GIS-based atmospheric dispersion modelling

Forecasting the pesticide atmospheric spray drift from a vineyard plot to a watershed

Nicolas Bozon · Bijan Mohammadi

Received: 21 October 2008 / Accepted: 5 May 2009 / Published online: 10 June 2009
© Società Italiana di Fotogrammetria e Topografia (SIFET) 2009

Abstract A multileveled, low-complexity model for flow field reconstruction and transport of passive scalars within an open-source geographic information system (GIS) is presented. Similitude solutions are used in a nonsymmetric metric for the transport, and the model parameters' identification is based on data assimilation. The approach does not require the solution of any partial differential equations and, therefore, is mesh-free. The model also permits to access the solution in one point without computing the solution over the whole domain. A multilevel algorithm permits to locally add partially available information. The model is coupled with GIS in order to handle both realistic topographies using digital elevation models and a spatial agrometeorological database. Numerical results and georeferenced outputs are illustrating the application of the model to various configurations over realistic topographies, and it can be used as a functional basis for atmospheric pollution prediction and spatial risk analysis.

Keywords Atmospheric dispersion · Reduced-order modelling · Nonsymmetric geometry · Multilevel · Open source GIS · Digital elevation model · Pollution risks prediction

Introduction

Let us consider environmental multiscale phenomena presented by the atmospheric dispersion process. These are common in industrial pollutant emissions, fire smoke transport, and agricultural pesticide spray drift (see Fig. 1). The latter is part of our research interests, and this paper focuses on pesticide spray drift modelling over vineyards, using mathematical modelling and geographical information systems (GIS). Our aim is to provide some georeferenced atmospheric transport simulation models at very low calculation cost using reduced-order modelling and open-source GIS. This is necessary to make affordable assimilation-simulation and very helpful to perform atmospheric pollution risk analysis for viticulture applications.

Atmospheric dispersion equations are based on parameters that are strongly affected by spatial and temporal variations (Markiewicz 2006). These variations significantly affect the spray drift behavior and trajectory. The spatial dimension is thus essential in atmospheric dispersion modelling, but also represents the GIS's paradigm (Dragosits et al. 1996). GIS have so become an adequate tool to analyze and visualize spatial-based environmental models (Karimi and Houston 1997). The coupling of the model with such technologies is presented in this paper, regarding the use of digital elevation models (DEM), the role that

N. Bozon (✉)
CEMAGREF of Montpellier in the Protech Research
Team - UMR ITAP, CEMAGREF of Montpellier,
34196 Montpellier, France
e-mail: nicolas.bozon@gmail.com

B. Mohammadi
Research Mathematics and Modelling Institute - UMR 5149,
University of Sciences of Montpellier 2,
34095 Montpellier, France
e-mail: bijan.mohammadi@univ-montp2.fr

Fig. 1 Early morning pesticide cloud observed at Neffies (34) near vineyards

GIS plays in the enhancement of the former model, and the computational aspects.

The optimization of the mathematical model regarding the spatial dimension first appears as necessary as we wish to couple a low-complexity model with the landscape reality provided by GIS. This work implies establishing links between the spray drift model and GIS algorithms, by selecting the most significant geospatial parameters that can enhance the calculation. Topography effects and scale variations have been naturally chosen as we need to get an optimized z dimension, in order to implement more reality in the drift phenomenon calculation. Then, a second side of the coupling consists in integrating the model within a GIS environment with the aim to manage and automate the model inputs and to promote its outputs using cartographic rendering.

In a previous work, we have treated the problem of agricultural pesticide emission and dispersion by coupling local (i.e., emission and near-field distribution) and global (i.e., transport over large distance) models (Brun 2006; Mohammadi et al. 2006). In this approach, the local model provides the inlet conditions for the levels above. The dependency between levels is a major asset to avoid the solution of partial differential equations, using model reduction. This is based on adapting search spaces for the solution of a given model using a priori information. More precisely, a near field (to the tractor's injection device) search space is built using experimental observations. Once this local solution is known, the amount of specie leaving the atmospheric sublayer is evaluated using analytical integration of the governing equations (Mohammadi et al. 2006; Brun and Mohammadi 2006). A priori local information is

once again included during this analytical solution, looking for special solutions. The resulting quantity is then considered as a candidate for transport over long distances. Similitude solutions are then used for mixing layers and plumes (Simpson 1997) and generalized in a nonsymmetric travel-time-based metric, with the aim of accounting for general windflow fields. The introduction of this new metric involves, first, a construction step for the flow field from observation data. The constructed flow field must be divergence-free, and transported quantities have to verify some constraints such as conservation, positivity, and linearity of solutions.

The goal of the present work is, first, to improve the existing long-range transport model, and then to account for nonuniform topographies to get more realistic simulations using GIS. We also want to be able to locally optimize the calculation accuracy according to scales variations by using nested numerical zoom and modifying the wind flow field construction algorithm. Thus, two main aspects are presented in this paper. The optimization of the dispersion model according to topography and scale changes is first presented using mathematical modelling. The coupling with GIS is then detailed in a second part, regarding the model integration and the main computational tasks.

Reduced-order modelling

Introduction

Let us first formally define what we call reduced-order modelling. Consider the calculation of a state variable:

$$F(V(p)) = 0 \quad (1)$$

$V(p)$ is function of independent variable p . Our aim is to define a suitable search space for the solution $V(p)$ instead of considering a general function space.

This former approach corresponds to the finite-element methods, for instance, when we look for a solution V_h expressed in some finite-dimensional subspace $S_N(\{W_i, i = 1, \dots, N\})$:

$$V_h = \sum_{i=1}^N v_i^{\text{opt}} W_i, \quad (2)$$

S_N is generated by the chosen functional basis $\{W_i, i = 1, \dots, N\}$. For instance, we can consider W_i as polynomial of degree one ($W_i \in P^1$) on each element of a discrete domain called a “mesh.” v_i^{opt} denotes the

values of the solution on the nodes of the mesh solution of a minimization problem:

$$v_i^{\text{opt}} = \operatorname{argmin}_{v_i} \|V - V_h\|_F, \quad i = 1, \dots, N \quad (3)$$

Hence, V_h is the projection of $V(p)$ over S_N . $\|\cdot\|_F$ is a norm involving the state equation. In this approach, the quality of the solution is monitored either through the mesh size (i.e., $N \rightarrow \infty$) and/or the order of the finite element (i.e., $W_i \in P^m$ with m increasing for higher accuracy) (Ciarlet 1978). If the approach is consistent, the projected solution tends to the exact solution when $N \rightarrow \infty$ or $m \rightarrow \infty$. In all cases, the size of the problem is large $1 \ll N < \infty$.

