**Atmospheric
Chemistry
and Physics**

Technical Note: The air quality modeling system Polyphemus

V. Mallet[1,3], **D. Quélo**[2], **B. Sportisse**[3,1], **M. Ahmed de Biasi**[1,3], **É. Debry**[3], **I. Korsakissok**[3,1], **L. Wu**[1,3], **Y. Roustan**[3,1], **K. Sartelet**[3], **M. Tombette**[3], and **H. Foudhil**[3]

[1]INRIA, Paris-Rocquencourt Research Center, France
[2]IRSN, Fontenay-aux-Roses, France
[3]Université Paris-Est, CEREA (Joint Laboratory ENPC – EDF R&D), Marne la Vallée, France

Abstract. Polyphemus is an air quality modeling platform which aims at covering the scope and the abilities of modern air quality systems. It deals with applications from local scale to continental scale, using two Gaussian models and two Eulerian models. It manages passive tracers, radioactive decay, photochemistry and aerosol dynamics. The structure of the system includes four independent levels with data management, physical parameterizations, numerical solvers and high-level methods such as data assimilation. This enables sensitivity and uncertainty analysis, primarily through multimodel approaches. On top of the models, drivers implement advanced methods such as model coupling or data assimilation.

1 Introduction

Air quality modeling has reached a point where models seem mature enough for operational use (e.g., daily photochemical forecasts) and for decision support with screening studies. This motivates the development of detailed models and advanced methods such as data assimilation algorithms and ensemble forecasting. The modeling system Polyphemus aims at covering most of the advanced abilities and features of modern air quality systems. In short, Polyphemus is designed and built to:

– handle several dispersion models, from local scale to continental scale, for passive transport, gaseous chemistry and multi-phase chemistry;

– host high-level methods in which a model is essentially a black box, e.g., model validation, data assimilation methods, ensemble forecast, models integration;

– provide facilities to manipulate atmospheric data and to manage numerous physical parameterizations.

Polyphemus is thus a platform that gathers several dispersion models, several advanced methods (in particular for data assimilation) and miscellaneous tools (mainly for data processing). The multiple choices and methods in the system justify its name: the roots of Polyphemus, in Ancient Greek, mean "multiple speeches". It was initially developed around the chemistry-transport model Polair3D (Boutahar et al., 2004) which has since been rewritten (Mallet, 2006) for further integration in Polyphemus.

To achieve its goals and to ensure a perennial framework, Polyphemus has been built with independent blocks (at global level) and with object oriented code (at low level, in each block). The design of Polyphemus is addressed in Sect. 2, with a focus on its main components and their relations. Technical details that would require specific background are not explained in this paper, but in Mallet et al. (2005).

Section 3 reviews Polyphemus content. It describes its dispersion models, from local scale to continental scale. The supported chemical mechanisms are introduced. It finally shows its advanced methods in data assimilation and ensemble forecast.

A few illustrations are proposed in Sect. 4: Gaussian dispersion with different dispersion parameterizations, aerosol simulation, ensemble forecast for photochemistry and data assimilation with sequential and variational methods.
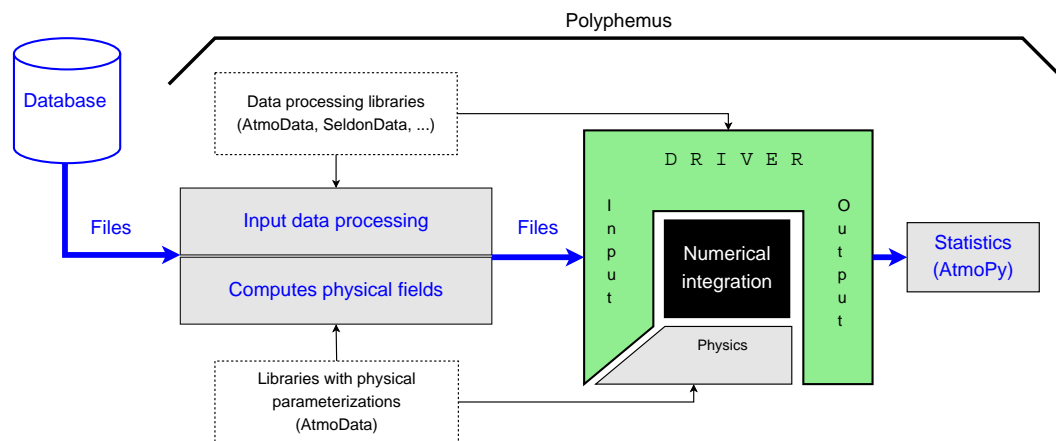
Correspondence to: V. Mallet
(vivien.mallet@inria.fr)

Fig. 1. Polyphemus overall work flow. Data is first processed with dedicated C++ libraries (mainly SeldonData and AtmoData). The library AtmoData (Mallet and Sportisse, 2005b; Njomgang et al., 2005) provides in addition most physical parameterizations needed in the preprocessing step. The preprocessing step outputs most coefficients of the reactive-transport equation. The numerical model (e.g., an Eulerian chemistry-transport model) integrates in time this equation, and a driver manages the numerical model (e.g., to perform data assimilation). For post-processing (statistics, visualization, ...), Python scripts and the Python library AtmoPy are used.

