Computation of the integrated flow of particles between polygons

Annie Bouvier, Kiên Kiêu, Katarzyna Adamczyk, Hervé Monod,
INRA, UR341 Mathématiques et Informatique Appliquées, 78350 Jouy-en-Josas, France

October 7, 2009

Abstract

To quantify the flow of particles over a heterogeneous area, some models require the integration of a pointwise dispersal function over source and target polygons. This calculation is a non-trivial task and may require a great deal of computing time. In this paper, an efficient and accurate algorithm is presented to integrate general individual dispersal functions between pairs of convex or non-convex polygons. Geometric calculations are performed using standard tools from computational geometry. Numerical integration is then performed either by a grid method or by an adaptive cubature method. The procedure is illustrated with a case study. It is shown that the cubature method is much more efficient than the grid method and that its error estimates are accurate. The algorithm is implemented in a C++ program, Califlopp.

Keywords:

Califlopp; Computational geometry; Cubature; DCUTRI; GENESYS model; Gene flow model; Integrated particle flow.

1 Introduction

In the environmental sciences, predictions are now often based on explicit spatial models. In general, spatial heterogeneity is taken into account by meshing. In other words, space is partitioned into homogeneous cells and flow between cells is modelled by equations derived from theoretical physical models. For instance, lava flow rates can be derived from a model for the downslope flow of a Bingham liquid (e.g., Dragoni, 1986 and Vicari, 2007). In Ojo (2007), pollutant transport in surface waters is modelled through partial differential equations. In Lenser (2007), movement of passive biomass in oceans is represented by movement tensors computed from numerical simulations involving the flow field.

Alternatively, for transport through a media considered as spatially homogeneous, as in the case of airborne transport, movement can be modelled using a dispersal function. For example, the dispersion of aircraft noise near airports is represented in Moreno-Jiménez (2003) by an equation in which the decrease of the sonorous pressure level is a function of the distance between the source and the affected site.

Dispersal functions are also used in agricultural models that focus on the diffusion of pollen, seeds (GENESYS: Colbach et al., 2001a,b; MAPOD: Angevin et al., 2008) or spores (BLIGHTSPACE: Skelsey, 2007). In such applications, the space is partitioned into fields or sub-regions inside fields, and simulation involves the computation of the amount of particles exchanged between cells. The inter-cell particle flow is determined from a plant-to-plant (or individual) dispersal function and from cell features such as crop density and genotype.

For models that involve a dispersal function and two-dimensional source and target cells, dispersion calculation is most often based on a point grid for the purpose of simplicity. Such a discretization requires rather heavy computations. Another limitation of current implementations is that they are restricted to cells with simple shapes such as quadrilaterals. As a result of our experience in gene flow modelling over landscapes, we have tried to compensate for both these gaps.

The calculation of the particle flow between polygons is a non-trivial programming task. Relatively simple algorithms can be designed for integrating the dispersal function over pairs of polygons, but they may not be able to properly address the wide diversity of polygon sizes and shapes encountered in actual crop fields. Moreover, the

calculation requires a great deal of computing time and therefore imposes severe limits on the size of the total area to be studied.

The algorithm for particle flow computation described in this paper aims to provide an efficient solution for this type of calculation. It first involves geometric computations: decomposition of polygons into non-overlapping convex parts, polygon triangulation (O'Rourke, 1998) and polygon intersection (O'Rourke, 1982). The particle flow is then computed using numerical integration. A required precision can be specified and an error estimate is provided (Berntsen, 1992). The algorithm can be used for any set of simple polygons. The dispersal function must be stationary, which means that it is invariant when both the target and source points are shifted together. However, it may be non-isotropic, for example, if we are dealing with a gradient due to wind. A polygon is considered as simple if it is bounded, connected and if it has no hole. A simple polygon does not need to be convex and may have an arbitrary number of vertices.

2 Integrated flow

2.1 Definition and main simplification

The basic inputs of the methods are the coordinates of the vertices of each polygon and the individual dispersal function ϕ. All points of the source polygon are considered to be emitting, while all points of the target polygon may receive a particle: the source and target areas are considered as *continuous*. Furthermore, the densities of emitting points and receiving points are supposed to be spatially *homogeneous* within a polygon. The individual dispersal function ϕ between two points x and y in $\mathbb{R}^2$ is assumed to depend uniquely on $y - x$, so that ϕ is a function of two-dimensional vectors. As a special case, dispersal may be isotropic and therefore depend only on the Euclidean distance between x and y.