In a low-complexity approach, we approximate $V(p)$ by its projection $\tilde{V}$ over a subspace $\tilde{S}_n(\{w_i, i = 1, \dots, n\})$ not generated by polynomial functions anymore. We rather consider $\{w_i, i = 1, \dots, n\}$ as a family of solutions (“snapshots”) of the initial full model ($p \rightarrow V(p)$):

$$\tilde{V} = \sum_{i=1}^n \tilde{v}_i^{\text{opt}} w_i \quad (4)$$

$$v_i^{\text{opt}} = \operatorname{argmin}_{v_i} \|V - \tilde{V}\|_F, \quad i = 1, \dots, n \quad (5)$$

The cost of the two methods depends on the cost of the solution of the minimization problems. For the reduced-order approach to be efficient, we aim, therefore, $n \ll N$ (Veroy and Patera 2005). This is only possible if the w_i family is well suited to the problem, in which case we can also expect a more accurate solution despite the small size of the problem.

In this work, we would like to adopt this reduced-order modelling approach with, however, removing the calculation of the snapshots w_i . Indeed, these calculations are not always easy and affordable especially when the original model involves a nonlinear partial differential equation. We would like, rather, to take advantage of what we know of the physics of the problem and replace the direct model $F(V(p)) = 0$ with an approximate model $f(V(p)) = 0$, for which snapshots are either analytically known or, at least, easier to evaluate. This can also be seen as changing the norm in the minimization problem above. Below, we will take advantage of this redefinition of the topology in the solution of our problem.

Considering a simple state equation is a natural way to proceed, as we do not need all the details on a given state. Also, it is sufficient for the low-complexity model to have a local validity domain: we do not necessarily use the same low-complexity model over the whole range of the parameters. However, this brings in the question of region of confidence for a given reduced-order model for which we need to know the

validity domain. We have used this approach in the incomplete sensitivity concept where the linearization of a functional is performed not for the full model but for an approximate state equation (Mohammadi and Pironneau 2001).

Transport and nonsymmetric geometry

We consider the situation of a source releasing a time-dependent quantity $c_{\text{inj}}(t)$ in the atmosphere at a given location. One aims to develop a low-complexity model to represent the dispersion of c_{inj} . The primary factors influencing the dispersion of a neutral plume are advection by the wind and turbulent mixing. The simplest model of this process is to assume that the plume advects downwind and spreads out in the horizontal and vertical directions. Hence, the distribution of a passive scalar c , emitted from a given point and transported by a uniform plane flow filed U along x coordinate, can be represented by:

$$c(x, y, z) = c_c(x) f\left(\sqrt{y^2 + z^2}, \delta(x)\right), \quad (6)$$

where

$$c_c(x) \sim \exp(-a(U)x) \quad (7)$$

and

$$f\left(\sqrt{y^2 + z^2}, \delta(x)\right) \sim \exp\left(-b(U, \delta(x))\sqrt{y^2 + z^2}\right) \quad (8)$$

c_c is the behavior along the central axis of the distribution and $\delta(x)$ characterizes the thickness of the distribution at a given x coordinate. An analogy exists with plane or axisymmetric mixing layers and neutral plumes where δ is parabolic for a laminar jet and linear in turbulent cases (Cousteix 1989; Simpson 1997). $a(\cdot)$ is a positive monotonic decreasing function and $b(\cdot, \cdot)$ is positive, monotonic increasing in U and decreasing in δ . In a uniform atmospheric flow field, this solution can be used for the transport of c^+ above. We would like to generalize this solution in a nonsymmetric metric defined by migration times based on the flow field and, hence, treat the case of variable flow fields.

Nonsymmetric geometry

In a symmetric geometry, the distance function between two points A and B verifies

$$d(A, B) = 0 \Rightarrow A = B \quad (9)$$

$$d(A, B) = d(B, A) \quad (10)$$

$$d(A, B) \leq d(A, C) + d(C, B) \quad (11)$$

However, the distance function can be nonuniform with anisotropy (the unit spheres being ellipsoids). In a chosen metric $\mathcal{M}$, the distance between A and B is given by:

$$d_{\mathcal{M}}(AB) = \int_0^1 \left(\overrightarrow{AB} \mathcal{M}(A + t\overrightarrow{AB}) \overrightarrow{AB} \right)^{1/2} dt, \quad (12)$$

where $\mathcal{M}$ is positive definite and symmetric in symmetric geometries. With $\mathcal{M} = I$, one recovers the Euclidean geometry and variable $\mathcal{M}$ permits to account for anisotropy and nonuniformity of the distance function. A nationwide driving travel time map is an example of an anisotropic geometric representation of a country. However, the geometry is still symmetric (i.e., one supposes it takes the same time to drive from A to B than from B to A) and relations hold.¹

The mentioned symmetry is not natural for many applications. Considering again the example of the driving map, one experiences that, everyday, in rush hours, driving from A to B is not equivalent to driving from B to A . We would like therefore to go one step further considering nonsymmetric geometries.

Consider now the following distance function definition:

If A is upwind with respect to B , then

$$d(B, A) = \infty \quad \text{and} \quad d(A, B) = \int_A^{B^\perp} ds/u = T_{AB} \quad (13)$$

T_{AB} is the migration time from A to $B^\perp$ along the characteristic passing by A . u is the local velocity along this characteristic and is, by definition, tangent to the characteristic. $B^\perp$ denotes the projection of B over this characteristic in the Euclidean metric. To guarantee nondegeneracy of d (i.e., $d(A, B) = 0 \Rightarrow A = B$), we require $B^\perp \neq A$. We suppose that the characteristic issued from A is unique, hence, avoiding sources and attraction points in the flow field. In case of nonuniqueness of this projection, we choose the direction of the projection, which satisfies best the constraint $(\vec{u} \cdot \nabla c = 0)$ in B . Finally, we define A being upwind with respect to B if there exists no $B^\perp$.

¹This approach is also suitable for adaptive sampling and mesh adaptation (Alauzet et al. 2002; Hecht and Mohammadi 1997; Borouchaki et al. 1997). Linking the metric to the Hessian of the variable, the metric permits to equidistribute the interpolation error over a given sampling or mesh and, for this former monitor, the quality of the solution.

Fig. 2 Poiseuille flow in travel-time metric. Triangular inequality does not hold in the travel-time-based metric

This definition of distance does not verify the triangular inequality. The inequality holds if C is upwind with respect to A or if B is upwind with respect to C . Otherwise, we can find counter examples. For instance, consider a Poiseuille flow in a channel (see Fig. 2) with A and B at the same distance of the same wall and close to it (therefore, $B^\perp = B$) and C in the middle of the channel downstream with respect to A and upstream to B (for instance, $x_C = x_{(A+B)/2}$). We can see that the flow is faster in the middle of the channel

$$\int_{C^\perp}^B ds/u \geq \int_C^{B^\perp} ds/u, \quad (14)$$

and therefore,

$$\int_A^B ds/u \geq \int_A^{C^\perp} ds/u + \int_C^{B^\perp} ds/u \quad (15)$$

This corresponds to the intuition that a longer way might be faster (see Fig. 2). Now, we would like to use this positive function, which we improperly call distance, to define the solution of transport by variable flow fields (Fig. 3).