2   Design: technical description

2.1   Overall architecture

Atmospheric dispersion models usually rely on large amounts of data coming from miscellaneous sources, e.g. meteorological models, chemical databases or emission inventories. Data is then processed by means of interpolations, filtering or reanalysis and it is then involved in many computations. Among the computations, there is a prominent part for physical parameterizations. Many parameterizations are put together in order to estimate the coefficients of a reactive-transport equation. With these coefficients available, a numerical solver performs the time integration of species concentrations.

In addition, "high-level" methods (mainly data assimilation and ensemble forecast) are used to analyze or improve output concentrations. These methods are implemented on top of the numerical solver and may modify the concentrations and the input data to the numerical solver. At the end of the simulation process, post-processing is performed: statistical analysis, comparisons to measurements, visualization and other data manipulations.

The overall design of Polyphemus follows the main steps of the process; one may identify four independent levels:

1. data management, data processing facilities (input/output operations, coordinate transformations, interpolations, ...);

2. the physical parameterizations (turbulence closure, deposition velocities, parameterizations in the general dynamic equation for aerosols, ...);

3. the numerical solver that computes concentrations in the simulation domain and for all time steps (unless the model is stationary);

4. the high-level methods, in which the chemistry-transport model is simply viewed as a function.

Data management and processing is eased by the use of object-oriented libraries. The physical parameterizations are gathered, as much as possible, in a dedicated library, so that they may be reused in different parts of the system, or in other systems. In Polyphemus, preprocessing programs compute most coefficients of the reactive-transport equation. These programs rely on data management libraries and on the physical parameterizations. Details are provided in Sect. 2.2.

After the preprocessing step, the numerical solver is called (Sect. 2.3). Since there may be an additional layer for data assimilation or any high-level method, the numerical solver is encapsulated in a *driver* (Sect. 2.4). Each high-level method is implemented in a driver which makes the relevant calls to the underlying numerical solver (or model).

A summary of the work flow is shown in Fig. 1. A few details about the main steps are provided in the following sections.

The main components are split and supposed to be independent. Even if the preprocessing step is constrained by the requirements of the numerical solver (e.g., by its coordinate system), it is implemented and run independently. Preprocessing programs may thus be used with different drivers and numerical solvers. However the main reason for this split is to avoid all-in-one models.

Preprocessing programs may then be improved without working with the whole system. Alternative programs can

be easily added to the system, or written by the modeler for a specific application. Moreover it enables to save computational time in case a given simulation is launched many times (e.g., process study, ensemble forecast, ensemble filters in data assimilation).

Another split to emphasize is at driver level. High-level methods (drivers) and the numerical solver are derived independently. For instance, a Kalman filter is developed to be applied to any model.

Note that three computer languages are used in Polyphemus:

– C++ is the main language due to its advanced object-oriented abilities; it is used for most computations;

– Fortran 77 is used for parts of the code to be automatically differentiated (generation of the tangent linear model and of the adjoint model); it is also used because of historical reasons;

– Python covers other needs, that is, process management and post-processing (statistics, visualization).

A careful implementation guarantees a good portability of most parts of the system on Unix, Linux and Windows and on different hardware architectures.

2.2 Data processing and physical parameterizations

Data processing and calls to physical parameterizations are prominent in the preprocessing step. At this stage, raw input data (meteorological data, land use data, chemical data, ...) are interpolated and modified, and physical parameterizations are called to compute additional variables such as vertical diffusion coefficients or deposition velocities.

In Polyphemus, the guideline is to perform as much computations as possible in the preprocessing step. Computing parameterized fields during the numerical integration is still acceptable (and eased by the availability of the libraries), but the question of whether or not a given field should be computed during the numerical integration depends on:

– the number of available parameterizations to compute the field and the reliability of the parameterizations: if there are many parameterizations for the field (e.g. for the vertical diffusion coefficient: Louis, 1979; Troen and Mahrt, 1986), it should not be computed inside the numerical solver. It allows to work on the various parameterizations available for a given field without the constraints of the numerical integration. On the other hand, if a parameterization is reliable and is not going to change, it may be put directly in the integration process.

– the computational time needed to compute the field: if the computational time is high, the parameterization should be called only once and therefore outside the numerical solver. A given simulation is often launched several times (especially for ensemble forecast or data assimilation) and saving the computation of several fields is of high interest.

– the size of the data: if the data (output of a parameterization) cannot be stored on disk because of their huge size (e.g. the scavenging coefficients that depend on the time, the 3-D position and the species), they must be computed inside the numerical solver. When stored, every input data field is saved with its own timestep (e.g., hourly emissions), independently of the model timestep.

2.3 Numerical solver (model)

The numerical solver computes the concentrations (and maybe other fields like deposited quantities). It usually performs the time integration of the reactive-transport equation in 3-D, in an Eulerian framework. It could also be a Gaussian plume model, a Gaussian puff model, a Lagrangian model, a soil model, ...

A simulation is the combination of input fields and of a numerical solver. In next sections, the numerical solver is called the numerical *model* and it is assumed that the set of input fields is the *configuration* of the (numerical) model. The configuration is therefore determined by the preprocessing programs called and by their options.

From a technical point of view, in Polyphemus, a numerical model is a C++ object. It can be controlled through its interface. All models in Polyphemus have a minimum interface: to initialize them and to call their numerical solver. Based on this minimum interface, it is possible to perform a simple forward simulation. A more complete interface enables, for example, the use of data assimilation algorithms with the model (see Sect. 2.4).