The aim of the methods presented in this paper is to calculate the integrated flow from A to B for any pair of polygons A and B, in other words:

$$\mathcal{F}(A, B) = \int_A \int_B \phi(y - x)\, dy\, dx. \tag{1}$$

With the variable change $(x, y) \to (x, t = y - x)$, it can be shown that

$$\mathcal{F}(A, B) = \int_{\check{A} \oplus B} \text{area}(A \cap (B - t))\, \phi(t)\, dt \tag{2}$$

where $B - t$ is the polygon B translated by vector t, $\check{A}$ is the reflection of A with respect to the origin, and $\check{A} \oplus B$ denotes the Minkowski sum, i.e., the set of points $t = y - x$, where $x \in A$ and $y \in B$. Expression (2) requires 2D integration instead of 4D integration for expression (1). Sections 3 and 4 are devoted to the calculation of (2).

2.2 Example of an application

Much research has recently been devoted to gene flow models because they allow us to study how different types of agriculture can co-exist within a given landscape. In particular, the GENESYS model (Colbach et al., 2001a,b) for oilseed rape attempts to describe and predict cross-pollination between conventional and genetically-modified (GM) varieties of the same crop species. It includes an empirical dispersal function that was fitted to experimental data (Lavigne (1998)). In the GENESYS model, the rate of cross-pollination on a given non-GM field B is calculated as the proportion of pollen arriving on B that has been emitted by GM fields. In the simplest case, this is equal to:

$$\mathcal{C}(B) = \frac{\sum_{\text{GM}} q_A\, \mathcal{F}(A, B)}{\sum_{\text{GM}} q_A\, \mathcal{F}(A, B) + \sum_{\text{non-GM}} q_A\, \mathcal{F}(A, B)}, \tag{3}$$

where $\sum_{\text{GM}}$ (respectively $\sum_{\text{non-GM}}$) is the summation over GM (respectively non-GM) fields A, and q_A denotes the quantity of pollen per square meter emitted by field A. The quantity $\mathcal{F}(A, B)$ is as defined in equations (1) and (2), with $\phi(y - x)$ representing the proportion of particles emitted at point x that arrives at point y. Thus, in order to calculate the cross-pollination rate for all non-GM fields, it is necessary to calculate $\mathcal{F}(A, B)$ for all pairs of fields where A is any oilseed rape field and B is a non-GM oilseed rape field. Note that the second term in the denominator of $\mathcal{C}(B)$ includes $\mathcal{F}(B, B)$, since the major source of non-GM pollen for field B is usually field B itself. In the GENESYS model, cross-pollination is evaluated over several years with different crop allocations from one year to the next. Consequently, $\mathcal{F}(A, B)$ is calculated, by default, over all pairs of fields.

3 Geometric algorithms

The numerical computation of an integral of the type (2) requires the evaluation of the following:

- the polygon $A \cap (B - t)$,
- its area,
- the polygon $\check{A} \oplus B$.

Standard algorithms for these computations are available from computational geometry (see O'Rourke (1998) for an introduction).

In the specific case when both polygons A and B are convex, the intersection $A \cap (B - t)$ is computed using the "chasing" algorithm (O'Rourke, 1982). The complexity of this algorithm is linear with respect to the total number of A and B vertices. The computation of the area of $A \cap (B - t)$ from its vertex coordinates is performed using the well-known formula for polygon area (based on signed areas of triangles and valid even for non-convex polygons). The Minkowski sum $\check{A} \oplus B$ is computed using the star diagram-based algorithm (Guibas, 1983), which results in a linear complexity as well.

One way to extend this approach to non-convex polygons is to decompose A and B into unions of non-overlapping convex polygons (see Figure 1). The total flow between A and B is the sum of partial flows between parts of A and parts of B:

$$\mathcal{F}(A, B) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathcal{F}(A_i, B_j) \tag{4}$$

where n (resp. m) is the number of convex parts of A (resp. B). The condition related to non-overlapping is as follows: two distinct parts A_i and $A_{i'}$ may intersect only at their boundaries. Convex decomposition is achieved using a standard approach (Hertel, 1983):

- the non-convex polygon is triangulated using the ear-removal algorithm,
- edges of the triangulation are removed as long as all parts remain convex.

The following geometric computations are therefore involved in the algorithm: convex decomposition of arbitrary polygons, intersection of convex polygons, polygon areas and the Minkowski sum of convex polygons.

4 Numerical integration

An example of an integrand is shown in Figure 2. In general, numerical integration performance highly depends on integrand smoothness. In this case, the integrand is the product of two terms: $\text{area}(A \cap (B - t))$ and the dispersal function $\phi(t)$. The function $t \to \text{area}(A \cap (B - t))$ is piecewise linear. The integrand is therefore, at best, a piecewise continuously differentiable function. The actual smoothness of the integrand also depends on the dispersal function ϕ. In general, ϕ shows a peak at the origin. Eventually, ϕ may be singular at the origin (an example is given in Section 5). When A and B are overlapping, the origin is located inside the integration domain. When they are adjacent, as in Figure 2, it lies at the boundary of the integration domain, and when A and B do not intersect, it lies outside.

Two general methods for numerical integration are considered:

- a simple randomized discretization of the integral,
- a two-dimensional adaptive cubature method.