Generalized plume solution

By analogy with the transport by a uniform flow, we assume that the distribution of a passive scalar transported by a variable flow field $\vec{u}$ can be written as:

$$c(x, y, z) = c_c(d) f(d_E^\perp, \delta(d)), \quad (16)$$

where $d_E^\perp$ is the Euclidean distance in the normal direction local to the characteristic at $B^\perp$ along direction $BB^\perp$ (see Fig. 4).

Flow field

We should keep in mind that realistic configurations provide, most of the time, very poor information on the atmospheric flow details, compared to the accuracy we would like to obtain for the transport. As an example, the flow will be described probably by less than one point by several square kilometers. We consider that

Fig. 3 Examples of symmetric Euclidian and nonsymmetric travel time-based distances for a rotating wind (top)

the ground flow field is built from observation data as a solution of the following system:

$$\vec{u}_H = -\nabla\phi, \quad -\Delta\phi = 0, \quad (17)$$

under the constraint that

$$-\nabla\phi(x_j) = \vec{u}_{\text{obs}}(x_j), \quad j = 1, \dots, n_{\text{obs}}, \quad (18)$$

where ϕ is a 2D scalar potential and n_{obs} is the number of observation points. One particularity of the present application is that the number of observations is small and that the distance between two observation points is large. The observations are close to the ground at $z = H$, and this construction gives a map of the flow near the ground. If we assume $\vec{u}_{\text{obs}}$ is divergence-free, the solution to (see Eq. 17) at a point x can be seen as

$$\vec{u}_H(x) \sim \vec{u}_H(x) = \sum_{j=1}^{n_{\text{obs}}} \lambda_j(x) \vec{u}_{\text{obs}}(x_j), \quad 0 \leq \lambda_j(x) \leq 1, \quad (19)$$

where $\lambda_j(x)$ are barycentric functions such as

$$\sum_{j=1}^{n_{\text{obs}}} \lambda_j(x) = 1, \quad \text{and} \quad \lambda_j(x_i) = \delta_{ij} \quad (20)$$

In order to account for error in measurements $\vec{u}_{\text{obs}}$, a Kriging construction can also be used (Krige 1951; Chiles and Delfiner 1999).² The main reasons to avoid numerical solving of the partial differential equation are that we need to use a mesh-free technique, but also because available information is poor (making numerical solution unrealistic) and that we can observe noise in measurements.

The plane velocity map $\vec{u}_H$ can be completed in the vertical direction using generalized wall functions (Mohammadi and Pironneau 1994; Mohammadi and Puigt 1998). These can be written as:

$$(\vec{u} \cdot \vec{\tau})^+ = (\vec{u} \cdot \vec{\tau})/u_\tau = f(z^+) = f(zu_\tau/\nu), \quad (21)$$

²If we assume $\vec{u}_H$ to behave as a stochastic variable, and supposing its mean and covariance being those of $\vec{u}_{\text{obs}}$, the Kriging predictor $\vec{u}_H$ minimizes the variance of the prediction error: $\varepsilon = \vec{u}_H - \vec{u}_{\text{obs}}$

Fig. 4 Two-dimensional sketch of plume models in a cartesian metric for a uniform flow field (*top*) and in a travel-time based metric for a rotating flow field (*bottom*)

where $\vec{\tau} = \vec{u}_H / \|\vec{u}_H\|$ is the local tangent unit vector to the ground in the direction of the flow and we assume that $(\vec{u} \cdot \vec{n}(z = H) = 0)$ if $\vec{n}$ is normal to the ground ($\vec{n} = (0, 0, -1)$) (with no topography variations).

This is a nonlinear equation that provides the friction velocity, u_τ , knowing $(\vec{u} \cdot \vec{\tau})_H$. It is used to define the horizontal velocity $\vec{u} \cdot \vec{\tau} = u_\tau f(z^+)$ for $z > H$. This

construction gives two components of the flow, and the divergence-free condition implies that the third component is constant and, therefore, vanishes as it is supposed as zero at $z = H$. This construction could be improved, but we find it sufficient for the required level of accuracy. For ground variations modelling, the flow will be locally rotated to remain parallel to the ground (see Fig. 5).

Calculation of migration times

As we said, our approach aims to provide the solution at a given point without calculating the whole solution. Being in point B , we need an estimation of the migration time from the source in A to B using the construction in “Flow field.”

The construction of characteristics is avoided using an iterative polynomial definition for a characteristic $s(t) = (x(t), y(t), z(t))$, $t \in [0, 1]$, starting from a third-order polynomial function verifying for each coordinate:

$$P_n(0) = x_A, P_n(1) = x_B, P'_n(0) = u_A^1, P'_n(1) = u_B^1 \quad (22)$$

(same for y and z). If $P'_n(\zeta) \neq u^1(x = P_n(\zeta))$, then this new point should be assimilated by the construction increasing the polynomial order by one. $\zeta \in]0, 1[$ is chosen randomly.

The migration time is computed over this polynomial approximation of the characteristic. Here, we make the approximation $B^\perp = B$, which means that the characteristic passing by A passes exactly by B , which is unlikely. In a uniform flow, this means we suppose that the angle between the central axis and AB is small (cosine near 1). We introduce, therefore, a correction factor of $2/3 = 0.636$ on the calculated times. This is

Fig. 5 Left: a typical DEM (x and y coordinates range over 2 km). Atmospheric dispersion in a uniform north wind with (*middle*) and without (*right*) the DEM

the stochastic averaged cosine value for a white noise for angles included between 0 and π .

Once d is calculated by this procedure, we need to define $d_E^\perp$, which is unknown, as $B^\perp$ is also unknown. We propose the following approximation $d_E^\perp \sim d_E(B, B^*)$, where B^* is the projection of B over the vector $\vec{u}$, which corresponds to the averaged velocity along the polynomial characteristic. This approach gives satisfactory results for smooth atmospheric flow fields (see Fig. 3), which is our domain of interest, as agricultural treatments are avoided when the wind is too strong, in order to limit the spray drift (e.g., for winds stronger than 20 km/h). This is also why the polynomial construction above gives satisfaction on low-order polynomials.

