t)$. A fraction $\alpha_k \in [0, 1]$ of these newly created cells will differentiate into working cells, while the fraction $1 - \alpha_k$ will differentiate into reproductive cells. Here $\alpha_k = \alpha_k(p_{2,k}(t))$ and $\alpha_k(\cdot)$ is a monotonically nondecreasing function. The dynamics of this system are then given by

$$
\begin{aligned}
p'_{1,k}(t) &= ((1 - 2\alpha_k(p_{2,k}(t))\lambda_k(p_{2,k}(t))) - \delta_{1,k}(m_k(t)))\, p_{1,k}(t), \\
p'_{2,k}(t) &= 2\alpha_k(p_{2,k}(t))\lambda_k(p_{2,k}(t))p_{1,k}(t) - \delta_{2,k}(m_k(t))p_{2,k}(t),
\end{aligned}
$$

$t \geq 0$, $(p_{1,k}(0), p_{2,k}(0))^\top = (p_{1,k,0}, p_{2,k,0})^\top \geq 0$ given. It can be shown that this system has a stationary point in the case $m_k = \text{const}$ and that this point is stable if the functions α_k, λ_k, and $\delta_{i,k}$ fulfill some simple conditions. More general models can easily incorporate a possible dependence of all of these functions on $m_k(t)$.

Chapter 5

Optimization

5.1 The Objective Function

5.1.1 The Effect Function

Let $G \subset \mathbb{R}^2$ be the ground surface of the modeled region. (We stress again that we are using a Cartesian coordinate system.) The discussion in the last chapters and the model derivation in Chapter 1 have shown that we are able to calculate not only the pollutant concentration in certain compartments, which can be anything from ecotrophic levels down to organs in individuals, but that we can also calculate the effect of a pollutant on certain cell populations in terms of toxicity and carcinogenicity. What remains is to find an agglomeration of the different effects on the different levels into one value. The minimization of *all* the quantified effects simultaneously is evidently a multicriteria problem for which the knowledge of some approximation of the efficient set would be of great use. However, the high computational demands to compute such an approximation as well as methodological difficulties force us at the present moment to globalize all the different effects by way of a utility function. The choice of such a utility is nontrivial, and it does not seem to be clear what effect a corresponding choice has on the solutions of the resulting single-criterion optimization problem. We prefer at the present moment to globalize all the different effects by way of a weighting function. It is clear that some of the efficient points of the abovementioned multicriteria problem will be missed in this way [9]. However, scalarizations with, e. g., the weighted max-norm, capable of computing all efficient points [16], lead to nondifferentiable problems or problems with a possibly large number of constraints, thereby adding computational costs which might well be able to prohibite the computation of a solution.

Let $n \geq 1$ be the number of compartments and $m_k(x,t)$ be the pollutant mass in compartment k $(k = 1, \ldots, n)$ at the point $x \in G$ and time

$t \geq 0$. Furthermore, let $r(k) \geq 0$ be the number of cell populations in compartment k and $p_{i,k}(x,t)$ be the number of cells of type i in compartment k ($i = 1, \ldots, r(k)$) at the point $x \in G$ and time $t \geq 0$. Set $N := \sum_{k=1}^{n} r(k)$. The modeler has then to specify $n + N$ real-valued weights $\hat{\omega}_k, \omega_{i,k}$ ($k = 1, \ldots, n$, $i = 1, \ldots, r(k)$) (one for each compartment and one for each cell population). Let $\lambda = (y_1, y_2, w, h)^\top \in \mathrm{IR}^4$ be the vector of decision variables. The vector $(y_1, y_2) \in G$ represents the location of the pollutant emitter, while $w \geq 0$ is the width of the "smokestack" and $h \geq 0$ is it's height. With the weights given as above, define the function

$$W(x,t,\lambda) := \sum_{k=1}^{n} \left(\hat{\omega}_k m_k(x,t) + \sum_{i=1}^{r(k)} \omega_{i,k} p_{i,k}(x,t) \right),$$

which measures the agglomerated effect of the decision λ at the point $x \in G$ at time $t \geq 0$. Here, the dependance of m_k and $p_{i,k}$ on λ has been skipped, as it has been done above. The agglomeration of effects over a time interval $[0,T]$, $T > 0$, can be done in many ways. The method employed in OLAF is the time integral over the effect

$$e(x,\lambda) := \int_0^T W(x,t,\lambda)\,\mathrm{d}t. \tag{5.1}$$

This approach has the advantage that

$$e(x,\lambda) = \sum_{k=1}^{n} \left(\hat{\omega}_k \int_0^T m_k(x,t)\,\mathrm{d}t + \sum_{i=1}^{r(k)} \omega_{i,k} \int_0^T p_{i,k}(x,t)\,\mathrm{d}t \right).$$

The time variable t is already discretized by the air pollution model, and it is this equidistant discretization which is used in the numerical evaluation of the integral above. Note that this approach allows for hormetic values, i. e. $e(x,\lambda) < 0$ for some points $x \in G$ and decisions $\lambda \in \mathrm{IR}^4$. In such a case, the pollutant effect has actually lead to a reaction of the individual's cell population residing at x which can be interpreted as an increase in health [2, 3].

Alternatives

Other important and convenient agglomeration functions over time are weighted p-norms, i. e,

$$e(x,\lambda) := \left(\int_0^T |\omega(t) W(x,t,\lambda)|^p \mathrm{d}t \right)^{1/p}$$

($1 \leq p < +\infty$) or

$$e(x,\lambda) = \sup_{t \in [0,T]} |\omega(t) W(x,t,\lambda)|,$$

where $\omega : [0, T] \longrightarrow \mathrm{IR}$ is a weighting function with $\omega(t) > 0$ almost everywhere. This can easily be generalized to semi-norms in which $\omega(t) = 0$ may hold on a subset of $[0, T]$ with nonzero measure, but $\omega(t) > 0$ else. Weighting functions with this property can also be introduced in (5.1). When interpreting e as a sampling function, this means that some points (in time) are left unsampled. The corresponding agglomeration functions include

$$e(x, \lambda) = W(x, T, \lambda),$$

i. e. only the effect at the end of the time interval considered is accounted for.

Since there are too many different time agglomeration functions which are reasonable in different circumstances, no attempt has been made to provide the user of the software package with a number of templates. If agglomerations different from the one presently implemented are preferred, the source code of the optimization module has to be changed accordingly.

5.1.2 The Objective Function

Let $e(x, \lambda)$ be an agglomeration function for the pollutant effects on individuals residing at $x \in G$. If the cytodynamic model is used, this effect quantification measures the overall effect of the pollutant under consideration on one individual only. From the viewpoint of a health care decision maker, it is therefore important to incorporate the pollutant effect on the whole population. Fortunately, this poses no problem. Suppose that we have given a population density function $p : G \longrightarrow \mathrm{IR}$, $p(x) \geq 0$ for all $x \in G$. The objective function to be minimized is then given by

$$f(\lambda) := \int_G p(x) e(x, \lambda) \, \mathrm{d}x. \tag{5.2}$$

Obviously, to evaluate f we do not only need to evaluate $e(\cdot, \lambda)$ at several points, but we also need a numerical integration routine. Since in the present version of OLAF the pollutant concentration and the health effect function is computed on a fixed Cartesian grid of size $N \times M$ with grid points $x_i \subset G$, the integral in (5.2) can be replaced by the weighted sum over all grid points considered, where the weights are $1/NM$.

5.1.3 The Gradient of the Objective Function

An efficient optimization algorithm needs necessarily local information of higher order about the objective function. While there does not seem to be a simple way to compute $\partial f / \partial w$ and $\partial f / \partial h$, the gradient with respect to the locational decision variables y_1, y_2 (which are arguably the most important ones) can be

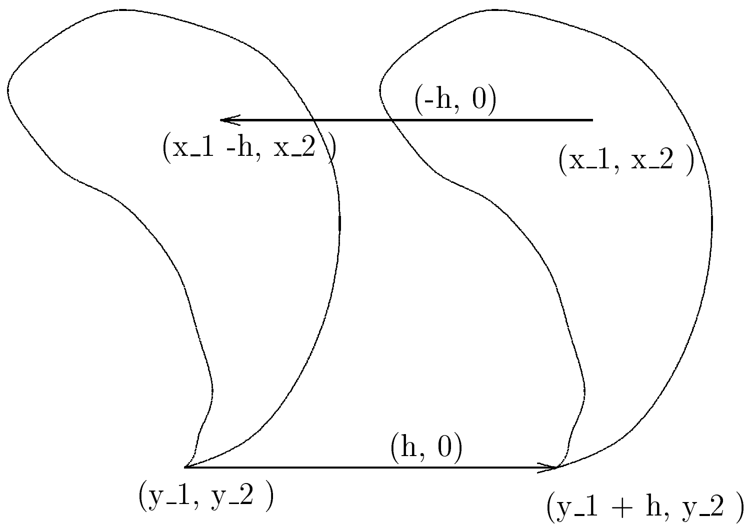

Figure 5.1: Translating the origin $(y_1, y_2)\top$ of the pollutant source results in a translation of the pollutant cloud and of the effect.

approximated in the following way. Under suitable differentiability assumptions and integrability of $e(\cdot, \lambda)$, we have that

$$\frac{\partial f(\lambda)}{\partial y_1} = \frac{\partial}{\partial y_1} \int_G p(x) e(x, \lambda)\, \mathrm{d}x = \int_G p(x) \frac{\partial}{\partial y_1} e(x, \lambda)\, \mathrm{d}x. \tag{5.3}$$

Taking a look at Figure 5.1, it is reasonable to make the assumption

$$e((x_1, x_2)^\top, (y_1 + \delta, y_2, w, h)^\top) = e((x_1 - \delta, x_2)^\top, (y_1, y_2, w, h)^\top) + o(|\delta|)$$

for $\delta \to 0$. As a consequence, for $e(\cdot, \lambda) \in C^1(\mathrm{int}(G))$ we get

$$\begin{aligned}
&\frac{\partial}{\partial y_1} e((x_1, x_2)^\top, (y_1, y_2, w, h)^\top) \\
&= \lim_{\delta \to 0} \frac{1}{2\delta} \Big(e((x_1, x_2)^\top, (y_1 + \delta, y_2, w, h)^\top) - e((x_1, x_2)^\top, (y_1 - \delta, y_2, w, h)^\top) \Big) \\
&= \lim_{\delta \to 0} \frac{1}{2\delta} \Big(e((x_1 - \delta, x_2)^\top, (y_1, y_2, w, h)^\top) - e((x_1 + \delta, x_2)^\top, (y_1, y_2, w, h)^\top) \\
&\qquad + o(|\delta|) \Big) \\
&= -\frac{\partial}{\partial x_1} e((x_1, x_2)^\top, (y_1, y_2, w, h)^\top).
\end{aligned}$$

This is the approach chosen in OLAF. The differentials $\partial e / \partial x_1$ are approximated by finite differences of third order, where the discretization size is exactly the grid size of the computational grid in x_1-direction. In this way, the approximations to $e(\cdot,\lambda)$ on the grid discretizing G are used to calculate the differential of $e(\cdot,\lambda)$, and no further simulation run of the dispersion model or the cytodynamic model is necessary. Under analogous assumptions, the same reasoning holds for $\partial f/\partial y_2$. Additional objective function evaluations due to time-intensive numerical differentiation routines can therefore be avoided. Computational tests showed that the use of this scheme results in a large decrease in computation time, although the total number of iterations increases slightly when compared to an implementation where central differences have been used to approximate the gradient of the objective function. This might be attributed to the fact that in practical applications the accuracy of the approximation scheme above is lower than central differences. Consequently, descent algorithms which employ local information of higher order need more iterations, but less total computation time.

We may go even one step further. Using (5.3) and partial integration, we see that

$$\begin{aligned}\frac{\partial f(\lambda)}{\partial y_1} &= -\int_G p(x)\frac{\partial}{\partial x_1}e(x,\lambda)\,\mathrm{d}x \\ &= -\int_{\partial G} p(z)e(z,\lambda)\nu(z)\,\mathrm{d}z + \int_G e(x,\lambda)\frac{\partial}{\partial x_1}p(x)\,\mathrm{d}y\end{aligned}$$

holds whenever G has a smooth boundary and the function $p(\cdot)e(\cdot,\lambda)$ has compact support. Here, $\nu(y)$ is the outer normal of G at a boundary point $y \in \partial G$. By choosing G large enough (or choosing a population distribution p with compact support small enough), we can make sure that the boundary integral gets arbitrary small, and we can approximate the differential of f with respect to y_1 by

$$\frac{\partial f(\lambda)}{\partial y_1} \approx \int_G e(x,\lambda)\frac{\partial}{\partial x_1}p(x)\,\mathrm{d}y. \tag{5.4}$$

Since $\partial p(x)/\partial x_1$ is independent of λ, it can be computed or approximated in whatever way necessary before any optimization takes place. Approximating $\partial f(\lambda)/\partial y_1$ then involves only the evaluation of the integral in (5.4), which can be done in the same way as the evaluation of f at λ.

Unfortunately, none of these schemes works for the derivatives with respect to stack height and stack diameter. As a consequence, numerical differentiation with central differences is used to approximate the derivative of f with respect to h and w.

A different possibility would be to use automatic differentiation to compute derivatives of f with respect to h and w. Preliminary numerical experience [25, Chapter 1] suggests that this approach is computationally very expensive for parametrized air dispersion models. No effort has been made to use automatic differentiation within OLAF.

5.2 Optimization

5.2.1 Constraints

The optimization routine considers the locational parameters of the polluting facility as parameters to be optimized and minimizes the function f as defined above. As outlined in Chapter 1, the decision variables for the optimization process are: the location of the facility, the stack height of the pollutant emitter, and the stack diameter of the pollutant emitter. Usually, constraints on $\lambda \in \mathbb{R}^4$ are natural results of the modeling process. They include representations of engineering restrictions on stack height resp. diameter as well as locational constraints representing coastlines, jurisdictional boundaries, etc. As a consequence, almost all optimization problems to be solved are constrained ones. We can, however, assume that all constraints are written in standard form

$$g_{\leq}(\lambda) \leq 0$$

or

$$g_{=}(\lambda) = 0$$

with functions $g_{\leq} : \mathbb{R}^4 \longrightarrow \mathbb{R}^{n_1}$, $g_{=} : \mathbb{R}^4 \longrightarrow \mathbb{R}^{n_2}$, $n_1, n_2 \geq 1$. Moreover, both functions can be assumed to be continuously differentiable. This is usually considered as a strong assumption in locational analysis, especially if one wants to model regions forbidden for placement of the smokestack. However, recent research [8] has shown that a corresponding problem formulation is not necessarily more complicated than other locational models, provided that the right modeling tools are used.