In order to include a model in Polyphemus, a C++ object should be written, possibly on top of an existing model. Starting from a model written in Fortran 77 (for example), if the model is at least slightly structured, it should be possible to build an interface object in C++ with little changes in the core of the model.

2.4 Drivers

A driver controls the model and may interact with it. The model is seen as a black box with an interface to manage it.

The most simple driver performs a forward simulation. It first calls the method (that is, a function part of the interface) that initializes the model. It then executes the time loop with calls to the method that integrates the underlying equation over one time step.

Advanced drivers require an interface with further methods. For instance, a sequential data assimilation algorithm updates the species concentrations at every step where observations are available. Consequently, for such an algorithm, the interface of the numerical model must include a method to modify the value of the concentrations. With a few other

methods, it is possible for the driver to control the model for data assimilation.

The drivers included in Polyphemus are developed independently of the models. Any combination of drivers and models should be valid providing the model interface is sufficient.

Note that a driver is an object, just like any model. If a few methods are added to the driver, it may be seen as a model and may in turn be used within a driver. This can be useful to embed an algorithm in another one, e.g., to include in ensemble forecasts a model with data assimilation. The use of driver is also recommended to couple models (e.g. an atmospheric model and a soil model), which results in a new model.

2.5   Process management and post-processing

Although Polyphemus design is less advanced with respect to post-processing and process management, a guideline is to share as much material as possible and to find a unity among all models and applications. The versatile computer-language Python was chosen to cover these needs. This language has proven to be suited to process management in computer science. In addition, its object-oriented abilities together with interactive command line let it be a convenient post-processing tool. It is noteworthy that other research teams in atmospheric sciences have come to the same conclusion (e.g. with the Climate Data Analysis Tools – http://www-pcmdi.llnl.gov/software-portal/cdat/).

The Python library AtmoPy includes classes and functions for Polyphemus post-processing. It is split into modules for file input/output operations, configuration files management, observations processing, statistics, ensemble methods and visualization. For example, it is used for complete comparisons to observations, e.g. following EPA guidelines (US EPA, 1991). It also includes ensemble combination methods ("superensemble", machine learning – Mallet and Sportisse, 2006a). It provides visualization functions based on Matplotlib and Basemap (http://matplotlib.sourceforge.net/).

3   Content: Polyphemus features

3.1   Models: from local scale to continental scale

Four atmospheric (numerical) models are currently available in Polyphemus. All models have one or more variants: gaseous version, aerosol version, passive version, reactive version, data assimilation versions. The four models are:

- a Gaussian plume model;

- a Gaussian puff model;

- an Eulerian chemistry-transport model called Polair3D (Boutahar et al., 2004), relevant up to continental scale;

- an Eulerian chemistry-transport model called Castor, also relevant up to continental scale, clone written in C++ of the gaseous version of Chimere (Schmidt et al., 2001).

The first two models are applied to dispersion at local scale, possibly with radioactive or biological decay. Both models have a gaseous and an aerosol version. They include dry deposition (Chamberlain or Overcamp formulations: Arya, 1999) and wet deposition (Belot parameterization for gaseous species, Slinn parameterization for aerosol species: Underwood, 2001). The dispersion coefficients may be computed with Briggs parameterization (based on Pasquill classes diagnosed with Turner, 1969), Doury formulation (Doury, 1976) or a parameterization based on similarity theory.

The Eulerian model Polair3D solves the reactive-transport equation, up to continental scale, using operator splitting (sequence: advection, diffusion and chemistry). The advection scheme is a direct space-time third-order scheme with a Koren flux-limiter. Diffusion and chemistry are solved with a second-order order Rosenbrock method (Verwer et al., 2002). At the preprocessing stage, several options and parameterizations are available to compute emissions, deposition velocities, vertical diffusion coefficients, ... A reference simulation may rely on:

- ECMWF or MM5 meteorological data;

- vertical diffusion coefficients based on the Troen and Mahrt parameterization described in Troen and Mahrt (1986) and the Louis parameterization found in Louis (1979);

- USGS[1] or GLCF[2] land cover map;

- the RACM chemical mechanism (Stockwell et al., 1997);

- the SIREAM size-resolved aerosol model (Debry et al., 2007);

- anthropogenic emissions from EMEP[3] processed according to Middleton et al. (1990);

- biogenic emissions computed as proposed in Simpson et al. (1999);

- deposition velocities from Zhang et al. (2003) for gaseous species and Zhang et al. (2001) for aerosols;

---

[1]U.S. Geological Survey, http://edcsns17.cr.usgs.gov/glcc/

[2]Global Land Cover Facility, http://glcf.umiacs.umd.edu/data/landcover/data.shtml

[3]Co-operative Programme for Monitoring and Evaluation of the Long-range Transmission of Air Pollutants in Europe

– boundary conditions from the global chemistry-transport model Mozart 2 (Horowitz et al., 2003) run over a typical year, for gas, and output from the global model GOCART (Chin et al., 2000) for aerosols (sulfate, dust, black carbon and organic carbon).

There are a passive version of Polair3D, a version with gaseous chemistry and a version with aerosols. Their performances have been evaluated in Quélo et al. (2007); Mallet and Sportisse (2004); Sartelet et al. (2007) respectively. The tangent linear models and the adjoint models of the first two versions can be automatically generated (Mallet and Sportisse, 2004) by O∂yssée (Faure and Papegay, 1998). The adjoint model (with respect to the concentrations) is part of the Polyphemus distribution.