Multilevel construction

In realistic configurations, simulation needs be carried out over several hundred square kilometers in domains. At the same time, we need to be able to account for local topography variations with details provided every few meters. We saw previously that wind measurements are, most of the time, available on very coarse grids, with only two measurement points being usually distant for several kilometers. These constraints make it unrealistic and inefficient to perform the whole simulation with a “metric” topographic accuracy. Rather, we would like to somehow account for large-scale variations of topography on a coarse level simulation and gradually include the details of the ground variations near the main points of interest. To perform this task, we recursively apply the modelling described above on a cascade of embedded rectangular homothetic domains $\omega^i, i = 0, \dots$, with $\omega^0 = \Omega$ the full domain.³ Figure 7 shows a sketch of this construction where information is transferred from coarse to fine levels on corners, as shown in Fig. 6.

No information is transferred at this time from fine to coarse. Indeed, we emphasize that the grids correspond to the locations where topographic data are available. As we mentioned, our approach is mesh-free in the sense that no meshes are used for calculation. Only evaluated information on wind and concentration are stored at these locations. The total wind field is expressed as:

$$\vec{U}_H = \sum_{i=0}^{n_{\text{level}}} \vec{u}^i \chi(\omega^i), \quad (23)$$

³For the sake of simplicity, and also because this is rich enough for environmental applications, we deliberately limit ourselves to rectangular configurations.

where $\vec{u}^0 = \vec{u}_H$ is calculated in Eq. 17 for the coarser level and $\chi(\omega^i)$ is the characteristic function for the subdomain on which level i is defined. In other words, the correction is equal to zero outside ω^i . n_{level} is the total number of levels used. For $i > 1$, velocity restriction from level $i - 1$ to i is evaluated using Eq. 17, with the observation point being the information at the four corners q_j of a rectangle, as described below (see Figs. 6 and 7):

$$\vec{u}^i = -\nabla \phi^i, \quad -\Delta \phi^i = 0, \quad \phi^i(q_j) = \phi^{i-1}(q_j), \quad j = 1, \dots, 4 \quad (24)$$

Once again, we can take advantage of the linearity of the operator to use a similar decomposition for $\vec{u}^i$ as for $\vec{u}^0$, where the observation quantities become the values at the corners of the homothetic restriction:

$$\vec{u}^i(x) = \sum_{j=1}^4 \lambda_j(x) \vec{u}^{i-1}(q_j), \quad (25)$$

where

$$\sum_{j=1}^4 \lambda_j(x) = 1, \quad \text{and} \quad \lambda_j(x_i) = \delta_{ij}, \quad i, j = 1, \dots, 4 \quad (26)$$

If $\vec{u}^{i-1}$ is divergence-free, then this construction guarantees that

$$\int_{\partial \omega^i} \vec{u}^i \cdot \vec{n}^i dS = \int_{\partial \omega^i} \vec{u}^{i-1}|_{\partial \omega^i} \cdot \vec{n}^i dS = 0$$

Hence, the velocity restriction in ω^i remains divergence-free and is compatible with the overall field. In the simulation presented here, three levels have been used to link $\Omega = \omega^0 \sim 10 \text{ km}^2$ to the $\omega^2 \sim 10 \text{ m}^2$.

Once velocity restriction is defined, the concentration restriction is defined as follows:

$$c^i = \sum_{j=1}^4 \lambda_j(x) c^{i-1}(q_j) \quad (27)$$

Fig. 6 Two homothetic successive levels for multileveled construction. The smaller extent comes from the coarser level

Fig. 7 Example of three-level construction of concentration distribution. The *upper picture* shows the hierarchical constructed velocity fields on the three levels and the two points where velocity measurements were provided

This construction guarantees that the total mass in ω^i fits the entering quantity:⁴

$$\int_{\omega^i} c^i dV = \int_{\partial\omega^i} c^i \left(\frac{\vec{u}^i}{\|\vec{u}^i\|} \cdot \vec{n}^i \right) dS \quad (28)$$

In practice, we use the same sampling for each level (see Fig. 6). Typically, we use a 20×20 grid, which means that topography needs to be defined every 500 m at coarse level and every 50 cm at the finest level. This is interesting as it permits providing a more locally detailed DEM.

⁴The four-points trapezoidal rule is exact for numerical integration of bilinear functions $\int_{\omega} f(x, y) dx dy = \frac{1}{4} \sum_{j=1}^4 f(q_j)$.

Integral data

Once the species distribution $c(x, y, z)$ is found and the total injected quantity in time interval $[0, T]$ is known, we can assume:

$$K = \int_0^T c_{inj}(t) dt \quad (29)$$

Various quantities can be computed. For instance, we can have an estimation of the amount of species that has reached the ground using:

$$C_g(x, y) = \int_{z \leq z_0} c(x, y, z) dz \quad (30)$$

or estimate the quantity still in the atmosphere beyond a distance R_0 from the source, using:

$$C_a = K - \int_{R \geq R_0} C_g(x, y) dV \quad (31)$$

$R = \sqrt{x^2 + y^2}$ corresponds to the radius from source.

In the same way, time evolution of concentrations can be analyzed. Indeed, the following definition of the distance (Eq. 13) permits us to access the concentration distribution at time τ : *If A is upwind with respect to B, then*

$$d(B, A) = \infty \quad \text{and} \quad d(A, B) = \min(\tau, T_{AB}) \quad (32)$$

with T_{AB} defined in Eq. 13. Hence, one can realize snapshots of the concentration distribution evolution

Fig. 8 Snapshots of the concentration distribution evolution in time

in time as shown in Fig. 8. This simplified time integration allows us to animate the plume representation using proprietary GIS advanced functionalities, such as ESRI ArcScene ©. Indeed, as specified in Shephard et al. (2006), we can convert the resulting snapshots into several raster layers, load them in the ArcScene animation module, and easily generate an avi or vrml video directly from GIS.

Multilevel correction for ground variations

Multilevel correction

At this point, we would like to account for the topography or ground variations $((x, y) \rightarrow \psi(x, y))$ in the prediction model above. These are available from DEM (ArcGIS 2006). Despite the fact that this plays an important role in the dispersion process, it is obviously hopeless to launch direct simulations using a CFD model, based on a detailed ground description. We should mention that ground variation effects are implicitly present in observation data for wind as mentioned in “Flow field.” However, as we said, wind observations are quite incomplete. In particular, wind measurements are available every few kilometers while topographic data are available on a metric basis.