5.2.2 The Algorithm

At the present moment, an SQP-method [34] is used to solve the location problem. Since in this SQP-method the Hessian of the Lagrangian of the optimization problem is updated by way of a modified Pantoja-Mayne update formula, only first-order information on the objective function and the constraint functions is needed. This means that, in principle, the use of approximations of order 3 to $\partial e / \partial x$, as outlined in Section 5.1.3, is unnecessarily accurate. However, other optimization routines might benefit from this accuracy, and the bottleneck of the computation does not lie in the gradient approximation anyway. Of course, it is trivial to replace the optimization routine in use with a different one.

Note that this optimization method is able to find local optimizers of the objective function, but cannot decide if a global optimizer is found. Any user of the software package is urged to rerun the code from several different starting points in order to make sure that the probability of finding the global optimum is increased. This multistart technique mimics, albeit roughly, the application of a clustering technique, one of the most well known and most successful global optimization approaches [23].

Chapter 6

A Numerical Study

The software package OLAF was used for a numerical study concerning the location of an air polluting facility in the region of central Europe.

"Central Europe" is a rather vague term. We mean here a region roughly encompassing the Benelux countries (Belgium, The Netherlands, and Luxembourg), the western part of Germany, southeastern Britain, and northeastern France. More precisely, the region used in this case study is a spherical rectangle with the Greenwich meridian (0° longitude) as it's eastern bound and 8°E longitude as it's western boundary, while the latitude coordinate runs from 47°N to 53°N. This region has been chosen given the availability of data needed by us, see Section 6.1 and especially 6.1.3. In Figure 6.1 and 6.2 (p. 46 and 47) the region under consideration is depicted (maps have been produced with GMT (*G*eneric *M*apping *T*ools), see [43]).

6.1 Data

6.1.1 Population Data

The origin of the population data is described in [36]. This data has a resolution of $0.083° \times 0.083°$ (about 54.7km^2) i. e. the earth is divided into a number of spherical rectangles of size as above and the number of inhabitants in some of them is given, and represents the human population of the earth for 1994. Note that it will be hard to use realistic population data with much better resolution, since the actual point of living of the humans is not of great interest to us. In western Europe, north America and similar regions of the earth, the location of living of a human can differ significantly from the location where the individual spends a large amount of his time. It has been estimated that the average daily activity radius for a human being living in western Europe is ca. 15km [36]. Although the actual radius is cultural, environmental, social, and urban/rural dependent, we see no reason to use a higher spatial resolution

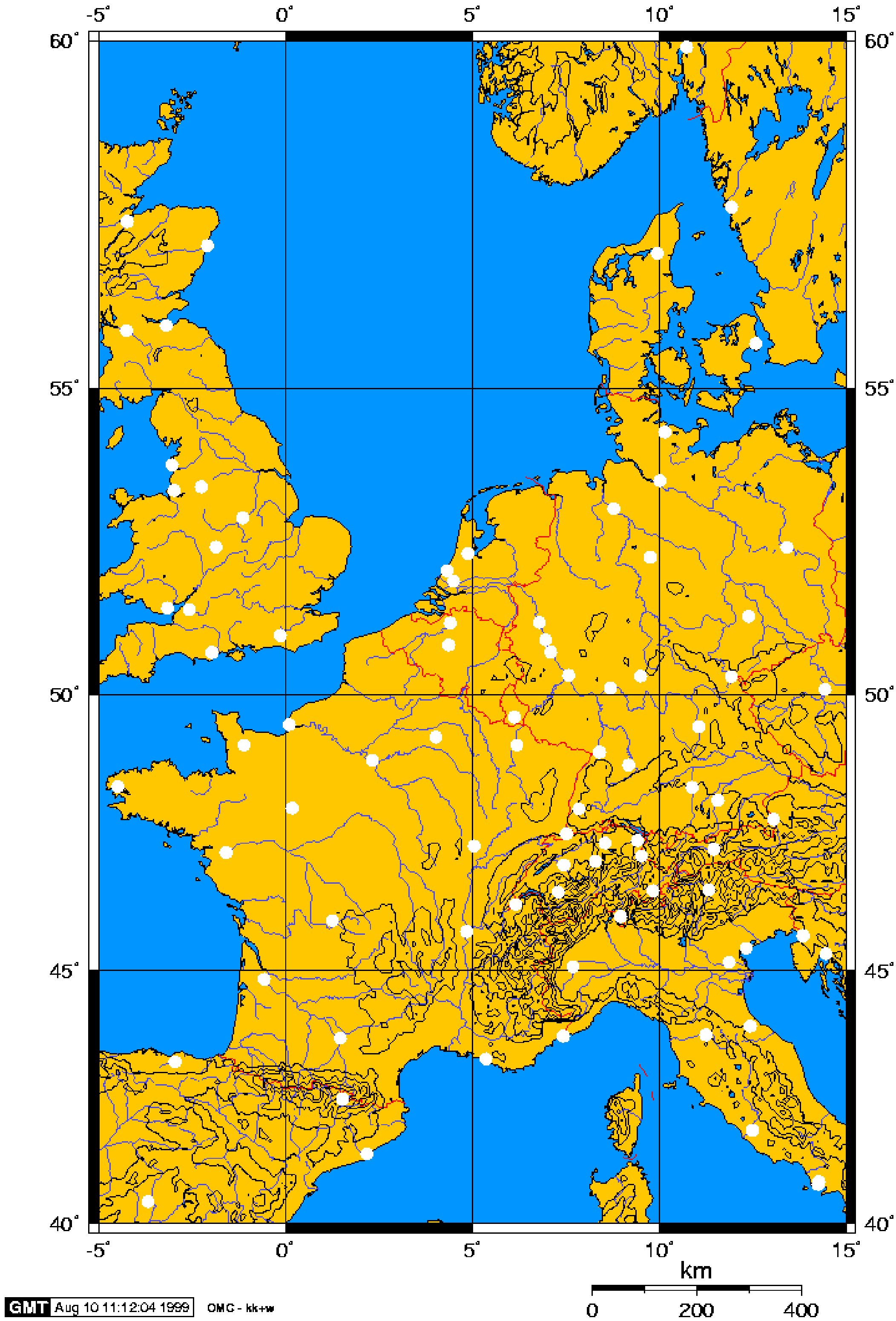

Figure 6.1: An overview of the modeled region and it's surroundings.

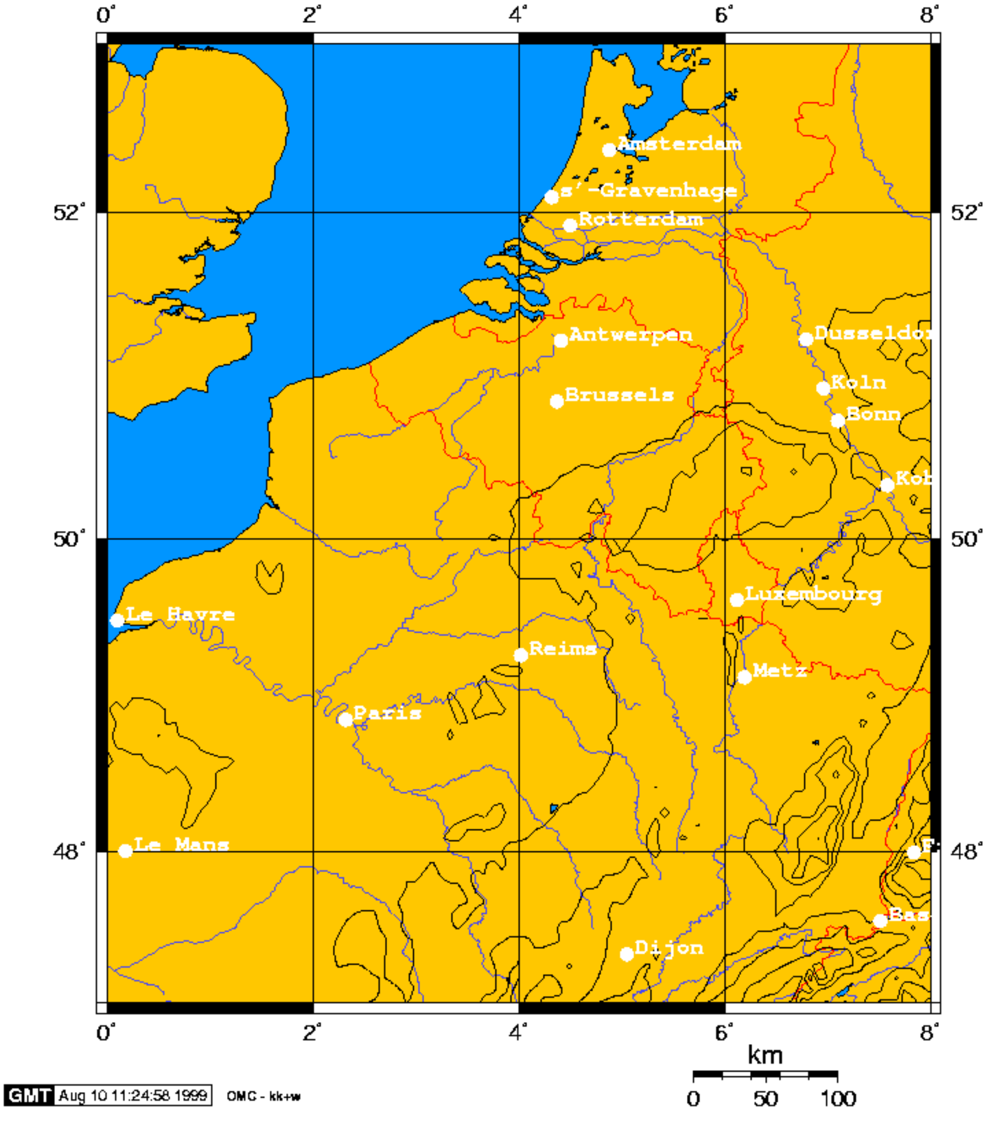

Figure 6.2: The modeled region.

of the population distribution pattern. No attempt has been made to extrapolate the data given to the time period in which the simulation takes place (see Section 6.1.3). While this might be an important issue in developing countries to account for migrational patterns, birth and death rates, and outdated data, it was felt that this does not compromise data accuracy here.

Since the population data used by OLAF is discretized on the meteorological grid, the original data has been bilinearly interpolated to generate the corresponding values. Figure 6.3 and 6.4 (p. 49, 50) show the resulting discrete distribution function interpolated by bicubic splines.

In total, roughly 58 million people live in the region which the case study considers.

6.1.2 Land Use Data

Land use data has been obtained from [31], where the surface of the earth is divided into 44 different land ecosystem complexes of size $0.5° \times 0.5°$ each. Although the resolution needed was much higher than the resolution given, no attempt has been made to add more detailed land use data by way of manual editing, use of digitized cartographic data, etc. A filter was used to translate the ecosystem complexes given into the twelve different land use categories needed by the meteorological preprocessor (see Table 6.1, p. 51). Again, no attempt has been made to extrapolate the data to the time period of interest.

6.1.3 Meteorological Data

A realistic study of the health effects of an airborne pollutant has to use meteorological data of at least one continuous year in a given spatial domain. Longer time periods may be averaged to one year or kept as they are, but the prevalent weather phenomena in Europe do not allow to draw detailed conclusions on computations based on smaller time periods. However, according to a statement of the KMI (Koninklijk Meteorologisch Instituut van België) [26], a data set consisting of the needed observational variables and encompassing a period of one year can be made available only for 341 610 Belgian francs (ca. 8469 euro), and this amount is independent of the fact that the data is needed for a scientific noncommercial project.

Meteorological data has instead been obtained from the FASTEX archive [17] for free. The data consists of, among others, all the information necessary for the MESOPAC module (cmp. Subsection 2.1) and covers a time period from January 6, 1997, 0:00 GMT, to February 28, 1997, 18:00 GMT. While this time period is shorter than actually necessary, we feel confident that we can at least demonstrate the feasibility of the approach taken by solving an optimization

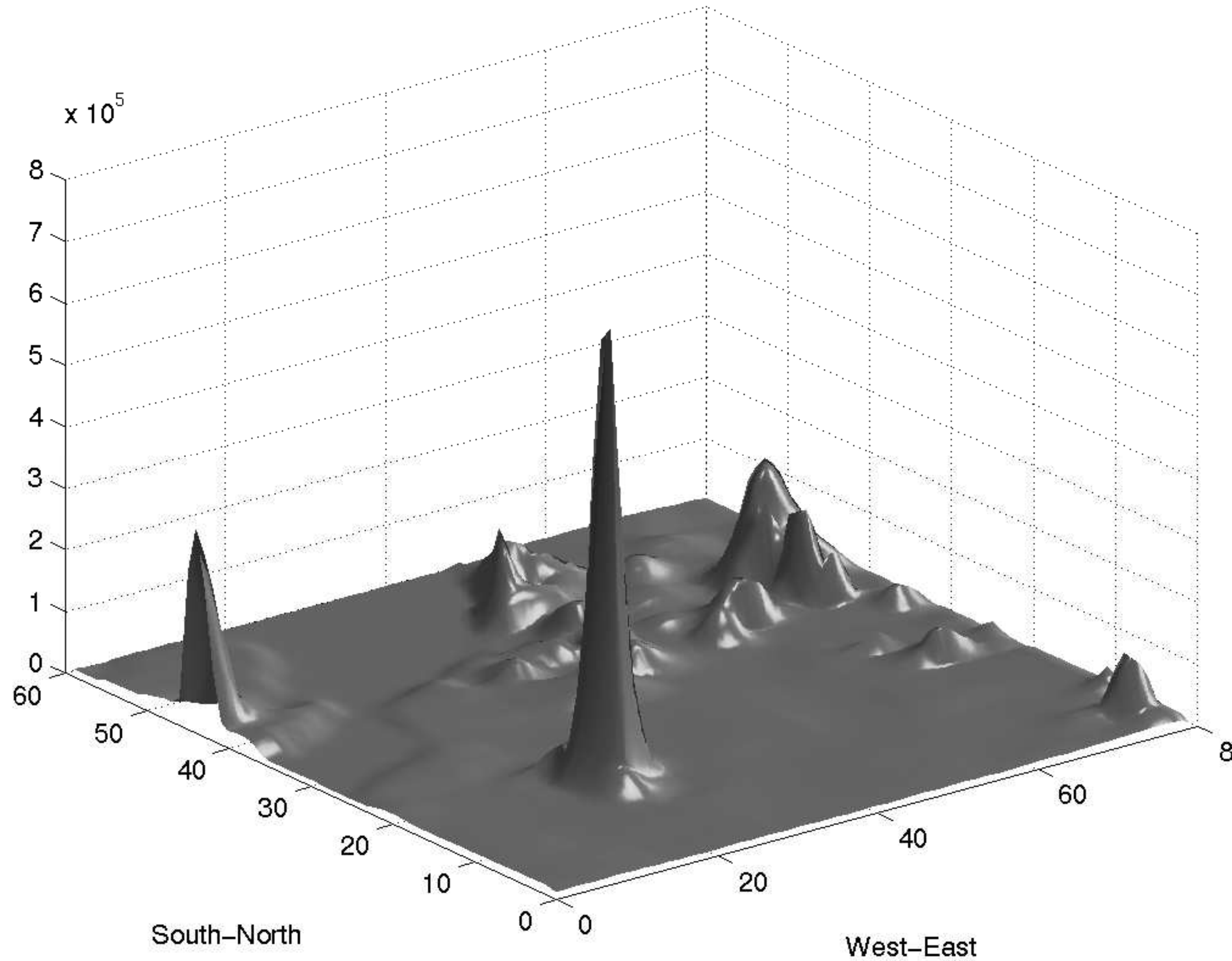

Figure 6.3: Population distribution on the computational grid as seen from the southwest. The highest peak represents Paris, the "hills" near the far away corner are the cities in the Rhine-Ruhr area. On the left side, greater London area is just cut off.