3.2 Data assimilation

Four data assimilation algorithms are available in Polyphemus: optimal interpolation, ensemble Kalman filter (Evensen, 1994), reduced-rank square root Kalman filter (Heemink et al., 2001) and 4D-Var (Le Dimet and Talagrand, 1986). They are implemented independently of the models and therefore virtually applied to any model. Currently the Eulerian models Castor and Polair3D have a sufficient interface for sequential algorithms (optimal interpolation and Kalman filters). Only Polair3D is suitable for 4D-Var since the variational algorithm requires an adjoint model.

Assimilated observations are handled by an observation manager (C++ class). There are observation managers for twin experiments (simulated observations) and for measurements from monitoring networks (such as EMEP). Other observation managers can be plugged to Polyphemus. For instance, an observation manager for columns of concentrations (satellite observations) has been used for an ESA project.

3.3 Ensemble forecast

Polyphemus hosts four base models (two Gaussian models and two Eulerian models). New models may be derived from them, using new numerical schemes, new physical formulation and new input data to the models. In a model such as Polair3D, almost every component may be changed (including the chemical mechanism or the numerical schemes) so that a wide range of models can be emulated in its frame. This generates ensemble forecasts with a multimodel approach (Mallet and Sportisse, 2006b).

Just like Monte Carlo simulations are embedded in the ensemble Kalman filter, a driver for Monte Carlo simulations is available in Polyphemus. Castor and Polair3D already have the interface for Monte Carlo simulations. The multimodel approach and Monte Carlo simulations can be coupled this way.

The Python library AtmoPy is used for ensemble post-processing. It enables to linearly combine ensemble simula-

tions so as to produce a better forecast (Mallet and Sportisse, 2006a). The weights of the linear combination are computed based on observations (preceding the forecast). AtmoPy includes machine learning algorithms, algorithms to compute "superensembles" (Krishnamurti et al., 2000) and other aggregation methods.

3.4 Other abilities

In short, the adjoint of Polair3D may be used for sensitivity analysis (Mallet and Sportisse, 2005a).

Modules, called *output savers*, independent of the models and the drivers are available to save concentrations or other quantities defined in models. Gaussian and Eulerian models share the same output savers. They enable to save three-dimensional concentrations, concentrations in a sub-domain, interpolated concentrations at given locations, (dry or wet) deposition fluxes, boundary conditions for a nested simulation (in order to nest Eulerian models), and predicted and analyzed concentrations (in data assimilation algorithms).

A driver can couple two models. Although they are not in Polyphemus yet, two such drivers are in preparation. The first one couples a soil model (degradation) and an atmospheric model for persistent organic pollutants. The second one couples an Eulerian model and a Gaussian puff model for an improved treatment of point emissions (plume-in-grid model).

4 A few illustrations

4.1 Dispersion with a Gaussian model

As stated in Sect. 3.1, Gaussian models may use different dispersion parameterizations: Briggs formulae, Doury formula or a formulation based on similarity theory. Their performances have been evaluated in Korsakissok (2007) and compared to other Gaussian models.

Several statistical measures have been used including the fractional bias FB, the correlation Corr and the proportion FAC2 of concentrations within 50% and 200% of observed values. Let $(y_i)_i$ be the sequence of the n simulated concentrations and $(o_i)_i$ the sequence of the n observations. Their means are $\bar{y} = \frac{1}{n} \sum_i y_i$ and $\bar{o} = \frac{1}{n} \sum_i o_i$. Then

$$\text{FB} = 2 \frac{\bar{y} - \bar{o}}{\bar{y} + \bar{o}} \tag{1}$$

$$\text{Corr} = \frac{\sum_{i=1}^n (y_i - \bar{y})(o_i - \bar{o})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^n (o_i - \bar{o})^2}} \tag{2}$$

$$\text{FAC2} = \frac{|\{i \in [\![1, n]\!] \ / \ \frac{1}{2} o_i \le y_i \le 2 o_i\}|}{n} \tag{3}$$

where $|\cdot|$ denotes the cardinal. A summary of Polyphemus models performances, with comparisons to other models, is shown in Table 1. In Fig. 2, a scatter plot of simulated
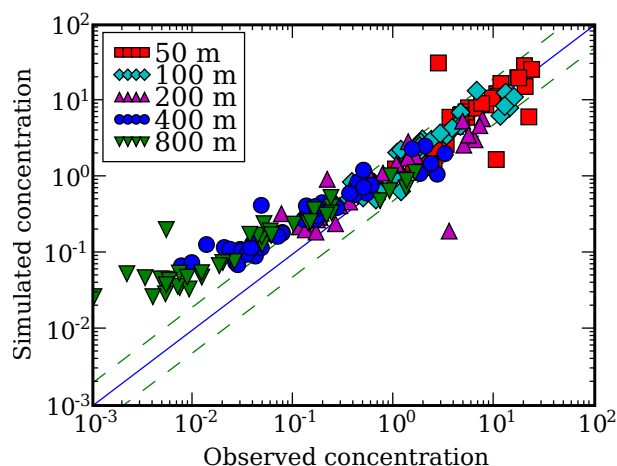
Fig. 2. Simulated concentrations versus measurements for maximum concentrations at 50 m, 100 m, 200 m, 400 m and 800 m from the source and for 43 Prairie Grass experiments. The simulated concentrations are computed with a dispersion parameterization derived from similarity theory.
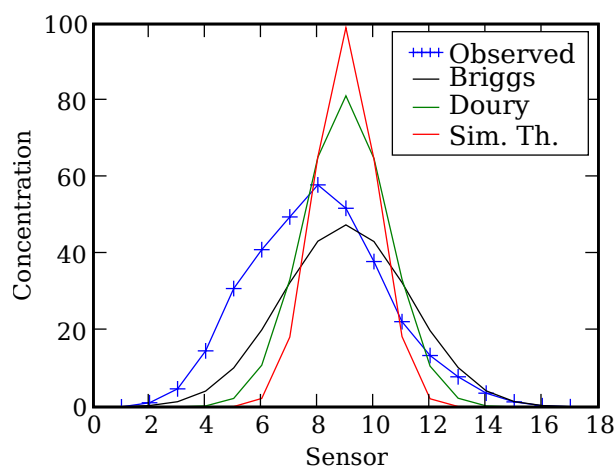