At each level i of the construction (see Fig. 9), we introduce a correction $\vec{u}_t^i$ to the restriction

$$\vec{U}_H = \sum_{i=0}^{n_{\text{level}}} (\vec{u}^i + \vec{u}_t^i) \chi(\omega^i) \quad (33)$$

Fig. 9 Example of three-level construction of a flow field, generated by the model and loaded into GIS as a vectorial layer

Fig. 10 Sketch of topography variation and normal definitions

Various local modelling can be considered for $\vec{u}_t^i$ going from simple algebraic expressions to more sophisticated local CFD models. We propose the following correction⁵:

$$\vec{u}_t^i = -\frac{1}{\rho} \text{sgn}(U_t) \nabla p^i, \quad p^i = p_r^{i-1} U_t^2 \quad (34)$$

where

$$U_t = \frac{\vec{u}^{i-1}}{\|\vec{u}^{i-1}\|} \cdot \vec{n}_t^i \quad \text{and} \quad p_r^{i-1} = \frac{1}{2} \rho (\vec{u}^{i-1} \cdot \vec{n}_t^i)^2 \quad (35)$$

ρ is the density of the fluid. p_r is a local pressure reference based on averaged entering velocity into sub-domain i :

$$(\vec{u}^{i-1} \cdot \vec{n}_t^i)_- = \frac{1}{n_-} \sum_{j=0}^4 \min(0, \vec{u}^{i-1}(q_j) \cdot \vec{n}_t^i) \quad (36)$$

$1 \leq n_- < 4$ is the number of entering flow corners. The normal to the ground evaluated from the digital terrain model restriction at level i is denoted by $\vec{n}_t^i$. This is different from the normal $\vec{n}^i$ to subdomain i . In the absence of ground variations, the two normals are orthogonal (see Fig. 10).

In case the topography is not constant, we have

$$\vec{u}^{i-1} \cdot \vec{n}_t^{i-1} = 0, \quad \text{but} \quad \vec{u}^{i-1} \cdot \vec{n}_t^i \neq 0 \quad (37)$$

This multilevel correction improves the predictive capacity of the model introducing a dependency between ground variations and migration time. However, this is not sufficient to correctly account for ground variations in dispersion. For instance, it is clear that, even in a

⁵This is the Bernoulli–Newton formula widely used in aeronautics and reproducing well the pressure distribution over a cylinder for a potential flow.

Fig. 11 Sketch of topography variation and nonsymmetry in cross-definition for a constant velocity field

uniform flow, cross diffusion is not symmetric on a sloppy ground when dispersion is performed parallel to the iso-level contours (see Fig. 11).

Hence, we also need to correct the functions a and b appearing in the dispersion modelling. As we have assumed that the construction is only coarse to fine without feedback from fine to coarse levels, we assume the correction conservative in the sense that the incoming mass into subdomain i ,

$$K^i = \left(c^{i-1} \frac{\vec{u}^{i-1}}{\|\vec{u}^{i-1}\|} \cdot \vec{n}^i \right)_{-} = \frac{1}{n_{-}} \sum_{j=0}^4 \min \left(0, c^{i-1}(q_j) \frac{\vec{u}^{i-1}(q_j)}{\|\vec{u}^{i-1}(q_j)\|} \cdot \vec{n}_j^i \right), \quad (38)$$

defines the integral expression with or without topography changes:

$$K^i = \int_{\omega^i} c^i dV = \int_{\omega^i} c_t^i dV, \quad (39)$$

where c_t is the modified expression for the concentration to account for topography changes and $\vec{n}_j^i$ is the normal to face $j = 1, \dots, 4$ of subdomain i . This implies a constraint on the modified expressions of a and b (e.g., correction on b can be deduced from a) through our analytical dispersion model.

$$K^i = \int_{\omega^i} c_t^i(a, b) dV \quad (40)$$

The correction in a corresponds to a scaling by a positive monotonic decreasing function worth one in the absence of topography changes. For instance, we can assume:

$$a_t^i = a^i \frac{\|\vec{u}^i + \vec{u}_t^i\|}{\|\vec{u}^i\|} \quad (41)$$

Hence, in case a change in topography increases the local velocity, the dispersion goes further downstream with less cross-diffusion due to decreasing b through constraint 40.

Unsteadiness and uncertainties

Let us recall the multilevel dependency chain in our simulation from topography and wind measurements to the species distribution:

$$(\psi, \vec{u}_{\text{obs}}) \rightarrow \{\vec{u}^i, i = 1, \dots, n_{\text{level}}\} \rightarrow \{c^i, i = 1, \dots, n_{\text{level}}\} \quad (42)$$

For the sake of simplicity, we assume the velocity field to be unchanged during the drift process and, therefore, stationary. Let us decompose the observation at a given point into a mean and a fluctuating part with zero mean:

$$\vec{u}_{\text{obs}} = \overline{\vec{u}_{\text{obs}}} + \vec{u}'_{\text{obs}}, \quad \overline{\vec{u}'_{\text{obs}}} = 0, \quad (43)$$

where time average is performed over the time interval of interest T :

$$\overline{\vec{u}_{\text{obs}}} = \frac{1}{T} \int_0^T \vec{u}_{\text{obs}}(t) dt \quad (44)$$

If the flow is stationary, $\vec{u}'_{\text{obs}} = 0$ and $\vec{u}_{\text{obs}} = \overline{\vec{u}_{\text{obs}}}$. If perturbations are weak, the deviation from the mean tendency is small and can be represented by a normal law, for instance:

$$\vec{u}'_{\text{obs}} = \mathcal{N}(0, \sigma_{\text{obs}}), \quad 0 \leq \sigma_{\text{obs}} \ll 1 \quad (45)$$

As mentioned in “Flow field,” these deviations can be accounted for using Kriging interpolation (Kriging 1951; Chiles and Delfiner 1999).

Another elegant way to account for small variations of observations while species are emitted and which is not subject to the limitations related to Kriging⁶ is to take advantage of the low-complexity feature of the simulation platform and perform Monte Carlo simulations. Hence, we consider a set of observations (simulations) $j = 1, \dots, n_{\text{trials}}$:

$$(\psi, \vec{u}_{\text{obs}}^j) \rightarrow \{\vec{u}^{i,j}, i = 1, \dots, n_{\text{level}}\} \rightarrow \{c^{i,j}, i = 1, \dots, n_{\text{level}}\}, \quad (46)$$

where the trials are performed for “admissible” random choices of $\vec{u}_{\text{obs}}^j$ through

$$\vec{u}_{\text{obs}}^j = \overline{\vec{u}_{\text{obs}}} + \vec{v}^j, \quad \vec{v}^j \in \{\mathcal{N}(0, \sigma_{\text{obs}})\}^2 \quad (47)$$

⁶Mainly, we need to know the variogram to establish the covariances.