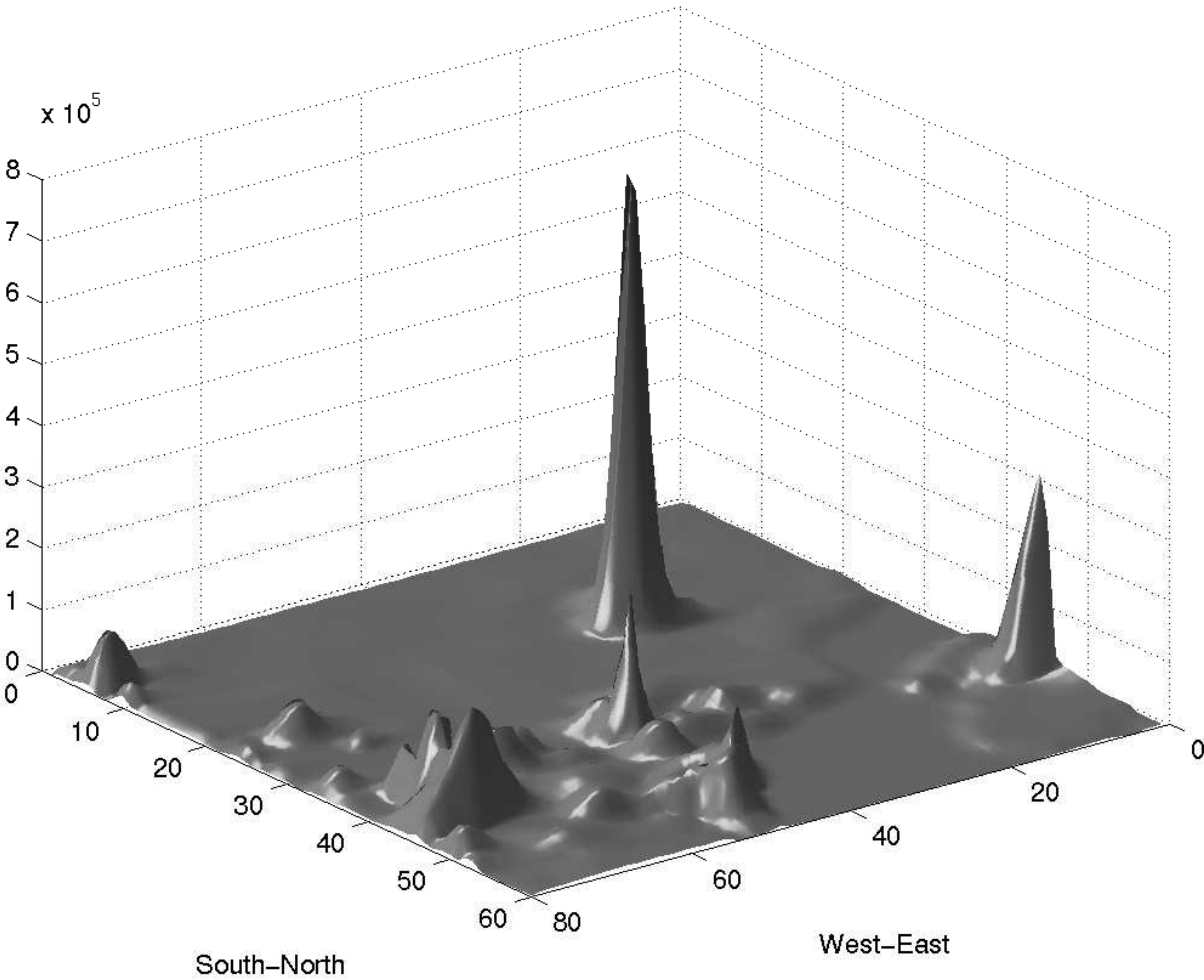

Figure 6.4: Population distribution on the computational grid as seen from the north east. Compare also Figure 6.3. The small peak just in front of Paris is Brussels.

ecosystem code	ecosystem description	land use code	ecosystem code	ecosystem description	land use code
0	water	12	46	med. grazing	2
17	antarctica		47	highland shrub	8
20	main taiga	5	48	semiarid woods	7
21	main taiga	5	49	low scrub	8
22	cool conifer	5	50	sand desert	
23	cool mixed	5	51	hot desert	
24	warm deciduous	5	52	cool desert	
25	warm mixed	5	53	tundra	
26	warm mixed	5	54	tundra	
27	warm conifer	5	55	cool field/woods	2
28	trop. montane	5	56	warm for./field	2
29	trop. seasonal	4	57	cool for./field	2
30	cool crops	1	58	warm field woods	2
31	warm crops	1	59	succulent thorns	
32	tropical dry for	4	60	south. taiga	
33	eq. evergreen	5	61	e. south. taiga	
36	paddylands		62	north. taiga	
37	warm irrigated	3	63	wooded tundra	4
38	cool irrigated	3	64	heaths, moors	9
39	cold irrigated	3	65	coastal edges	
40	cool grass/shrub	6	66	coastal edges	
41	warm grass/shrub	6	67	coastal edges	
42	siberian parks		68	coastal edges	
43	trop. savanna	8	69	polar desert	
44	bogs, bog woods	10	70	ice	
45	marsh, swamp	10	71	hot desert	

Table 6.1: Ecosystem categories and their mapping to land use categories. Missing values have not been translated.

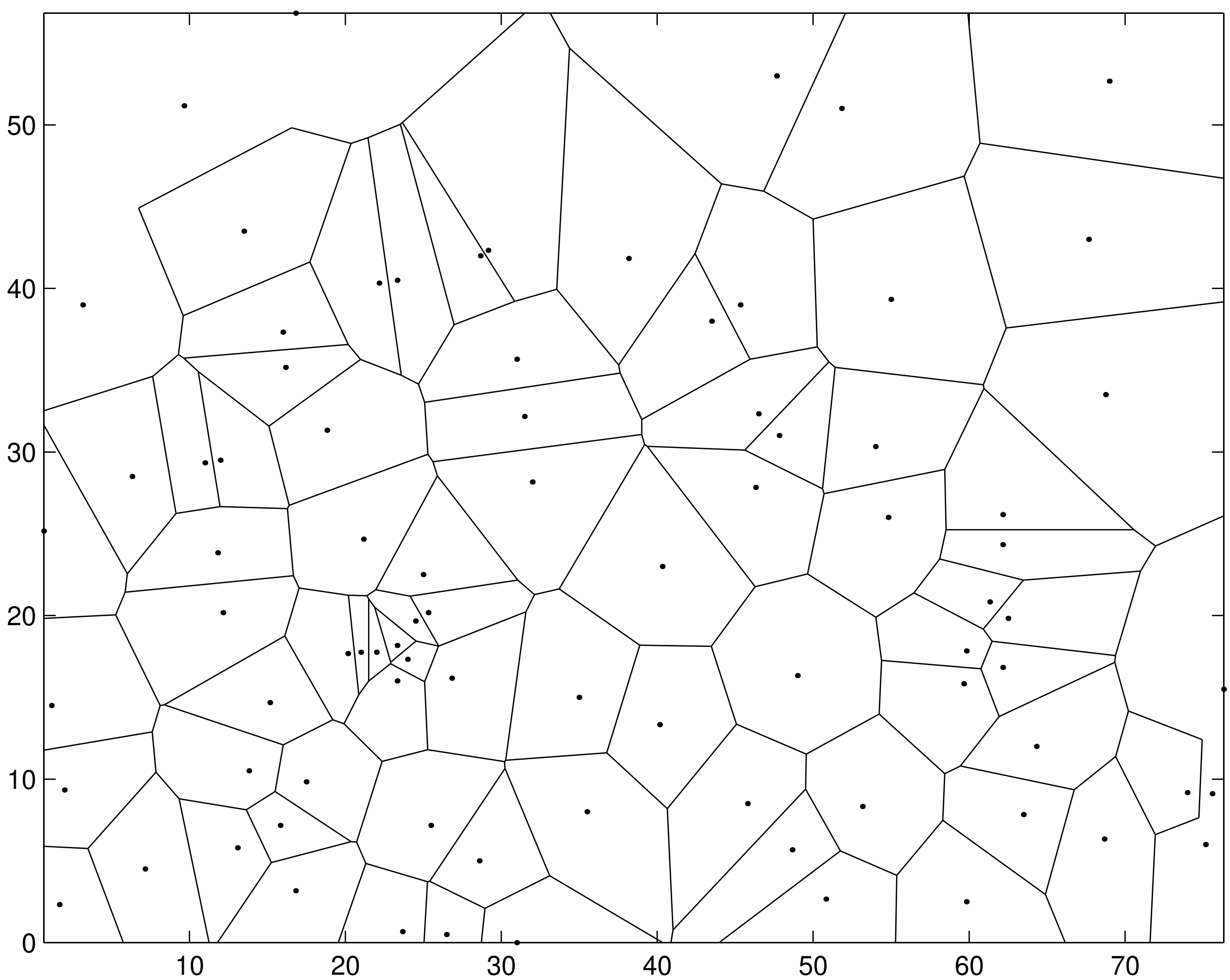

Figure 6.5: Voronoi diagram of 84 surface air stations used.

problem as realistic as possible. A filter routine has been used to transform the data from standard WMO (World Meteorological Organization) format to CD-144 format. The 'present weather' code of field WW (WMO Table 020003) has been recast to CD-144 precipitation format according to Table 6.2. Missing data has been interpolated by a simple one-dimensional linear interpolation routine (in time). No attempt has been made to use other meteorological data in the interpolation process, although it has to be expected that the quality of the complete data set can be greatly improved in this way. A Voronoi diagram of the surface stations whose data has been used can be found in Figure 6.5, while the corresponding diagram for the upper air stations can be found in Figure 6.6. Note that not all upper air stations are within the boundaries of the computational domain, and that the data set in use is rather sparse.

WMO code	CD-144 prec. field	CD-144 prec. code	WMO code	CD-144 prec. field	CD-144 prec. code
14	2	4	72	2	2
15	2	4	73	2	2
16	2	4	74	2	3
20	2	5	75	2	3
21	1	5	76	3	8
22	3	1	77	4	8
23	4	1	78	3	8
24	2	7	79	5	2
25	1	4	80	1	4
26	4	1	81	1	5
27	5	5	82	1	6
50	2	4	83	1	7
51	2	4	84	1	8
52	2	5	85	4	1
53	2	5	86	4	2
54	2	6	87	3	4
55	2	6	88	2	4
56	2	7	89	5	5
57	2	8	90	5	5
58	1	1	91	1	1
59	1	1	92	1	2
60	1	1	93	3	1
61	1	1	94	3	2
62	1	2	95	1	2
63	1	2	96	5	5
64	1	3	97	1	3
65	1	3	99	5	5
66	1	1	111	3	8
67	1	2	121	1	1
68	1	1	122	2	4
69	1	2	123	1	2
70	3	1	124	3	2
71	2	1	125	1	8
291	5	5			

Table 6.2: Transformation of WMO present weather code to CD-144 precipitation code. WMO codes missing have not been translated.

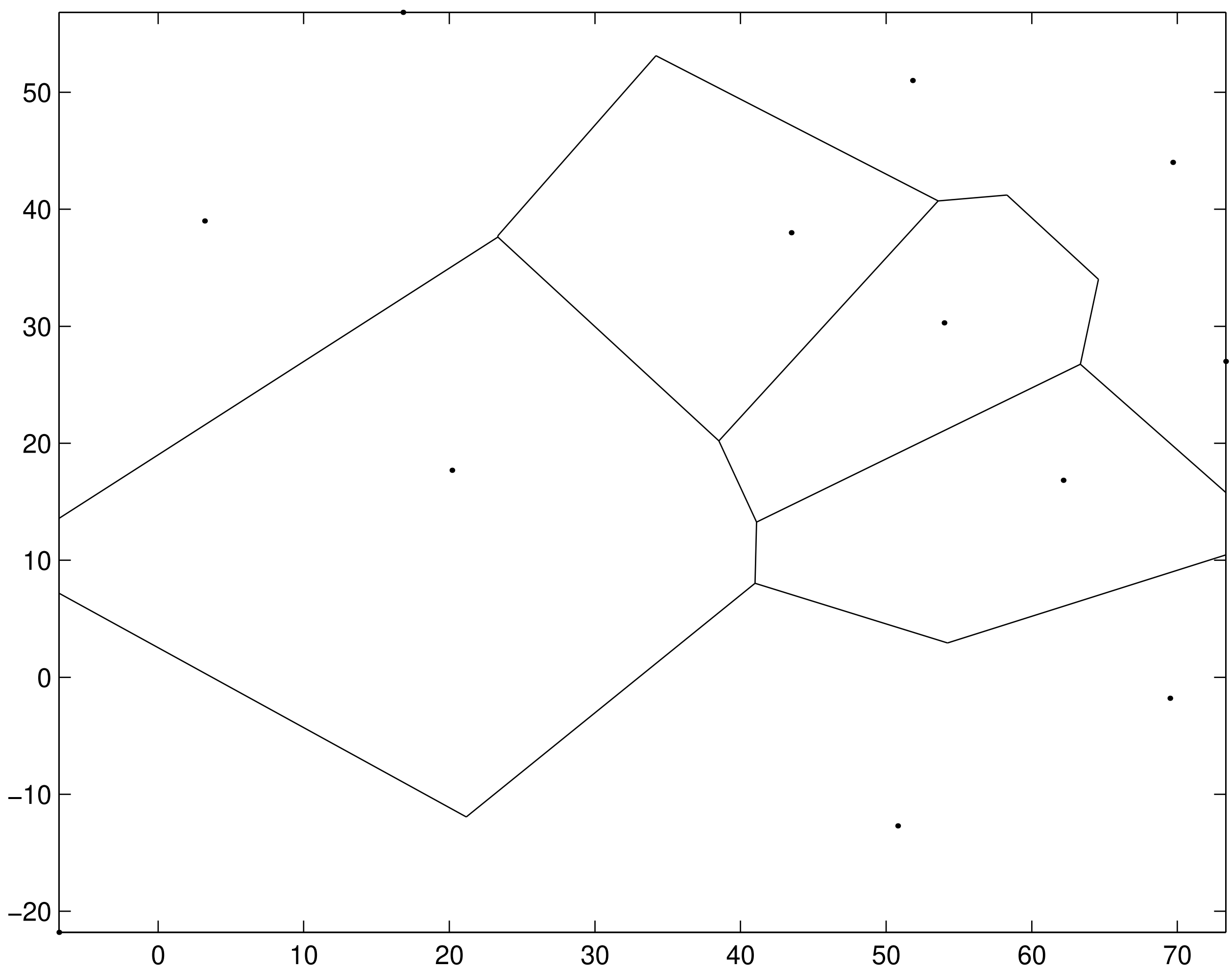

Figure 6.6: Voronoi diagram of the twelve upper air stations used. Note that three of the stations are not located inside the boundaries of the computational grid.