Fig. 3. Simulated concentrations and measurements at 100 m from the source and during one Prairie Grass experiment. The simulated concentrations are computed using Briggs formulae, Doury formula and a parameterization derived from similarity theory.

concentrations versus measurements illustrates the results for one model. The concentrations for one experiment are shown in Fig. 3.

4.2 Aerosol simulation

The chemistry-transport model Polair3D, together with the size-resolved aerosol module SIREAM, has been evaluated at continental scale over Europe (Sartelet et al., 2007). The evaluation simulations cover the year 2001, with 0.5 degree of horizontal resolution and with 5 levels up to 3000 m in the

Table 1. Performances of several Gaussian models with Prairie Grass observations ($mg\,m^{-3}\,g^{-1}\,s^{-1}$). The observations are the maximum concentrations at several distances from the point source (50 m, 100 m, 200 m, 400 m and 800 m) and for 43 experiments. The first comparisons are detailed in (McHugh et al., 1999). The comparisons for Polyphemus are performed with slightly different observations and with Briggs formulae, Doury formula and a physical formulation (based on similarity theory).

Model	Mean	FB	Corr	FAC2
Observations	2.14	0.000	1.000	1.000
ADMS	1.20	−0.566	0.641	0.456
AERMOD	2.14	0.000	0.749	0.759
ISCST3	2.01	−0.064	0.716	0.621
Observations	2.32	0.000	1.000	1.000
Polyphemus – Briggs	2.33	0.000	0.785	0.744
Polyphemus – Doury	1.46	−0.456	0.425	0.274
Polyphemus – Phys.	2.53	0.085	0.823	0.614

vertical. The configuration is roughly described in Sect. 3.1. The meteorological fields are provided by ECMWF. The photochemical mechanism is RACM.

The performances are evaluated for several pollutants. The measurements come from three networks: EMEP, BDQA (French network) and AirBase. Table 2 shows a summary of the error measures for O_3, NO_2, NH_3, HNO_3, SO_2, $PM_{2.5}$ and PM_{10}.

4.3 Ensemble forecast: multimodel approach

Polyphemus gathers several alternative physical parameterizations and can process data from different sources. In addition the transport models are quite flexible, in particular Polair3D: several chemical mechanisms and numerical schemes may be plugged in. Hence virtually all choices and components that define a model can be modified. This allows to build multimodel ensembles.

In Mallet and Sportisse (2006a) an ensemble of 48 models is built to simulate photochemistry over Europe during summer 2001. The models differ in their chemical mechanism (RACM, Stockwell et al., 1997, or RADM 2, Stockwell et al., 1990), their vertical diffusion parameterization, their emissions and other input data, their numerical approximations, ... The ensemble shows a wide spread, as illustrated by Fig. 4. In Fig. 5, the relative standard deviation of the ensemble estimates the spatial distribution of the uncertainty.

The ensemble samples the distribution of output concentrations. Of course, there are models (in distribution tails) less probable than others. Nonetheless several models are likely to perform the best forecast at a given location (see Fig. 6). One may want to improve the forecast through model selection or linear combination of the models. For instance, Polyphemus (AtmoPy) implements an optimal linear combi-
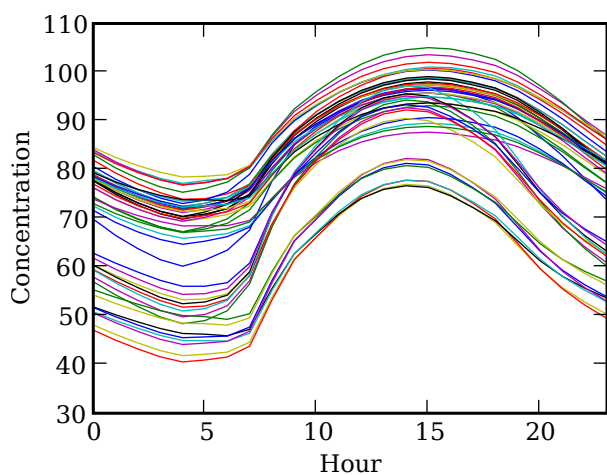
Fig. 4. Ozone daily profiles of 48 models built in Polyphemus. The concentrations are in $\mu g \, m^{-3}$ and are averaged over Europe (at ground level) and over summer 2001.
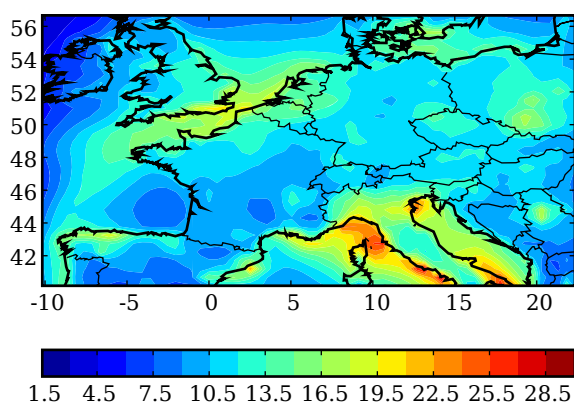