We can then define ensemble averages for the calculated velocity field and species:

$$\bar{\vec{u}}^i \sim \frac{1}{n_{\text{trials}}} \sum_{j=1}^{n_{\text{trials}}} \vec{u}^{i,j}, \quad \bar{c}^i \sim \frac{1}{n_{\text{trials}}} \sum_{j=1}^{n_{\text{trials}}} c^{i,j} \quad (48)$$

For a given level i , we can also have an estimation of the deviation from mean tendency for the velocity field and species concentration:

$$\vec{w}^i = \vec{u}^i - \bar{\vec{u}}^i, \quad s^i = c^i - \bar{c}^i, \quad (49)$$

and because $\overline{\bar{\vec{u}}^i} = \bar{\vec{u}}^i$ and $\overline{\bar{c}^i} = \bar{c}^i$, we have:

$$\overline{\vec{w}^i} = 0, \quad \overline{s^i} = 0 \quad (50)$$

with corresponding local standard deviations using, for instance, the maximum-likelihood estimate after assuming normal distribution for the results around their means:

$$\sigma_u^i \sim \left(\int_{\omega^i} \|\vec{w}^i\|^2 dV \right)^{1/2}, \quad \sigma_c^i \sim \left(\int_{\omega^i} (c^i)^2 dV \right)^{1/2} \quad (51)$$

Fig. 12 *Top*: drift based on a flow field evaluated from an instantaneous measurement. *Middle*: mean drift based on ensemble average and Monte Carlo simulation. *Bottom*: standard deviation

Figure 12 shows an example of mean and standard deviations for a plume in an unsteady flow. The unsteady perturbation corresponds to $\sigma_{\text{obs}} = 0.1$. We can see that, compared to an evaluation based on an instantaneous measurement, the ensemble average based on Monte Carlo simulation introduces an eddy diffusion well known in turbulent flow calculations. Beyond unsteadiness, this approach can be used to analyze the effect of any randomness or uncertainties in data.

GIS-based coupling

Linking dispersion model and GIS

First of all, we should remind the reader that we want to couple a low-complexity physical model, which involves unsteadiness and uncertainties, with GIS that tend to provide an accurate numerical copy of the study area surface. Thus, GIS can be used to apply the model in a richer georeferenced numerical environment. GIS capabilities regarding DEM generation and exploitation are significantly improving the former dispersion equations, as they allow the model to be run on any local topography. Furthermore, GIS permits directly mapping the drift process and getting standard atmospheric concentrations at given geographical coordinates. Then, it becomes rather easy to make the pesticide cloud interact with other relevant geodata and, so, to proceed to advanced risk analysis regarding, for example, bystander exposure and agricultural plot or water course contamination after treatments. Although GIS allows to gain some more precision and to tend toward being a useful predictive tool, we should keep in mind that the main objective of the reduced-order modelling approach is to provide mean tendencies of the spray drift with very low calculation cost, and that potential errors are duplicated into the GIS. In addition to this, automated geoprocessing tasks that are carried out, like point-based data interpolation or topography smoothing, can also add some more spatial incoherence and introduce a new level of uncertainty. As those limits regarding precision and application on real situations have been raised, it is now interesting to explain more precisely how the model and GIS communicate, and how we can get the best of spatial techniques to improve the model's efficiency.

Spray drift model programming

The reduced-order modelling approach made the model's programming aspects easier. The former equations, which are coupling local and global models, as

said in the “[Introduction](#),” have been transcribed in Fortran language, including the time-transport-based transport model and the wind flow calculation as routines. Every parameter of the models, such as the domain’s spatial extent, location of plots and wind point coordinates, vehicle speed and direction, quantity of pesticide, and local topography, is read by the program from input files. The fastness of the Fortran compiler (Page 2005) and the mesh-free approach allows us to compute the solution in only a few seconds depending on the size of the domain and on the elevation data resolution. The results are then written to an output file, which contains point-based information for the whole domain. The program has not yet been transformed into an independent GIS class as the integrated method suggests, but is used as a stand-alone and fast executable. Input and output files must communicate with GIS for the coupling, and this implies both spatial concept implementation and technical GIS programming skills.

Loose coupling method

Several ways to couple GIS and environmental models are known in the literature, mainly “tight” or “loose” coupling and, more recently, “integrating” systems (Karimi and Houston 1997). Each technique presents assets and limitations and is more or less adapted depending on the complexity of the model. In our case, the loose coupling has been chosen for several reasons that have to be explained. As the latest describes an approach where interfaces are developed with minimal assumptions between the sending/receiving parties, therefore, the risk that a change in one application will force a change in another application is reduced (Corwin and Wagenet 1996). Loose coupling also has multiple assets regarding development costs, as we want to couple an environmental model with existing GIS, and not coding an entire GIS software able to natively implement the dispersion model, as the integrated approach would suggest. As more and more GIS programs are being made available by open-source communities, we opted for Quantum GIS (QGIS) software to achieve the coupling, as it is one of the most highly capable open-source tools and offers advanced programming possibilities (Sherman et al. 2007a). Indeed, QGIS is based on a robust C++ API that presents plenty of spatial algorithms and native GIS functions. Those have been recently made accessible through Python bindings, which allow a simpler programming environment for developing specific QGIS plugins that directly interact with the core source code (Sherman et al. 2007b). We opted for this technical solution to

propose a user-friendly pesticide atmospheric spray drift plugin. It is dedicated to agricultural atmospheric pollution prediction and has been designed to be fast and to perform well, mixing reduced-order modelling and GIS development.

GIS as input data provider

The first roles of GIS deal with the automatic DEM extraction, needed by the model to compute the effects of ground variations on the windfield and the pesticide cloud movement. As the multilevel has been conceptualized to gain in topographic accuracy, We have to work with several DEM resolutions and to be able to extract pixel values from any loaded DEM in the GIS. Using the Python bindings, the pixel extraction can be simply done with some common Geospatial Data Abstraction Library (GDAL)⁷ commands. In our case, we use two successive *gdal translate* commands (Warmerdam 2005–2008), as described below:

```
gdal_translate -ot Float32 -projwin
    "+str(xmin) +" "+str(ymax) +"
    "+str(xmax)+" "+ str(ymin)+"
    srtm.tif clip.tif
```

This first command is done to clip the loaded DEM according to the user-defined extent for calculation.

```
gdal_translate -of AAIGrid clip.tif clip.asc
```

This second command is done to extract the elevation value of each pixel of the extent to an ESRI grid file. The obtained grid is then converted into *x, y, z* triplets (Finlayson 2007) needed by the model as topography inputs, using the *grd2xyz* python class.⁸ These successive commands permit to get the topographic input data in any GDAL supported format for the dispersion model, keeping the user’s DEM resolution and spatial projection.

The secondary roles of GIS as input data provider concern local meteorological data storage, also used by the dispersion model. Real meteorological data are difficult to acquire over long time series, as we said in the first section, so we had to collect data in the field with a movable weather station. Fortunately, we also had access to a larger amount of data by cooperating with the European Life-Aware project. The latter aims to demonstrate how the optimization of pesticide application techniques in wine growing can limit surface

⁷The GDAL/OGR open source library was developed by Franck Warmerdam, an independent developer from Egganville (CA), and the GDAL/OGR community developers.