6.1.4 Pollutant Data

In this case study, a hypothetical pollutant is used whose gas-phase characteristics are those of SO_2, a chemical often used in atmospheric dispersion modelling. As such, the standard dispersion parameters [39, Tables 6-1,6-2] and the standard canopy resistances [39, Tables 6-3] have been used in the air dispersion model. However, for the chemokinetic submodel, it was assumed that the pollutant behaves similar to perchlorethylene (PERC), a typical and widely used carcinogene. Further data on PERC can be found in [15].

6.2 Modeling Issues

6.2.1 Meteorologics

With the data available, we decided to use a computational grid of size 80×60. Given the sparsity and the resolution of the population data, land use data, and meteorological data available, this seemed to be a comfortable compromise between computational accuracy and data accuracy. The grid spacing chosen was 9086.9m. This a compromise between several values, since 0.1° in north-south direction has a length of 11 113.65m on the surface of the earth, but the same angle in west-east direction has a length of 7 743.52m on 47° latitude and 6 685.22m on 53° latitude.

All settings in the air dispersion model used are standard ones as recommended in [6], i. e. the time step length is one hour, four puffs are released per hour, twice per hour a sampling of pollutant concentration takes place, etc. No air chemistry has been included in the case study, and no precipitation data has been used. The emission rate of the hypothetical pollutant described in Section 6.1.4 was set to 1000 gs^{-1}. Note that changing this value does *not* represent a scaling of the objective function, due to the nonlinear Michaelis-Menten-term for metabolism in the liver (see next section). The temperature of the exhaust gas was set to 434 °K, while the exhaust speed was set to 50 ms^{-1} for a opening diameter of 4m. No other pollution emitter has been placed on the computational grid.

6.2.2 Chemokinetics

No specific ecosystem model has been developed for this study, but the chemokinetic model in use is an extension of the one found in [15] (cmp. Figure 6.7). Note that the lung compartment from [15] has been split up into lung air and lung blood, since the main uptake route of a pollutant into the body should be modeled as detailed as possible. In this computational study, the ground flux of the pollutant as computed by the MESOPUFF package has not been

Compartment	Volume [m^3]
Air	infinite
Ground	infinite
Lung Air	0.28800
Lung Blood	0.34800
Arterial Blood	0.34800
Rapidly Perfusing Tissue	0.00350
Slow Perfusing Tissue	0.04340
Fat	0.01610
Liver	0.00182

Table 6.3: Volumes of the compartments.

used to calculate further effects of the pollutant when taken up by the human body from the ground, since it was felt that the generic ground flux computed by the meteorological module does not represent adequately the surface flux which most humans are in contact with. For benchmarking reasons, a compartment `GROUND` has been added to the kinetic model as indicated in Figure 6.7. More compartments, representing uptake pathways through the skin or through the intestinal system, can easily be added. Note that only in the liver any metabolism takes place. Since the standard time unit in the meteorological model is 1h, the volume of the lung and the three blood compartments had to be rescaled, see Table 6.3 and cmp. [15]. The volume of each blood compartment is equal to the cardiac output per hour of an average healthy human male of weight 70kg [15], while the volume of the lung compartment is equal to the alveolar ventilation of the abovementioned individual. The air and the ground compartment have an unbounded volume. With the numbering scheme as depicted in Figure 6.7, the matrix Λ from (3.4) has been chosen

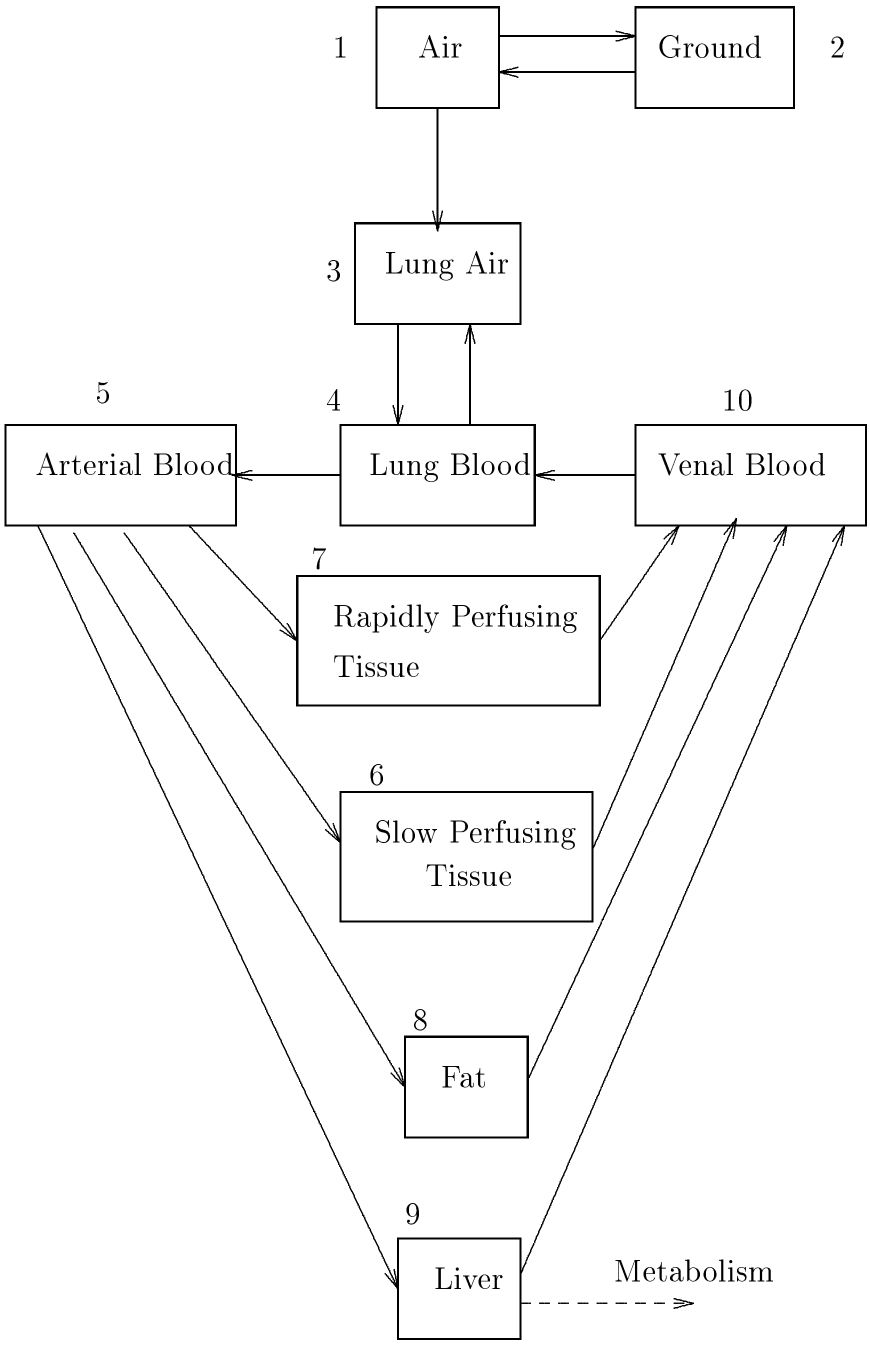

Figure 6.7: Chemokinetic model used in the case study. Numbers represent the numbering scheme for the compartments.

$-1.3994011813 + 0.8065828491i$
$-1.3994011813 - 0.8065828491i$
-9.8612047633
-8.6647859017
-0.9934359459
-0.9344963164
-0.5340749882
-0.0539250117
-0.0252458574
$6.605649012 \times 10^{-40}$

Table 6.4: Eigenvalues of Λ.

to be constant in time and takes the form

$$\Lambda^\top = \begin{pmatrix} -\frac{61}{125} & \frac{1}{10} & & & & & & & & \\ \frac{1}{5} & -\frac{1}{10} & & & & & & & & \\ \frac{61}{125} & & -\frac{29}{31} & \frac{2}{31} & & & & & & \\ & & \frac{29}{31} & -\frac{33}{31} & & & & & & 1 \\ & & & 1 & -1 & & & & & \\ & & & & \frac{1}{4} & -\frac{87}{115.444} & & & & \\ & & & & \frac{51}{116} & & -\frac{153}{17.675} & & & \\ & & & & \frac{1}{20} & & & -\frac{29}{2737} & & \\ & & & & \frac{151}{580} & & & & -\frac{90.6}{9.191} & \\ & & & & & \frac{87}{115.444} & \frac{153}{17.675} & \frac{29}{2737} & \frac{90.6}{9.191} & -1 \end{pmatrix}$$

(zeroes have been omitted). A more realistic model might include time-dependent entries in Λ to account for different activity levels of the cardiac and alveolar system at different times of the day. To give the reader an idea of the speed of damping this matrix enforces, a list of eigenvalues (given with 10 digits of accuracy, computed with Maple VR5.1 to 40 digits of accuracy) is given in Table 6.4. The approximative value of 10^{-40} is a numerical artifact; due to the mass conservation property, at least one of the eigenvalues has to be 0. Of course, all remaining eigenvalues have negative real parts (see Section 3.1), and the system can be classified as moderatively stiff. The step size rule (3.5) employed within OLAF ensures that in this case 10 time steps are taken in the compartment model for one time step of the air dispersion model. The only metabolism takes place in the liver compartment, and the nonlinear

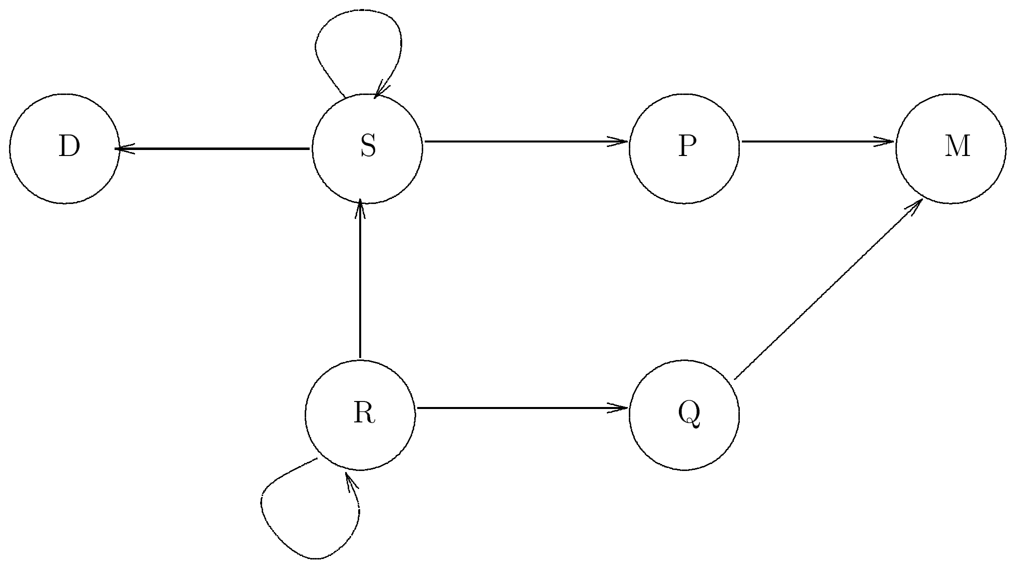

Figure 6.8: Cytodynamic model used in the case study.

term (cmp. Section 3.1.2) takes the form

$$-3.3 \times 10^{-3} \frac{m_9(t)}{2.6745 \times 10^{-4} + m_9(t)},$$

see [15].

6.2.3 Cytodynamics

The cytodynamic model follows the one outlined in [3] and is depicted in Figure 6.8. This model is only used for the cell population in the liver compartment In this model, stem cells (S) may give rise to premalignant cells (P), who, in turn, can mutate to malignant cells (M). But due to toxic effects, stem cells may also die or lose their ability to divide. In this case they are placed in a pool of 'dead' cells (D). Furthermore, cells from a reservoir (R) can replace stem cells, but also mutate into premalignant cells of a different type (Q). Denoting the population sizes by $p_R(t), \ldots, p_M(t)$, the system of ordinary differential equations governing the cell populations is as follows:

$$\begin{aligned}
p'_R(t) &= c(4+\alpha)p_R(t), \\
p'_S(t) &= c((4+0.0893m_9(t))p_R(t) + 3.748p_S(t)), \\
p'_Q(t) &= c(\alpha + 0.407m_9(t))p_R(t), \\
p'_P(t) &= c(\beta + 0.11m_9(t))p_S(t), \\
p'_M(t) &= c((\alpha + 0.407m_9(t))p_Q(t) + (\beta + 0.11m_9(t))p_P(t)), \\
p'_D(t) &= cm_9(t)p_S(t)/35,
\end{aligned}$$

where $c := 1/8760$, $\alpha := 0.703 \cdot 10^{-8}$, $\beta := 0.19 \cdot 10^{-8}$ and

$$\begin{pmatrix} p_R(0) \\ p_S(0) \\ p_Q(0) \\ p_P(0) \\ p_M(0) \\ p_D(0) \end{pmatrix} = \begin{pmatrix} 315 \\ 5000 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Obviously, we start with a completely healthy system. Moreover, the mutation rates are affine functions of the pollutant mass in the liver. In case a constant background pollution is assumed, one should better calculate an equilibium distribution for the compartments and a corresponding equilibrium point of the system above, which can then serve as a starting point.

6.2.4 The Effect Function

Given that a complete cytodynamic model is at our disposal and that in this case study ecological consequences are, hypothetically, of no concern, we used weights $\hat{\omega}_k = 0$ for all compartment numbers k, and choose $\omega_{1,9} = \omega_{2,9} = \omega_{3,9} = \omega_{4,9} = 0$, as well as $\omega_{5,9} = 10^8$ and $\omega_{6,9} = 1$. As a consequence, we value the carcinogenic effects as much more important than the cytotoxic effects (the integral over time of the number of dead cells, cell type no. 6). All other effects, especially the number of premalignant cells, are neglected.