Fig. 5. Map of the relative standard deviation of the ensemble, that is, spatial distribution of the ensemble spread. The standard deviation of the ensemble is computed in each cell and for each hour. The resulting standard deviations are averaged (over summer 2001) in each cell and divided by the mean concentration of the cell, which gives a relative standard deviation.

nation based on least squares (superensemble) which enables to decrease the root mean square error of forecasts by over 10% (Mallet and Sportisse, 2006a).

4.4 Data assimilation

Data assimilation is a good example of the purpose of drivers. We illustrate four algorithms applied to the model Polair3D: optimal interpolation, two Kalman filters and 4-D variational assimilation.

In the first algorithm, each time new observations are available, a new state vector c is computed based on the new
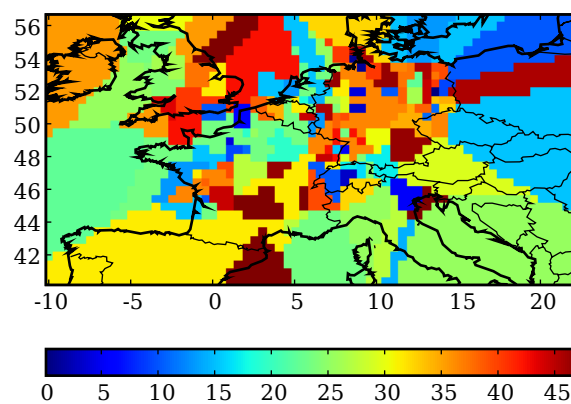

Fig. 6. Map of best model indices. In each cell of the domain, the color shows which model (marked with its index, in $[\![0, 47]\!]$) gives the best ozone peak forecast on 7 May 2001 at the closest station to the cell center. It shows that many models can deliver the best forecast at some point.

Table 2. Performances of Polair3D with SIREAM over Europe in 2001. The concentrations and the root mean square error are in $\mu g \, m^{-3}$. "Obs. Mean" refers to the measurements mean; "Sim. Mean" refers to the simulated mean concentrations; RMSE is the root mean square error; and Corr is the correlation. The comparisons are performed with daily peaks for O_3, hourly concentrations for NO_2 and daily averages for other species.

Species	Network	Obs. Mean	Sim. Mean	RMSE	Corr
O_3	EMEP	80.2	73.5	21.4	0.721
	AirBase	73.7	72.5	23.5	0.765
	BDQA	77.4	73.7	23.2	0.774
NO_2	EMEP	10.0	12.3	10.1	0.336
	AirBase	23.9	15.3	18.3	0.395
	BDQA	21.9	13.8	18.2	0.382
NH_3	EMEP	7.4	6.3	5.4	0.295
	AirBase	12.9	7.4	12.9	0.284
HNO_3	EMEP	0.7	1.3	1.4	0.265
SO_2	EMEP	2.0	5.2	4.8	0.475
	AirBase	6.4	6.9	6.5	0.445
	BDQA	7.8	6.6	6.4	0.364
PM_{10}	EMEP	16.9	15.6	12.6	0.545
	AirBase	24.9	15.4	16.6	0.440
	BDQA	19.8	15.8	9.6	0.570
$PM_{2.5}$	EMEP	12.6	8.4	8.6	0.541

observation vector o and the current state vector c^b (b stands for background):

$$c = c^b + PH^T (HPH^T + R)^{-1}(o - Hc^b) \quad (4)$$

where H is the observation operator that maps the state vector to the observation space, P is the covariance matrix of
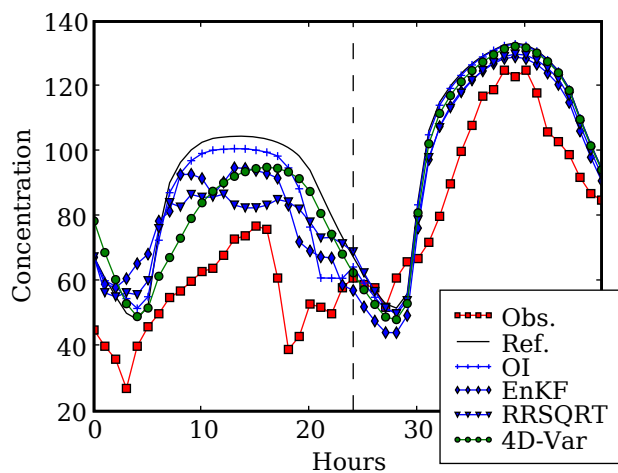
Fig. 7. Time evolution of ozone concentrations at EMEP station Montandon, for the reference simulation (without assimilation), the simulation with assimilation (optimal interpolation, OI; ensemble Kalman filter, EnKF; reduced-rank square root Kalman filter, RRSQRT; 4D-Var). The dotted vertical lines delimits the assimilation period.

background error and $\mathbf{R}$ is the (assumed diagonal) covariance matrix of observation error. In current implementation, the matrix $\mathbf{P}$ is either diagonal or in Balgovind form, that is, dependent on the distance r between two points:

$$f(r) = \left(1 + \frac{r}{L}\right)\exp\left(-\frac{r}{L}\right)v \qquad (5)$$

where L is a characteristic length and v is a variance.