⁸This code was inspired from one of David Finlayson from the US Geological Survey, Santa Cruz (USA).

and subterranean pollution by using new embedded technologies. Thus, several data sets are acquired in the field with a tractor's embedded measurement devices, mainly GPS, anemometers, and specific sensors for spraying measurements, all linked to a data logger for recording. Each data logger communicates with a spatial server, on which an agrometeorological PostGIS database was built. So, we can directly access the database from the plugin, using a QGIS/PostGIS connector (Sherman et al. 2007a), or via coding SQL queries into the Python code (Sherman et al. 2007b). This allows the plugin user to choose a specific source plot stored in the database, and so, to immediately use its linked meteorological data. Here is an example SQL query that we can use to access a chosen plot, asking PostGIS to return its identifier, geometry, and the corresponding wind data (Santilli et al. 1996):

```
SELECT id, winddirection, windspeed,
       the_geom FROM meteotable WHERE the_geom
       && 'POLYGON((0 0, 0 10, 10 10, 10 0))';
```

Spatialization of the model

Once those input parameters are made available for QGIS, we must use the latter to provide a georeferenced environment for the model's output data. This step deals with some basic file format conversion, the multileveled equation implementation, and some advanced geodata processing. The georeferencing technique is first presented, then we show how the GIS deals with the multiscaled dispersion model, and some cartographic ramblings are finally presented, regarding the best way to map pesticide atmospheric drift.

Georeferencing the model's topographic input data

The mathematical model works on a cartesian metric basis, which is not readable as it is by QGIS. As we want the plugin to be able to read any resolution in any geographic projection, the spatial properties of the image DEM have to be read and understood by the model. This is done by sending the resulting ASCII file of the *gdal translate* commands to the Fortran program, from which one reads the given tabular *x, y, z* file by accessing the standard comma-separated values (CSV) format (Warmerdam 2005–2008). The generated DEM is sent to Fortran using simple Fortran *open*, *do*, and *read* commands: Each triplet (i.e., each line of the former raster matrix) is then understood by Fortran and provides the elevation data on which the calculation has to be computed for every point of the domain. The main asset of this technique is to use QGIS raster

format capabilities directly, and so to be able to read many DEM formats.

Introducing the multileveled algorithm

The multileveled correction for ground variations let the user choose the number of levels wanted (i.e., *nlevels* in Eq. 23), as well as their spatial extent (see Fig. 9). This permits to define the local area where the DEM resolution must be finer in order to compute ground variations more precisely. This “microscale” area can be defined, for example, just around the considered source plot or any other area that presents particular topography or significative obstacle (like local depression, small hill or other interesting rock formations) to the spray drift. The accurate DEM layer must be loaded by the plugin user who has, therefore, the possibility to work with several layers. This could be done directly from Fortran from only one DEM layer that could be interpolated and resampled, as suggested in (Doytsher and Hal 2006), but we want the Fortran model and QGIS to stay independent and, especially, to let the QGIS user to load his/her DEM layers according the usual way (i.e., avoiding to work with predefined formats and DEM inputs in Fortran). In order to set the different spatial extent on which the several DEM have to be loaded, we use an adaptation of the *Region Tool*⁹ algorithm (Rowlingson 2007) in a recursive way:

```
def doneRectangle(self):
    level = self.iface.getMapCanvas().
        setMapTool(self.saveTool)
    self.updateBounds(self.r.bb)

def updateBounds(self, bb):
    self.xmindomain.setText(str(bb.xMin()))
    self.ymindomain.setText(str(bb.yMin()))
    self.xmaxdomain.setText(str(bb.xMax()))
    self.ymaxdomain.setText(str(bb.yMax()))
    newLevel = bb.xMin(), bb.yMin(), bb.xMax()
        , bb.yMax()
```

Cartography of pesticide clouds

The last step of the plugin development concerns the conversion of CSV outputs into standard GIS formats, but also the way we can enhance the cartographic rendering of the pesticide cloud. Using, once again, the QGIS API, we can first easily generate the model output results as ESRI shapefile (.shp) or any other OGR

⁹This code is inspired from one of Barry Stephen Rowlingson from the Mathematics and Statistics department of Lancaster University (UK).

Fig. 13 Example of vectorial pesticide cloud generated as an ESRI Shapefile (.shp) with applied continuous color on the concentration field

supported GIS vector format. This is done using the QGIS *QgsVectorFileWriter* class as presented below:

```
uri="plume.csv?delimiter=_v=QgsVectorLayer(
    uri,"vectorial plume")
QgsVectorFileWriter.writeAsShapefile(v,"
    vectorial-plume.shp")
```

Where *plume.csv* is the input CSV file that included longitude, latitude, and atmospheric concentration fields, and *vectorial-plume.shp* the created point shapefile. Once this was done, we could instantaneously apply some styling options to the created layer, in order to emphasize the concentration values. This can be done using the QGIS *QgsContinuousColorRenderer* class, by allotting a symbol type to the geometries and a couple of minimum and maximum colors for the continuous color rendering (see Fig. 13).

```
r=QgsContinuousColorRenderer(v.vectorType())
r.smin=QgsSymbol(v.vectorType(),"0","","")
r.smax=QgsSymbol(v.vectorType(),"1","","")
r.smin.setPen(QPen(Qt.green,1.0))
r.smax.setPen(QPen(Qt.red,1.0))
```

Thus, the resulting vectorial pesticide cloud is readable by any standard GIS program and can be used in a simpler way for spatial analysis and atmospheric pollution prediction.

Another point of interest for mapping pesticide clouds is the raster generation, as the spray drift is a diffuse phenomenon and a surfacic representation is much more readable than points in this case. The raster creation can significantly improve the cartographic message. In order to interpolate point-based values,

we opted for the inverse distance algorithm, assuming that the nearer a point to be interpolated is located to a point with a known value, the more similar the value of the point to be interpolated is to the known value in close distance. This can be done using the *gdal grid* interpolation capabilities, using the GDAL virtual format (i.e., VRT driver) (Warmerdam 2005–2008) and playing on the power and smoothing values:

```
gdal_grid -a invdist:power=1.0:smoothing
    =50.0 -txe"+str(xmin)+str(xmax)+"-tye"+
    str(ymin)+str(ymax)+"
-of GTiff -ot Float64 -l driftx driftx.vrt
    output.tif").readlines(),
```

where *-txe* is the spatial extent in which to interpolate (i.e., the user-defined extent via the region tool class), *-of* is the desired output format, and *-ot* is the raster type. As the point-based values are interpolated over the whole domain, we have to apply a vectorial mask, in order to account only for points with values, and so, to kill the raster nodata. This can be done using the clipping functions of GDAL, using a *gdal -clip* command line. Finally, and as for the vector outputs, we can apply coloring schemes and transparency values, using the QGIS *QgsRasterLayer* optional arguments (see Fig. 14), as suggested below:

```
r=QgsRasterLayer(fileName, baseName)
r.setDrawingStyle(QgsRasterLayer.
    SINGLE_BAND_PSEUDO_COLOR)
r.setColorShadingAlgorithm(QgsRasterLayer.
    PSEUDO_COLOR)
r.setTransparency(90)
```

Fig. 14 Example of interpolated and masked raster pesticide cloud

Conclusion and perspective

A low-complexity model for the prediction of passive scalar dispersion in atmospheric flows has been presented and coupled with open-source GIS. The solution search space has been reduced using a priori physical information and a nonsymmetric metric based on migration times has been used to generalize injection and plume similitude solutions in the context of variable flow fields.