6.2.5 Constraints

OLAF uses grid units for the locational decision variables (y_1, y_2) and meter as the physical dimension of the stack height resp. diameter. It is therefore comparatively simple to add constraints to the optimization process. The following constraints with respect to the decision variables (y_1, y_2, w, h) have been added:

1. Locational constraints:

   (a) The facility has to lie inside a circle with center at 4°W longitude and 50°N latitude and radius 180.738km (20 grid units). This translated readily into the convex quadratic constraint

   $$(y_1 - 40)^2 + (y_2 - 30)^2 - 400 \leq 0.$$

   (b) For hypothetical political reasons, the facility is not allowed to lie within a circle of radius 100km around Paris and within circles of radius 60km around Brussels and Reims. With a grid spacing of

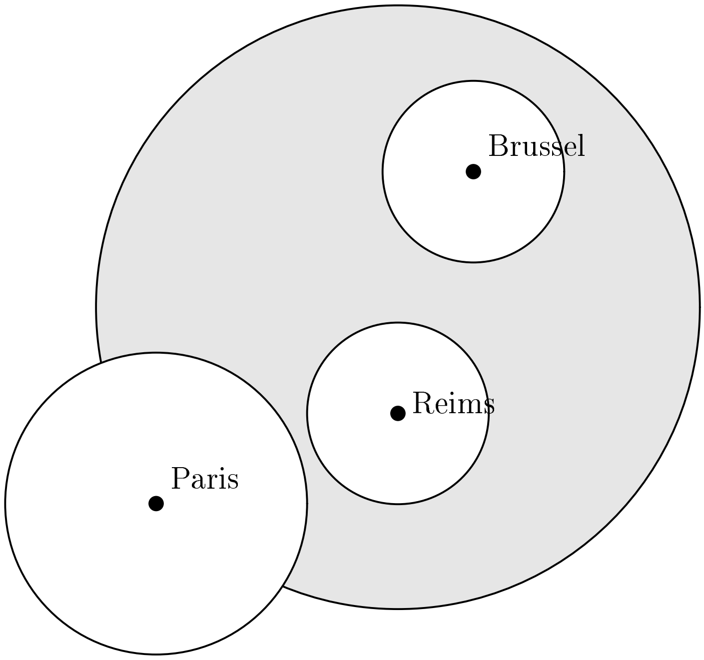

Figure 6.9: The feasible locational region (gray area).

9.0869km, these constraints can be expressed as concave quadratic constraints:

$$\left(\frac{60}{9.0869}\right)^2 - (y_1 - 45.33)^2 - (y_2 - 39)^2 \leq 0$$

for Brussels,

$$\left(\frac{100}{9.0869}\right)^2 - (y_1 - 24)^2 - (y_2 - 17.33)^2 \leq 0$$

for Paris, and

$$\left(\frac{60}{9.0869}\right)^2 - (y_1 - 40.33)^2 - (y_2 - 23)^2 \leq 0$$

for Reims (cmp. Figure 6.9).

2. Engineering constraints:

   (a) The stack height has to be between 20m and 90m, i. e.

   $$20 \leq h \leq 90.$$

   (b) The stack diameter at the top has to be between 2 m and 7m, i. e.

   $$2 \leq w \leq 7.$$

Note that these simple box constraints for the physical characteristics of the stack are rather unrealistic. More realistic constraints have to include a nonlinear coupling between h and w.

6.3 Computational Issues

The starting point was chosen as

$$\begin{pmatrix} y_1 \\ y_2 \\ w \\ h \end{pmatrix} = \begin{pmatrix} 40 \\ 30 \\ 4 \\ 80 \end{pmatrix}.$$

The optimization routine needs an estimate for the accuracy of the objective function evaluation. This value was set to 1×10^{-2}. All other optimization parameters have their standard settings as described in [34].

6.4 Results

The code was developed and tested on an Intel Pentium 330MHz with 128MB main memory, 512KB cache and Linux operating system (Kernel 2.0.35). The compiler used on this platform was the `g77` GNU Fortran compiler version 0.5.19.1. The only optimizing compiler options which have been used are `-fthread-jumps` and `-fdefer-pop`. Care was taken that no similar computationally intensive process was running during the time the case study was benchmarked.

Additional tests have been made on a Silicon Graphics Challenge-L with 8 CPUs (200MHz, 4MB secondary cache each), equipped with IRIX 5.3 operating system and SGI MIPSpro `f77` compiler, and on a Silicon Graphics Indigo^2 with IP26 board and IRIX64 6.1 operating system. On these platforms, only the `-static` compiler option has been used. No porting problems have been experienced. Tests made using shorter time periods confirmed that all machines produced the same numerical values up to 5–6 digits of accuracy.

The code compiled also on a Sun UltraSparc with `f77` compiler version 4.0. However, the resulting code was not able to handle the requested number of input files.

On all platforms, computation times have been obtained with the UNIX `time` command.

Preparing the input files by some filter and interpolation routines took 13 minutes on the Intel platform. After that, the MESOPAC preprocessor needed 30 minutes user time on the Intel, 69 minutes user time on the SGI Challenge, and 116 minutes on the Indigo to process the data described above. As a consequence, all further optimization runs on the SGI platforms have

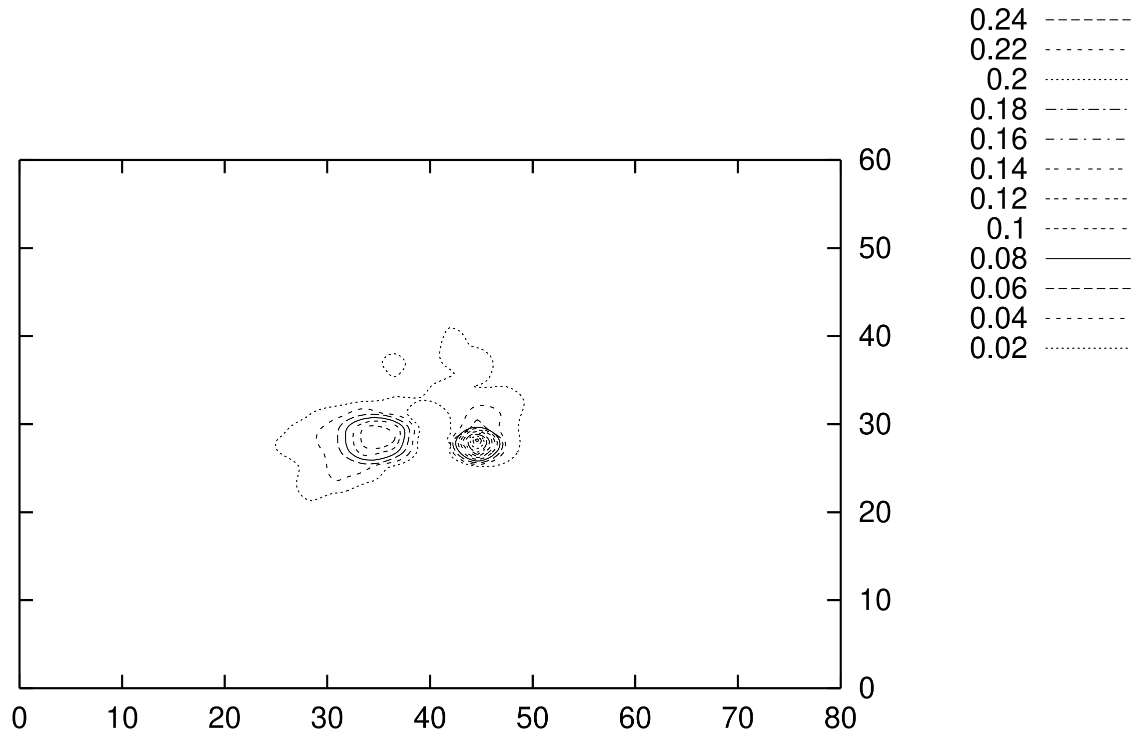

Figure 6.10: Some approximated level curves of the effect function $e(\cdot, \lambda^{(0)})$ for the starting configuration.

been postponed. The resulting data file containing the wind field and other meteorological variables had a size of ca. 250MB. Moreover, the meteorological input data files need an additional amount of disk space of ca. 12MB. Consequently, it has to be expected that a full year simulation takes about 4 hours time on the Intel platform and consumes ca. 1.9GB disk space.

OLAF saves all the time-dependent pollutant masses in the different compartments into temporary files. For the case study considered here, each temporary file needs ca. 25MB of disk space. Given that 10 compartments are used and the meteorological dispersion code writes some additional files, the amount of disk space needed by OLAF for this case study sums up to ca. 325MB (not counting the preprocessor files). Note that after the end of the optimization run these files stay on disk to allow for further investigation of the computed parameters.

Figure Figure 6.10 shows some approximated level curves of the effect function $e(\cdot, \lambda^{(0)})$ for the starting point $\lambda^{(0)}$, while Figure 6.11 (p. 64) shows the effect function $e(\cdot, \lambda^{(0)})$ overlayed over a map of the region considered.

Starting from an objective function value of 70.193, the code needed ca. 260 hours to calculate a local minimum with objective function value of 29.018.

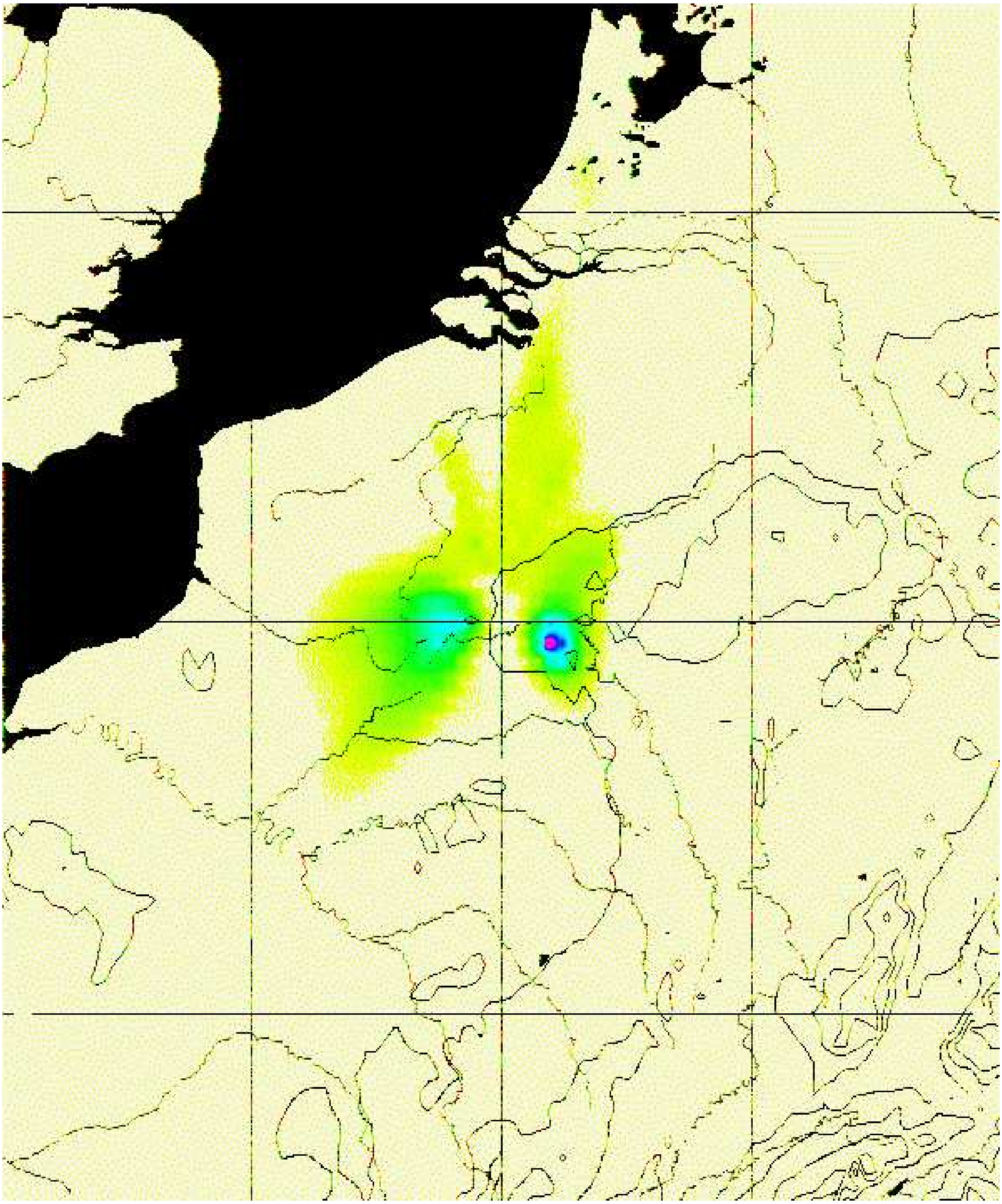

Figure 6.11: The effect function $e(\cdot, \lambda^{(0)})$ for the starting configuration.

The final point is

$$\begin{pmatrix} y_1 \\ y_2 \\ w \\ h \end{pmatrix} = \begin{pmatrix} 58.2918028724108 \\ 22.0181179424968 \\ 79.6872523914867 \\ 4.00000000000000 \end{pmatrix}.$$

To reach this point, the code needed 23 iterations and 201 function evaluations. While this number seems to be comparatively large, one has to take into account the fact that two of the derivatives are approximated by central differences. Therefore, each gradient evaluation (of which there is one per iteration step) needs 4 function evaluations. All other function evaluations are made during the simple Armijo-type linesearch. Therefore, it has to be expected that more efficient line search mechanisms can decrease the number of function evaluations significantly. Note that the stack height and width remain virtually unchanged. This might indicate that the pollution load is insensitive to small scale changes of these variables on the data used. Alternatively, one might have a scaling problem here, and a rescaling of the variables might be in order.

It has to be noted that the relative decrease of the objective function value, $29.018/70.193 \approx 41.3\%$, cannot serve as a measure for the efficency of the code. In fact, Figure 6.3 shows that the function value of the starting point can be made rather large simply by placing the polluting facility as close to Paris as possible. Therefore, the quotient above has to meaningful interpretation.

Figure 6.12 (p. 66) shows the iterations the locational decision variables underwent during the optimization process. Note that this picture is partially incomplete, since the whole decision space is four-dimensional. As it can be seen, the code seems to encounter some convergence problems close to the local minimum found. This might be attributed to the inevitable noise in the objective function, an effect which might also be seen in an iteration vs. function value plot, see Figure 6.13. A notable decrease in step sizes during the iterations has not been observed, see Figure 6.14, p. 67.

Figure 6.15 (p. 68) shows the effect function $e(\cdot, \lambda^{(23)})$ for the final point $\lambda^{(23)}$ overlayed over a map of the region, while Figure 6.16 (p. 69) shows some approximated level curves of the same function.