In Kalman filters, the background error covariance matrix is estimated from an ensemble of models (ensemble Kalman filter) or is propagated in reduced form (reduced-rank square root filter).

The 4-D-variational data assimilation method minimizes the cost function

$$J(c_0) = (c_0 - c_0^b)^T \mathbf{B}^{-1}(c_0 - c_0^b) + \sum_i (o_i - \mathbf{H}c_i)^T \mathbf{R}^{-1}(o_i - \mathbf{H}c_i) \qquad (6)$$

with respect to the initial conditions c_0 whose background error covariance matrix is $\mathbf{B}$. The minimization includes observations o_i for several time steps i and the corresponding simulated concentrations c_i. The minimization is performed by L-BFGS (Byrd et al., 1995), using gradients of the cost function. These gradients are computed with the adjoint model of Polair3D.

All four algorithms are applied to a test-case over Europe in summer 2001. For instance, observations are assimilated for one day, and a subsequent forecast is performed during the next day. The time evolution of ozone concentration is shown at one EMEP station for the four algorithms in Fig. 7.

5 Conclusions

The modeling system Polyphemus is built with independent levels dedicated to (1) data processing and management, (2) physical parameterizations as base for model formulation (in preprocessing steps), (3) numerical solvers (that is, numerical models), and (4) high-level methods in which the numerical model is viewed as a function.

As a result of the developments in this structure, Polyphemus covers applications from local scale to continental scale. It includes a Gaussian plume model, a Gaussian puff model and the Eulerian chemistry-transport models Castor and Polair3D. These models have passive and radioactive versions. Depending on their complexity, they include photochemistry and aerosols (with or without dynamics). In addition, Polair3D has an adjoint version of its gas-phase part.

On top of the models, drivers are developed in order to embed the models in advanced methods. This is the case of a plume-in-grid driver which couples an Eulerian model with a Gaussian puff model. Four data assimilation algorithms, including variational assimilation, are also implemented in drivers – independently of the models.

The system remains open in its structure, with the ability to host other preprocessing or postprocessing tools and other models, and in its development policy since it is open source, available online and associated with many resources (primarily documentation) for users and developers.

All material (source code, documentation, training sessions) is available at http://cerea.enpc.fr/polyphemus/. The supplementary material to this paper includes Polyphemus source code (version 1.1.1, 2007-06-21) and its user's guide: http://www.atmos-chem-phys.net/7/5479/2007/acp-7-5479-2007-supplement.zip.

References

Arya, S. P.: Air pollution meteorology and dispersion, Oxford University Press, 1999.

Boutahar, J., Lacour, S., Mallet, V., Quélo, D., Roustan, Y., and Sportisse, B.: Development and validation of a fully modular platform for numerical modelling of air pollution: POLAIR, Int. J. Env. and Pollution, 22, 17–28, 2004.

Byrd, R. H., Lu, P., and Nocedal, J.: A limited memory algorithm for bound constrained optimization, SIAM J. on Sci. and Stat. Comp., 16, 1190–1208, 1995.

Chin, M., Rood, R. B., Lin, S.-J., Müller, J.-F., and Thompson, A. M.: Atmospheric sulfur cycle simulated in the global model GOCART: Model description and global properties, J. Geophys. Res., 105, 24 671–24 688, 2000.

Debry, E., Fahey, K., Sartelet, K., Sportisse, B., and Tombette, M.: Technical Note: A new SIze REsolved Aerosol Model (SIREAM), Atmos. Chem. Phys., 7, 1537–1547, 2007, http://www.atmos-chem-phys.net/7/1537/2007/.

Doury, A.: Une méthode de calcul pratique et générale pour la prévision numérique des pollutions véhiculées par l'atmosphère, Tech. Rep. 4280, CEA, France, 1976.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res., 99, 10 143–10 162, 1994.

Faure, C. and Papegay, Y.: O∂yssée user's guide – version 1.7, Tech. Rep. 0224, INRIA, 1998.

Heemink, A. W., Verlaan, M., and Segers, A. J.: Variance reduced ensemble Kalman filtering, Mon. Wea. Rev., 129, 1718–1728, 2001.

Horowitz, L. W., Walters, S., Mauzerall, D. L., Emmons, L. K., Rasch, P. J., Granier, C., Tie, X., Lamarque, J.-F., Schultz, M. G., Tyndall, G. S., Orlando, J. J., and Brasseur, G. P.: A global simulation of tropospheric ozone and related tracers: description and evaluation of MOZART, version 2, J. Geophys. Res., 108(D24), doi:10.1029/2002JD002853, 2003.

Korsakissok, I.: Performance evaluation of Polyphemus Gaussian models with Prairie Grass experiment, Tech. Rep. 2007-15, CEREA, 2007.

Krishnamurti, T. N., Kishtawal, C. M., Zhang, Z., T. LaRow, D. B., and Williford, E.: Multimodel ensemble forecasts for weather and seasonal climate, J. Climate, 13, 4196–4216, 2000.