The pesticide spray drift model has been applied on realistic topographies and meteorological input data through coupling the inputs reading method with digital terrain models and “real-time” PostGIS database. GIS capabilities regarding spatial data storage and management have thus been fully exploited in order to fit better to the vineyards landscape reality, and so, to tend to a true-to-life calculation. The topography impacts on pesticide clouds calculation has been applied using SRTM and southern France DEM layers, and the ground variations have been enlightened by several simulations. Both uncertainties and unsteadiness have been raised using fast Monte Carlo simulations.

Furthermore, the multilevel algorithm and its correction for ground variations provide more accuracy. Current work now deals with the optimization of reading and interpreting the topography data, but also with the integration and the use of several DEM into a single domain and in its implementation in the Python plugin. This is one more link between fluid mechanics equations and GIS algorithms to be established. In order to fully validate the topography effect, we will also have to realize simulations over longer time series and different kinds of slopes and microreliefs, in order to compare the resulting values with real atmospheric concentration values. To achieve the terrain validation, a agricultural watershed will be monitored with air sampling devices that have to be positioned according to the major wind flows. Comparison between numerical results and chemical air analysis are planned for the future.

Finally, a QGIS python plugin for atmospheric pollution prediction has been presented and detailed by illustrating the coupling concepts and explaining some of the functional code snippets. One of the major assets of the reduced-order modelling approach is to simplify the programming aspects of the coupling, and the same logic has been used regarding the open-source GIS development. The result is that both dispersion model and GIS can communicate with each other but stay independent. Although the plugin is already a usable and user-friendly tool for pollution prediction, some more Fortran and Python hacks have to be developed

in order to provide both optimized wind flow and pesticide cloud calculation and additional automated GIS functionalities.

Acknowledgements This work is supported by Region Languedo-Rousillon and CEMAGREF. The authors would also like to thank Pr. Veronique Bellon-Maurel (UMR ITAP) and the Life-Aware project's members for their kindness and the access to various meteorological data they provided. Thanks also go to Sylvain Labbe (UMR TETIS) for providing advice and various DEM layers for the study.

References

- Alauzet F, George P-L, Frey P, Mohammadi B (2002) Transient fixed point based unstructured mesh adaptation. *Int J Numer Methods Fluids* 43(6):729–745
- ArcGIS (2006) Geographic information system. <http://www.esri.com/software/arcgis>
- Borouchaki H, George P-L, Mohammadi B (1997) Delaunay mesh generation governed by metric specifications. *Finite Elem Anal Des* 2:85–109
- Brun JM (2006) Modélisation à complexité réduite de la dérive. Thesis, University of Montpellier
- Brun J-M, Mohammadi B (2006) Similitude generalise et modelisation geometrique a complexite reduite. *C R Acad Sci* 2511
- Chiles J-P, Delfiner P (1999) Geostatistics, modeling spatial uncertainty. Wiley series in probability and statistics. Wiley, New York
- Ciarlet P (1978) The finite element method for elliptic problems. North-Holland, Amsterdam
- Corwin D, Wagenet R (1996) Applications of GIS to the modeling of nonpoint source in the vadose zone: a conference overview. *J Environ Qual* 25:403–411
- Cousteix J (1989) Turbulence et couche limite. Cepadues, Toulouse
- Doytsher Y, Hal J (2006) Interpolation of DTM using bi-directional third-degree parabolic equations with fortran subroutines. *Comput Geosci* 40(3):2101–2124
- Dragosits U, Place CJ, Smith RI (1996) The potential of GIS and coupled GIS/conventional systems to model acid deposition of sulphur dioxide. Third international conference/workshop on integrating GIS and environmental modeling, vol 2511
- Finlayson D (2007) grd2xyz—Grid to x,y,z converter python script. David Finlayson personnel website <http://david.p.finlayson.googlepages.com/gisscripts>
- Hecht F, Mohammadi B (1997) Mesh adaptation by metric control for multi-scale phenomena and turbulence. *AIAA Pap* 0859
- Karimi H, Houston B (1997) Evaluating strategies for integrating environmental models with GIS: current trends and future needs. *Comput Environ Urban Syst (Elsevier)* 20: 413–425
- Krige DG (1951) A statistical approach to some mine valuations and allied problems at the Witwatersrand. Master's thesis of the University of Witwatersrand
- Markiewicz M (2006) Modelling of air pollution dispersion. Technical report, vol 2511
- Mohammadi B, Pironneau O (1994) Analysis of the k-epsilon turbulence model. Wiley, New York

- Mohammadi B, Pironneau O (2001) Applied shape optimization for fluids. Oxford University Press, Oxford
- Mohammadi B, Puigt G (1998) Wall functions in computational fluid dynamics. *Comput Fluids* 23:9
- Mohammadi B, Brun JM, Pironneau O (2006) Reduced order modelling of dispersion. Numerical analysis and scientific computing with pde and their challenging applications. Springer, Berlin Heidelberg New York
- Page CG (2005) Professional programmer's guide to Fortran 77. University of Leicester Website <http://www.star.le.ac.uk/~cgp/prof77.html>
- Rowlingson BS (2007) Region tool algorithm. Barry Stephen Rowlingson personal website <http://www.maths.lancs.ac.uk/~rowlings/Software/regionTool>
- Santilli S, Leslie M, Hodgson C, Ramsey P (1996) Post-GIS manual. PostGIS refractions website. <http://postgis.refractions.net/docs>
- Shephard N, ESRI technical marketing analyst (2006) Animations in ArcScene. <http://www.esri.com/news/arcuser/0103/files/animation.pdf>
- Sherman G, Sutton T, Blazek R, Mitchell T (2007a) Quantum GIS software user guide. Quantum GIS documentation website <http://gisalaska.com/qgis/doc>
- Sherman G, Sutton T, Blazek R, Mitchell T (2007b) Quantum GIS API documentation. Quantum GIS API Doc <http://doc.qgis.org/>
- Simpson J (1997) Gravity currents in the environment and laboratory, 2nd edn. Cambridge University Press, Cambridge
- Veroy K, Patera A (2005) Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds *Int J Numer Methods Fluids* 47(2):773–788
- Warmerdam F (2005–2008) GDAL/OGR API documentation. GDAL/OGR website. <http://www.gdal.org/>