Figure 6.17 (p. 70) shows the mass of the pollutant in different compartments at grid point $(25, 19)$, which is a close approximation to Paris. Similarly, Figure 6.18 (p. 71) shows the corresponding values for the grid point $(40, 23)$, which is a close approximation to Reims. Moreover, the corresponding values are plotted for Metz (grid point $(61, 21)$, Figure 6.19, p. 72), Düsseldorf (grid point $(68, 43)$, Figure 6.20, p. 73), and Brussel (grid point $(45, 39)$, Figure 6.21, p. 74). Note that for all these figures the scales of the y-axes are differing from picture to picture. For example, the total pollution load in Paris, Reims and

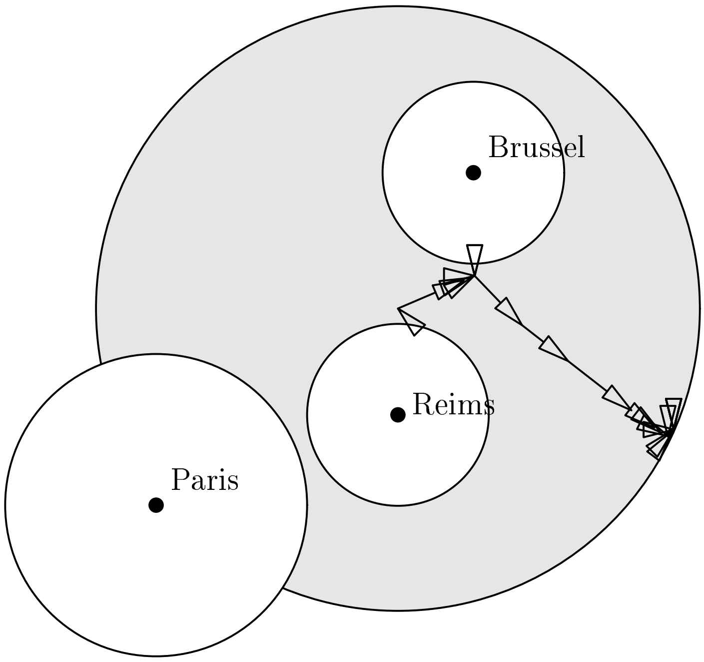

Figure 6.12: Iterations in the locational domain. Line searches and iterations in the stack decision variables are not depicted.

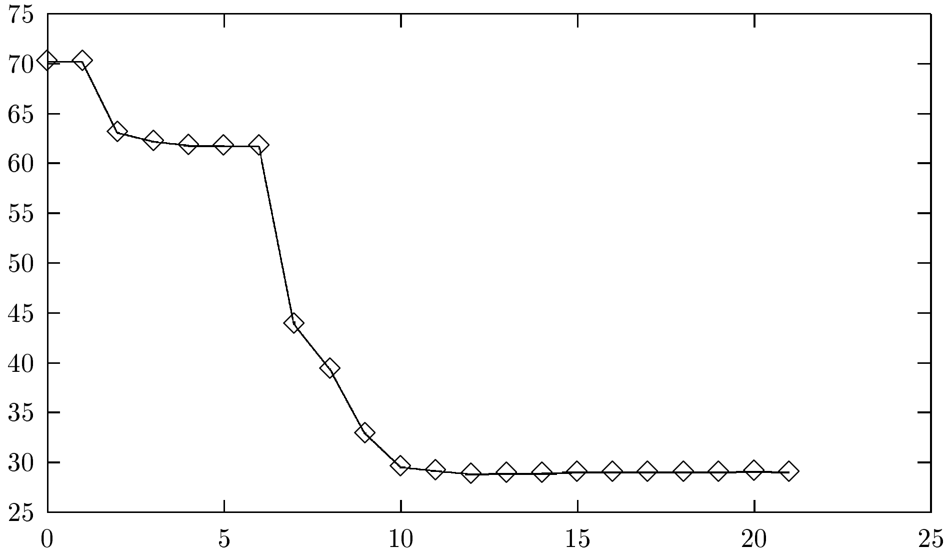

Figure 6.13: Iteration versus function values.

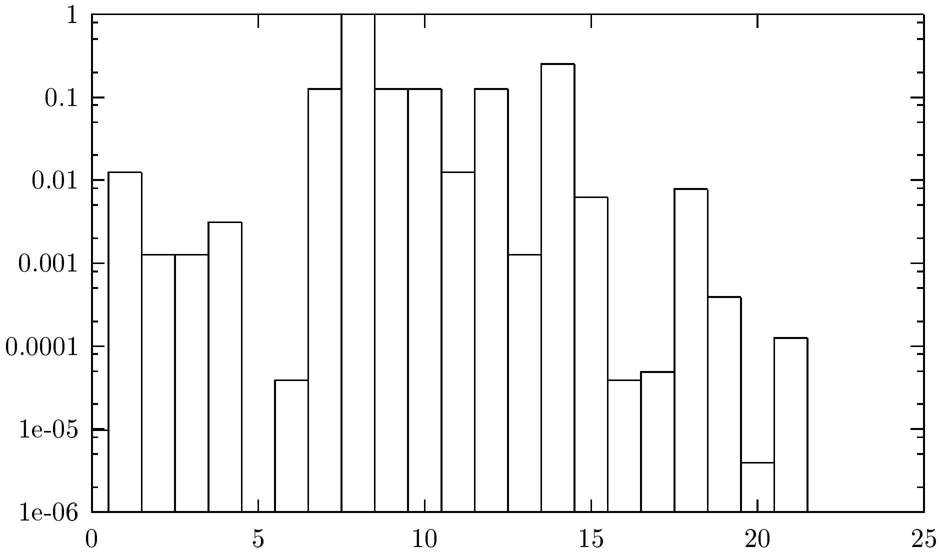

Figure 6.14: Step sizes chosen by the optimization routine. Note that the descent direction in iteration no. 5 was rejected.

%	Routine
65.79	`ODE`
16.64	`CMPCLLR`
7.17	`SAMPLE`
5.64	`GRICN2`
1.07	`ERFDIF`

Table 6.5: Percentages of total running time spent in subroutines. All other subroutines need less time.

Brussel is decreasing when the starting configuration $\lambda^{(0)}$ is compared to the final configuration $\lambda^{(23)}$. However, the opposite is true for the cities of Metz and Düsseldorf. Nevertheless, the decision represented by $\lambda^{(23)}$ has a smaller objective function value than $\lambda^{(0)}$. This is probably due to the high population density around Paris. The same effect can be observed when comparing Figure 6.11 and 6.10 with Figure 6.15 and 6.16.

Finally, Table 6.5 shows the profile of OLAF when run on the data used in the case study after two function evaluations, as reported by `gprof` Version 2.9.1. The work load in the optimization module involves mainly linear algebra on matrices of dimension 4×4, so the time spent in the corresponding routines can be neglected. As it can be seen, most of the computation time is spent in subroutine `ODE`, in which the system (3.3) is solved for all grid points. Subroutine `CMPCLLR`, which is called from within `ODE`, computes the cell death

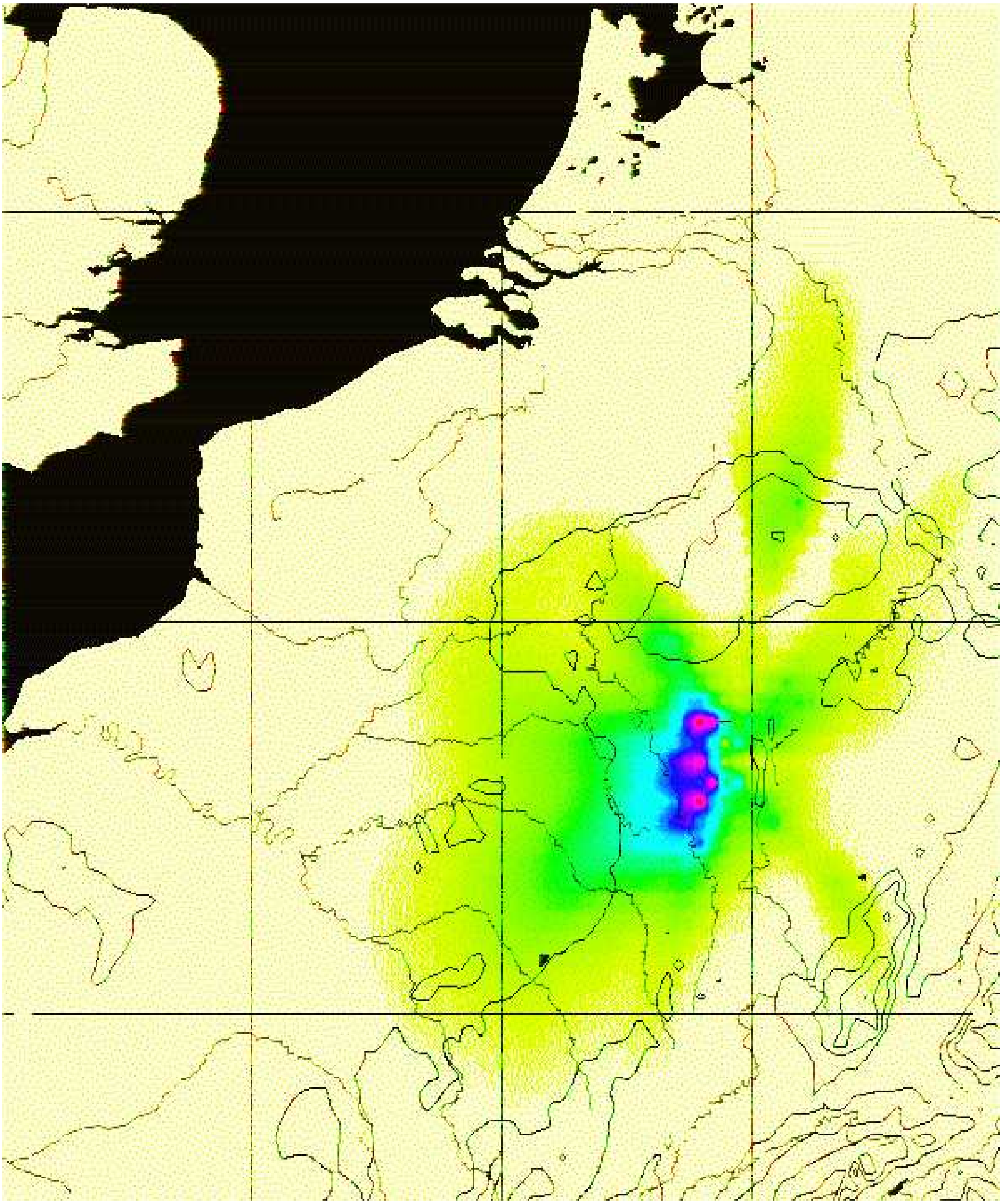

Figure 6.15: The effect function $e(\cdot, \lambda^{(23)})$ for the final configuration.

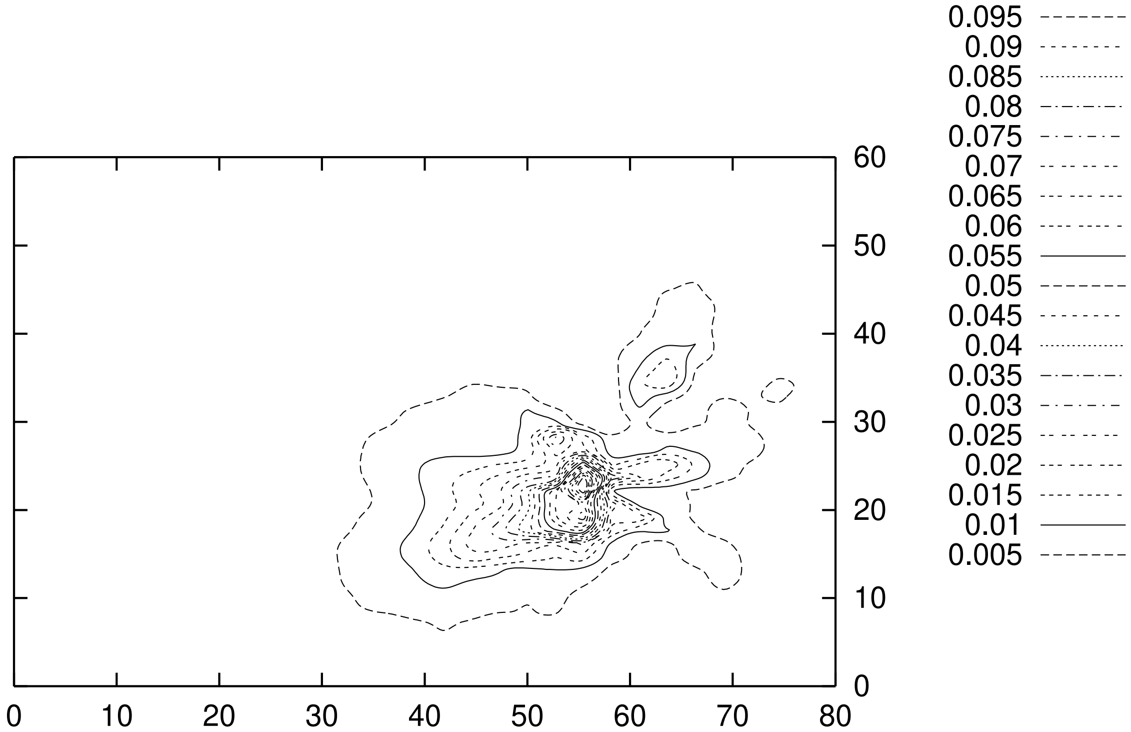

Figure 6.16: Some approximated level curves of the effect function $e(\cdot, \lambda^{(23)})$ for the final configuration.

and differentiation rates of the cytodynamic model described in Section 6.8. All other routines listed are subroutines of the MESOPUFF code. The table shows the typical sharp profile of a numerical analysis code, where most of the time is spent in a small number of routines, leaving ample space for further optimization and fine tuning.