Le Dimet, F.-X. and Talagrand, O.: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects, Tellus, 38A, 97–110, 1986.

Louis, J.-F.: A parametric model of vertical eddy fluxes in the atmosphere, Boundary-Layer Meteor., 17, 187–202, 1979.

Mallet, V.: Polair3D technical documentation, Tech. Rep. 2006-3, CEREA, 2006.

Mallet, V. and Sportisse, B.: 3-D chemistry-transport model Polair: numerical issues, validation and automatic-differentiation strategy, Atmos. Chem. Phys. Discuss., 4, 1371–1392, 2004, http://www.atmos-chem-phys-discuss.net/4/1371/2004/.

Mallet, V. and Sportisse, B.: A comprehensive study of ozone sensitivity with respect to emissions over Europe with a chemistry-transport model, J. Geophys. Res., 110(D22), doi:10.1029/2005JD006234, 2005a.

Mallet, V. and Sportisse, B.: Data processing and parameterizations in atmospheric chemistry and physics: the AtmoData library, Tech. Rep. 2005-12, CEREA, 2005b.

Mallet, V. and Sportisse, B.: Ensemble-based air quality forecasts: A multimodel approach applied to ozone, J. Geophys. Res., 111(D18), doi:10.1029/2005JD006675, 2006a.

Mallet, V. and Sportisse, B.: Uncertainty in a chemistry-transport model due to physical parameterizations and numerical approximations: An ensemble approach applied to ozone modeling, J. Geophys. Res., 111(D1), doi:10.1029/2005JD006149, 2006b.

Mallet, V., Quélo, D., and Sportisse, B.: Software architecture of an ideal modeling platform in air quality – A first step: Polyphemus, Tech. Rep. 2005-11, CEREA, 2005.

McHugh, C. A., Carruthers, D. J., Higson, H., and Dyster, S. J.: Comparison of model evaluation methodologies with application to ADMS 3 and US models, 6th international conference on harmonisation within atmospheric dispersion modelling for regulatory purposes, Rouen, France, 1999.

Middleton, P., Stockwell, W. R., and Carter, W. P. L.: Aggregation and analysis of volatile organic compound emissions for regional modeling, Atmos. Env., 24A, 1107–1133, 1990.

Njomgang, H., Mallet, V., and Musson-Genon, L.: AtmoData scientific documentation, Tech. Rep. 2005-10, CEREA, 2005.

Quélo, D., Krysta, M., Bocquet, M., Isnard, O., Minier, Y., and Sportisse, B.: Validation of the Polyphemus platform: the ETEX, Chernobyl and Algeciras cases, Atmos. Env., 41, 5300–5315, doi:10.1016/j.atmosenv.2007.02.035, 2007.

Sartelet, K. N., Debry, E., Fahey, K., Roustan, Y., Tombette, M., and Sportisse, B.: Simulation of aerosols and related species over Europe with the Polyphemus system. Part I: model-to-data comparison for 2001, Atmos. Env., 41, 6116–6131, doi:10.1016/j.atmosenv.2007.04.024, 2007.

Schmidt, H., Derognat, C., Vautard, R., and Beekmann, M.: A comparison of simulated and observed ozone mixing ratios for the summer of 1998 in Western Europe, Atmos. Env., 35, 6277–6297, 2001.

Simpson, D., Winiwarter, W., Börjesson, G., Cinderby, S., Ferreiro, A., Guenther, A., Hewitt, C. N., Janson, R., Khalil, M. A. K., Owen, S., Pierce, T. E., Puxbaum, H., Shearer, M., Skiba, U., Steinbrecher, R., Tarrasón, L., and Öquist, M. G.: Inventorying emissions from nature in Europe, J. Geophys. Res., 104, 8113–8152, 1999.

Stockwell, W. R., Middleton, P., Chang, J. S., and Tang, X.: The second generation regional acid deposition model chemical mechanism for regional air quality modeling, J. Geophys. Res., 95, 16 343–16 367, 1990.

Stockwell, W. R., Kirchner, F., Kuhn, M., and Seefeld, S.: A new mechanism for regional atmospheric chemistry modeling, J. Geophys. Res., 102, 25 847–25 879, 1997.

Troen, I. and Mahrt, L.: A simple model of the atmospheric boundary layer; sensitivity to surface evaporation, Boundary-Layer Meteor., 37, 129–148, 1986.

Turner, D. B.: Workbook of atmospheric diffusion estimates, Tech. Rep. 999-AP-26, US EPA, Washington, D.C., 1969.

Underwood, B.: Review of deposition velocity and washout coefficient, Tech. Rep., AEA Technology, Harwell, 2001.

US EPA: Guideline for regulatory application of the urban airshed model, Tech. Rep. EPA-450/4-91-013, US EPA, 1991.

Verwer, J. G., Hundsdorfer, W., and Blom, J. G.: Numerical time integration for air pollution models, Surveys on Math. for Indus., 10, 107–174, 2002.

Zhang, L., Gong, S., Padro, J., and Barrie, L.: A size-segregated particle dry deposition scheme for an atmospheric aerosol module, Atmos. Env., 35, 549–560, 2001.

Zhang, L., Brook, J. R., and Vet, R.: A revised parameterization for gaseous dry deposition in air-quality models, Atmos. Chem. Phys., 3, 2067–2082, 2003, http://www.atmos-chem-phys.net/3/2067/2003/.