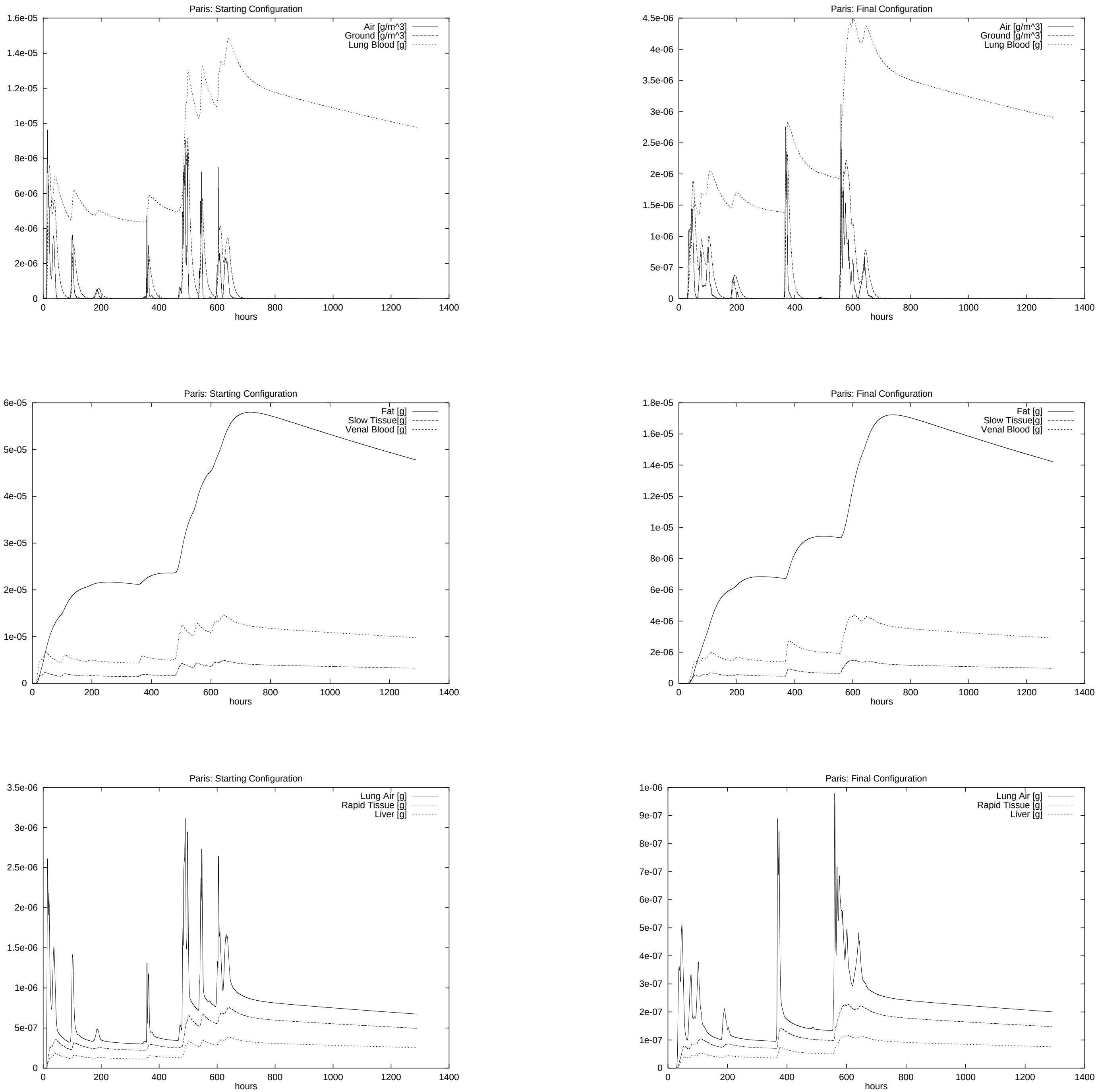

Figure 6.17: Time-dependent pollutant mass in different compartments at Paris for the starting configuration $\lambda^{(0)}$ and the final configuration $\lambda^{(23)}$.

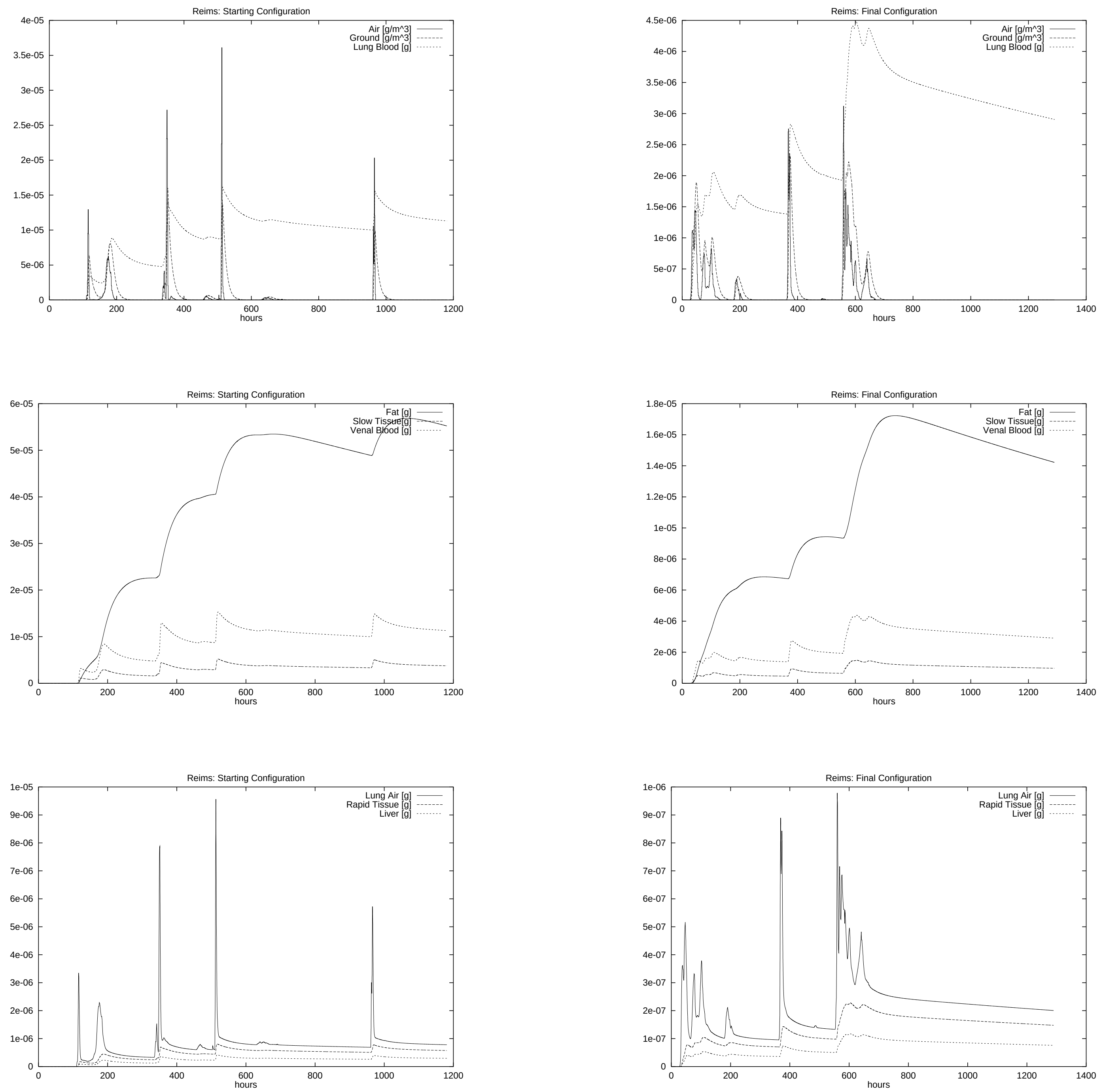

Figure 6.18: Time-dependent pollutant mass in different compartments at Reims for the starting configuration $\lambda^{(0)}$ and the final configuration $\lambda^{(23)}$.

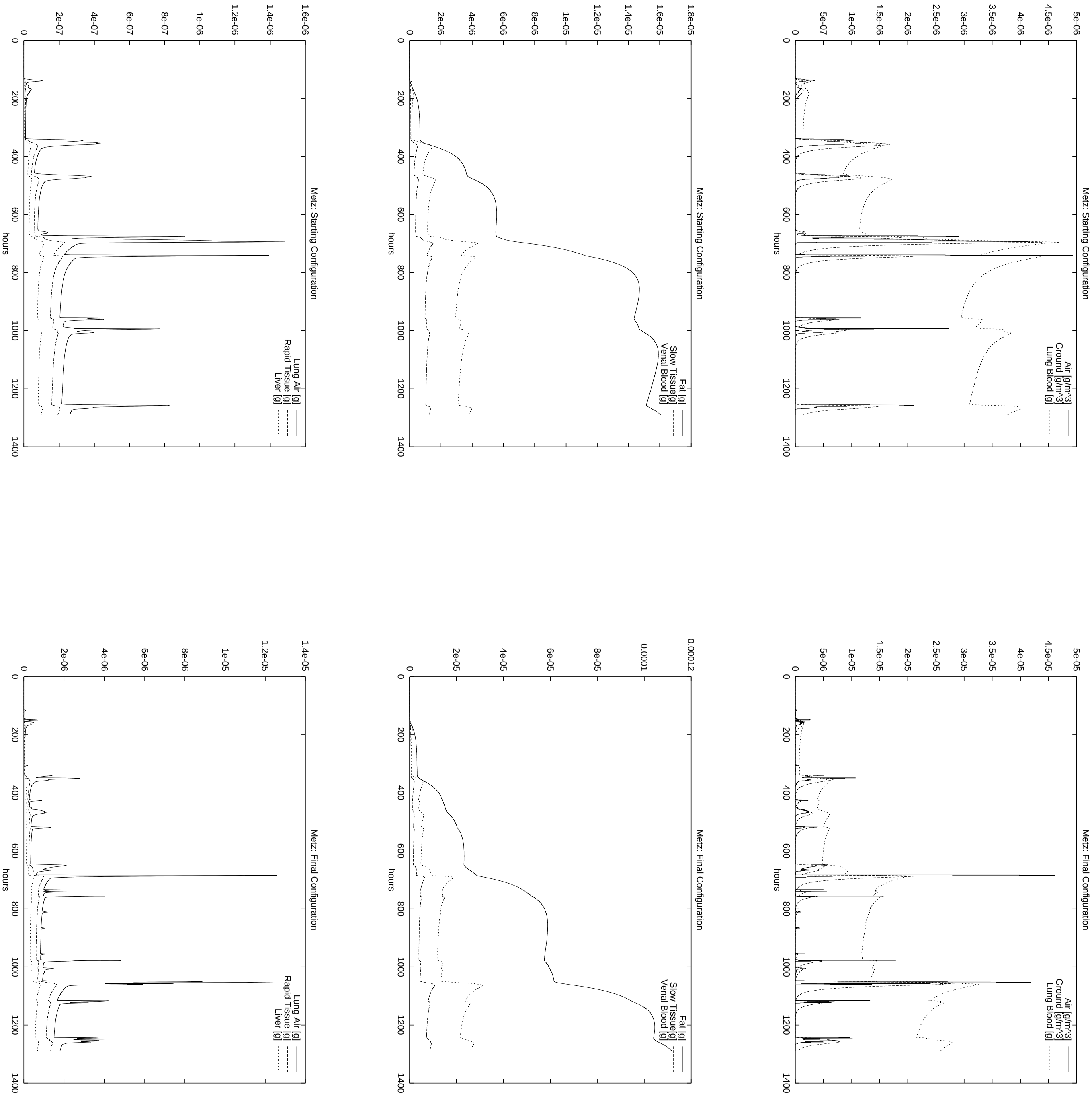

Figure 6.19: Time-dependent pollutant mass in different compartments at Metz for the starting configuration $\lambda^{(0)}$ and the final configuration $\lambda^{(23)}$.

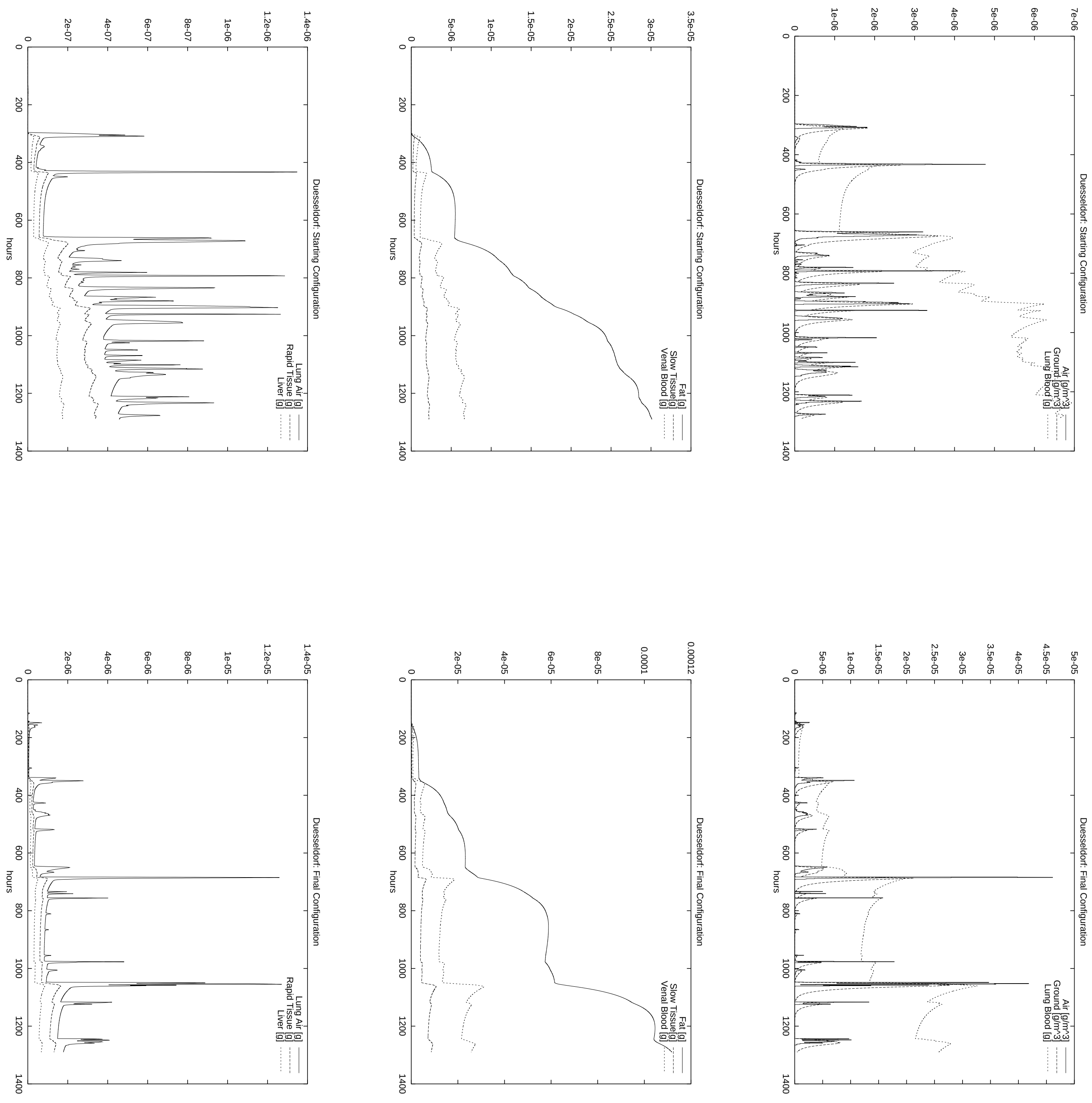

Figure 6.20: Time-dependent pollutant mass in different compartments at Düsseldorf for the starting configuration $\lambda^{(0)}$ and the final configuration $\lambda^{(23)}$.

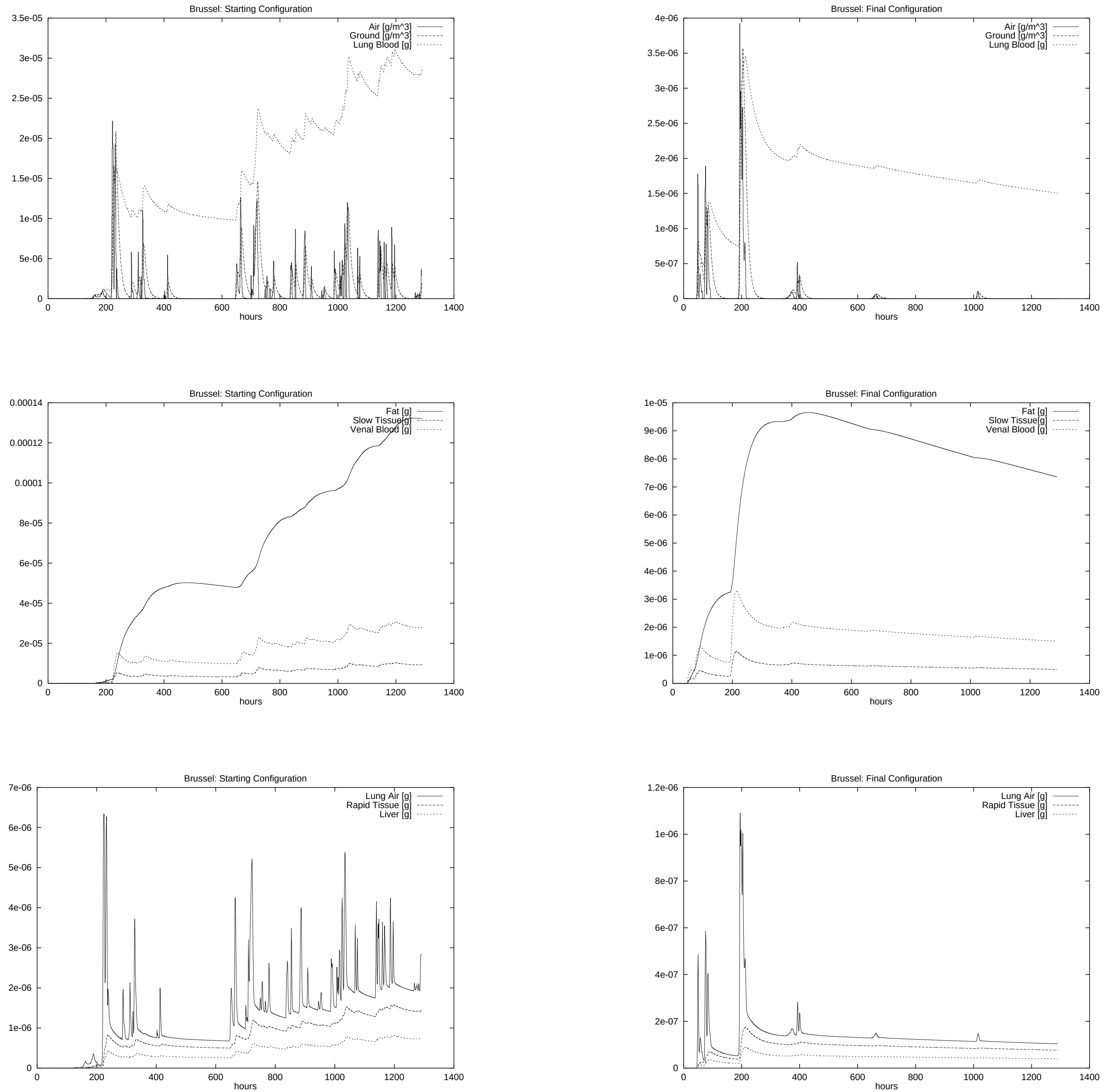

Figure 6.21: Time-dependent pollutant mass in different compartments at Brussel for the starting configuration $\lambda^{(0)}$ and the final configuration $\lambda^{(23)}$.

Chapter 7

Conclusions

The developed prototype code OLAF is capable of solving complex, highly realistic air pollution facility location problems. The code is able to follow the trace of the pollutant under consideration over different ecotrophic levels and into different individual organs of a modeled individual. There, detailed cytodynamic effects are simulated in order to evaluate the health effect of the pollutant on humans or other beings. A case study based on real data has shown that highly realistic models of pollutant fate and effect can be used together with realistic atmospheric dispersion models and in conjunction with an optimization process as a decision making tool. As it had to be expected, the computational workload to find optimal lcoations is far from trivial, and at the present moment the code can probably not be used in an interactive mode, except for very simple problems.

While we have shown that the overall strategy for estimating and minimizing ecological effects and health effects is feasible, several open questions remain. For example, at the present moment the code is able to compute local minima only. Global minima, are, of course, of high interest to the decision maker. It remains to be seen if highly efficient global optimization techniques can or should be applied to the pollution model developed or if it will be necessary to resort to stochastic techniques of inherently low efficiency. Moreover, several modeling issues, especially with respect to the agglomeration function, remain open. A robust, realistic and computationally effective formulation for the multicriteria problem at hand still needs to be found. If a reformulation of the problem as a stochastic optimization problem is helpful is not clear at all. Other atmospheric dispersion models which include nonlinear chemistry of the pollution should be tested, too, as it is the case with other numerical integration techniques for the ordinary differential equations occuring in the model. Another line of research is to investigate the effect of time-dependent emission rates, a particular interesting subject since these emission rates might be considered as decision variables, albeit heavily constrained ones. Moreover, further work is necessary with respect to the evaluation of synergistic effects caused by several pollutants instead of only one. This, together with other open questions, is a subject of further research.

Bibliography

[1] Mahyar A. Amouzega and Khosrow Moshirvaziri. Detemining optimal pollution control policies: An application of bilevel programming. *European Journal of Operational Research*, 119(1):100–120, November 16, 1999.

[2] Dose-response studies of genotoxic rodent carcinogens: thresholds, hockey sticks, hormesis or straight lines? *BELLE Newsletter*, 3(3), February 1995. Electronically distributed via `http://www.belleonline.com/`.

[3] Kenneth T. Bogen. Mechanistic Model Predicts a U-Shaped Relation of Radon Exposure to Lung Cancer Risk Reflected in Combined Occupational and U.S. Residential Data. *BELLE Newsletter*, 7(2), November 1998. Electronically distributed via `http://www.belleonline.com/n3v72.html`.

[4] Frank de Leeuw, Erik Berge, Knut Grønskei, and Maria Tombrou. Review on requirements for models and model application. Electronically available via `http://www.etcaq.rivm.nl/`, July 1996.

[5] Hisham A. El-Masri, John D. Tessari, and Raymond S. Yang. Exploration of an interaction threshold for the joint toxicity of trichlorethylene and 1,1-dichlorethylene: utilization of a PBPK model. *Archive of Toxicology*, 70:527–539, 1996.

[6] A modeling protocol for applying MESOPUFF II to long range transport problems. Technical Report EPA-454/R-92-021, U. S. Environmental Protection Agency, Office of Air Quality Planning and Standards, Technical Support Division, Research Triangle Park, North Carolina 27711, October 1992. Electronically distributed via `http://www.epa.gov/ttn/scram/guidance/guide/mesoprot.zip`

[7] Chapter 8. Dose-Response Modeling for 2,3,7,8-TCDD. Workshop Review Draft. Electronically distributed via `http://www.epa.gov/ORD/WebPubs/dioxin/dioxin8.pdf`, January 1997.

[8] Jörg Fliege and Stefan Nickel. An Interior Point Mehod for Multi-facility Location Problems with Forbidden Regions. Ergebnisberichte Angewandte Mathematik 142, Universität Dortmund, Fachbereich Mathematik, Universität Dortmund, 44221 Dortmund, Germany, March 1997. Submitted.

[9] Arthur M. Geoffrion. Proper Efficiency and the Theory of Vector Maximization. *Journal of Mathematical Analysis and Applications*, 22:618–630, 1968.

[10] Lev R. Ginzburg. *The Theory of Population Dynamics: Back to First Principles*, pages 70–79. Number 71 in Lecture Notes in Biomathematics. Springer Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1986.

[11] E[rnst] Hairer, S[yvert Paul] Nørsett, and G[erhard] Wanner. *Solving Ordinary Differential Equations I*. Number 8 in Springer Series in Computational Mathematics. Springer Verlag, Berlin, Heidelberg, New York, 1987.

[12] Thomas G. Hallam, Ray R. Lassiter, and S. A. L. M. Kooijman. *Effects of Toxicants on Aquatic Populations*, pages 352–382. In Levin et al. [22], 1989.

[13] J. L. Heffter. The variations of horizontal diffusion parameters with time for travel periods of one hour and longer. *Journal of Applied Meteorology*, 4:153–156, 1965.

[14] Ø. How, Z. Zlatev, R. Berkowicz, A. Eliassen, and L. Prahm. Comparison of numerical techniques for use in air pollution models with non-linear chemical reactions. *Atmospheric Environment*, 23(5):967–983, 1989.

[15] Sastry S. Isukapalli. *Uncertainty Analysis of Transport-Transformation Models*. PhD thesis, Rutgers, The State University of New Jersey, New Brunswick, New Jersey, U.S.A., 1999. Electronically availaible at `http://www.ccl.rutgers.edu/~ssi/thesis/`.

[16] J. Jahn. *Mathematical Vector Optimization in Partially Ordered Spaces*. Verlag Peter Lang, Frankfurt a. M., Germany, 1986.

[17] Geneviève Jaubert, Catherine Piriou, Scot M. Loehrer, Alain Petitpa, and James A. Moore. The FASTEX experiment data archive. Technical Report 57, Meteo-France/CNRM/GMME, Météo-France, 1 Quai Branly, 75340 Paris Cedex 07, France, April 1999. Electronically available via `http://www.cnrm.meteo.fr:8000/dbfastex/index.html`.

[18] S. E. Jørgensen. Modelling the distribution and effec of heavy metals in an aquatic ecosystem. *Journal of Ecologic Modelling*, 6:199–223, 1979.

[19] S. E. Jørgensen. *Fundamentals of Ecological Modelling.* Number 9 in Developments in Environmental Modelling. Elsevier, Amsterdam, Oxford, New York, Tokio, 1986.

[20] John Karkazis. Long range environmental impact assessment for activities causing air borne pollution — Applications in Belgium. Technical Report BEIF/104, Vakgroep Beleidsinformatica, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium, April 1998.

[21] John Karkazis and Brian Boffey. Modelling pollution spread. *Studies in Locational Analysis*, 7, November 1994. EWGLA 7 proceedings.

[22] Simon A. Levin, Thomas G. Hallam, and Louis J. Gross, editors. *Applied Mathematical Ecology.* Springer Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1989.

[23] Marco Locatelli and Fabio Schoen. Random Linkage: a family of acceptance/rejection algorithms for global optimisation. *Mathematical Programming*, 85(2):379–396, 1999.

[24] Daniel H. Loughlin and S. Ranjithan. The Neighborhood Constraint Method: A Genetic Algorithm-Based Multiobjective Optimization Technique. In *Proceedings of the 7th International Conference on Genetic Algorithms.* Morgan Kaufmann, July 1997.

[25] Daniel Hopkins Loughlin. *Genetic Algorithm-Based Optimization in the Development of Tropospheric Ozone Control Strategies: Least Cost, Multiobjective, Alternate Generation, and Chance-Constrained Applications.* PhD thesis, Civil Engineering Department, North Carolina State University, Raleigh, North Carolina, U.S.A., 1998. Electronically availaible at `http://www4.ncsu.edu/~dhloughl/Documents/Dissertation/`.

[26] H. Malcorps. Letter from the directeur of the Koninklijk Meteorologisch Instituut van België, March 5, 1999.

[27] Nicolas Moussiopoulos, Erik Berge, Trond Bøhler, Frank de Leeuw, Knut-Erik Grønskei, Sofia Mylona, and Maria Tombrou. Ambient air quality, pollutant dispersion and transport models. Electronically available via `http://www.etcaq.rivm.nl/`, January 1996.

[28] J[ames] D. Murray. *Mathematical Biology.* Springer Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1989.

[29] Van Luoc Ngo, Quang A. Dang, and Cong Dieu Nguyen. Analytic and numerical solution of some problems in air pollution. *Southeast Asian Bulletin of Mathematics*, pages 105–117, 1993. Special issue.

[30] M. Talat Odman. Horizontal Advection Solver Uncertainty in the Urban Airshed Model. Electronically distributed via `http://www.arb.ca.gov/eos/research/research.html`, April 30 1996.

[31] J. S. Olson, J. A. Watts, and L. J. Allison. Carbon in live vegetation of major world ecosystems. Technical Report Environmental Sciences Division Publication No. 1997, Oak Ridge National Laboratory, Oak Ridge, Tennessee, U.S.A., 1983.

[32] R. B. Rood. Numerical advection algorithms and their role in atmospheric transport and chemistry models. *Review of Geophysics*, 25(1):75–100, 1987.

[33] J. S. Scire, F. Lurmann, A. Bass, and S. R. Hanna. *User's guide to the* MESOPUFF II *model and related processor programs.* U. S. Environmental Protection Agency, Research Triangle Park, North Carolina, EPA-600/8-84-013 edition, 1984.

[34] Peter Spellucci. DONLP2 code and accompanying documentation. Electronically available via `http://www.mathematik.tu-darmstadt.de/ags/ag8/spellucci/spellucci.html` or `http://plato.la.asu.edu/donlp2.html`.

[35] Robert V. Thomann. *Deterministic and Statistical Models of Chemical Fate in Aquatic Systems*, pages 323–351. In Levin et al. [22], 1989.

[36] Walter Tobler, Uwe Deichmann, Jon Gottsegen, and Kelly Maloy. The Global Demography Project. Technical Report TR-95-6, National Center for Geographic Information and Analysis, Department of Geography, University of California, Santa Barbara CA 93106-4060, April 1995. Electronically available via `http://www.ciesin.org/datasets/gpw/globldem.doc.html`.

[37] Stefan Trapp and Michael Matthies. *Chemodynamics and Environmental Modelling.* Springer Verlag, Berlin, Heidelberg, New York, 1998.

[38] D. B. Turner. *Workbook of Atmospheric Dispersion Estimates.* Number 999-AP-26. U.S. Department of H.E.W., Public Health Service, 1970.

[39] U. S. Environmental Protection Agency, Research Triangle Park, North Carolina. *A revised user's guide to* MESOPUFF II *(V5.1)*, EPA-454/b-94-025 edition, August 1994. Electronically distributed via `http://www.epa.gov/ttn/scram/userg/other/`.

[40] M. van Loon. Tests on Semi-Lagrangian Transport and Interpolation. Technical Report NM-R9217, Centrum voor Wiskunde en Informatica, CWI, P. O. Box 94079, 1090 GB Amsterdam, The Netherlands, 1992.

[41] J. G. Verwer, W. H. Hundsdorfer, and J. G. Blom. Numerical Time Integration for Air Pollution Models. Technical Report MAS-R9825, Centrum voor Wiskunde en Informatica, CWI, P. O. Box 94079, 1090 GB Amsterdam, The Netherlands, November 30, 1998. Electronically available as `ftp://ftp.cwi.nl/pub/CWIreports/MAS/MAS-R9825.ps.Z`.

[42] J. Vogl, A. Heigl, and K. Schäfer, editors. *Handbuch des Umweltschutzes*, volume II-2, Appendix 3.1, chapter Technische Anordnung zur Reinhaltung der Luft. ecomed, Landsberg am Lech, Germany, 1986.

[43] Paul Wessel and Walter H. F. Smith. New version of the generic mapping tools released. Technical report, American Geophysical Union, 1995. Electronically available as `http://www.agu.org/eos_elec`.