The first method, referred to here as the "grid" method, involves simple discretizations of the integral based on regular grids (see Figure 3). An approximation of the integral is obtained by summing up the integrand at all grid points and by multiplying the sum by the area of a grid tile. When the grid location is uniformly randomized, the approximation is an unbiased estimator of the integral. Furthermore, by using independent replicates, it is possible to assess the precision of the result. This method is quite robust (unbiasedness and convergence even for non-smooth integrands). However, the sampling scheme is rather simplistic and does not perform as well as adaptive schemes.

When several replicates are used, the final integral approximation is obtained by averaging.

The second method (see Figure 4) is based on the DCUTRI algorithm (Berntsen, 1992). This algorithm provides approximations of two-dimensional integrals defined on polygonal domains. An initial triangulation must be entered into DCUTRI. The contribution of the integral of each triangle is approximated using a linear combination of 37 integrand values (cubature rule). The cubature rule has a degree 13: it provides an exact result for any polynomial integrand with a degree less than or equal to 13. Local error estimates as proposed in DCUTRI are obtained by comparing the results of lower degree cubature rules using the same integrand values. The triangle with the largest error estimate is selected and divided into four sub-triangles. Once again, the contributions of the sub-triangles to the integral are approximated based on the cubature rule. This step is iterated until a nominal error is determined. The error estimation procedure is empirical (see Berntsen (1992) for details). It does not ensure convergence for any integrand. For instance, if the integral on a triangle is greater than the nominal error, and if the integrand happens to be null at all points involved in the cubature rule, DCUTRI will fail to determine the nominal error.

Adaptive cubature is expected to perform better when singularities lie on the edges of the initial triangulation. For instance, if the dispersal function is singular at the origin and if the domain of integration $\check{A} \oplus B$ contains the origin, a quicker convergence is expected when the origin is a vertex of the triangulation. Therefore,

- if $\check{A} \oplus B$ does not contain the origin, it is triangulated from an arbitrary vertex,
- if $\check{A} \oplus B$ contains the origin O, it is triangulated from O (see Figure 4).

A further refinement is to avoid integration over areas where the dispersal function is considered as negligible, that is, beyond a chosen distance $r_{\max}$. Thus, instead of integrating over the whole Minkowski sum $\check{A} \oplus B$, it is possible to integrate only on its intersection with a disk centered at the origin and with radius $r_{\max}$: the dispersion over the part of the Minkowski sum outside the disk is considered to be null. In practice, it is simpler to replace the disk by a regular polygon, e.g., an octogon containing it (see Figure 5). Integration is therefore still performed on a convex polygon.

5 A case study

5.1 Test data

To illustrate contrasted situations, four pairs of polygons were studied:

- (1,1): two identical convex polygons,
- (2,3): two convex polygons next to each other,
- (4,5): two polygons next to each other, one convex, the other non-convex,
- (6,6): two identical, highly irregular polygons.

These polygons are plotted in Figure 6; their areas and the number of vertices are given in Table 1.

Pairs of identical and pairs of contiguous polygons are the most complex cases for integration. In addition, they are of particular importance in the gene flow applications because the major pollen sources of a given field are itself and the contiguous fields.

The individual dispersal function, $\phi(t)$, is the oilseed rape pollen dispersal function described by Klein (2006) up to a distance of 50 meters, and by Devaux (2006) for distances greater than 50 meters. It is equal to:

$$\begin{array}{rlcccl} d + er + fr^2 & \text{when} & 0 & \leq & r & \leq & 1.5 \\ \dfrac{ab}{a + r^c} & \text{when} & 1.5 & < & r & \leq & 50 \\ \dfrac{ab}{a + h^c}\dfrac{(1+h)^g}{(1+r)^g} & \text{when} & 50 & < & r & & \end{array}$$

where r is the length of the vector t between the source and target points, and $a = 3.80$, $b = 0.039$, $c = 3.12$, $d = 0.34$, $e = -0.405$, $f = 0.128$, $g = -2.29$, $h = 50$. Distances are given in meters. A plot of the dispersal function is shown in Figure 7.

pair	polygon 1		polygon 2		estimated flow	error	number of
	vertices	area (m^2)	vertices	area (m^2)			evaluations
1–1	4	12 780	-	-	12 273.70	3.63	1 011 200
2–3	4	18 690	4	22 260	167.28	0.056	1 475 520
4–5	4	8 880	6	12 280	132.47	0.016	1 438 720
6–6	17	15 470	-	-	14 613.90	2.32	11 364 800

Table 1: Geometrical features of the tested pairs of polygonal fields and results of the randomized discretization method. The grid step is equal to 0.25 m; 50 replicates have been generated. The errors are calculated by the procedure described in Section 5.2. The last column is the average number of evaluations of the dispersal function at each grid generation.

5.2 Obtaining reference values

To obtain reference results, we first apply the grid method, which is a robust method, with high-resolution (0.25 m spacing) grids and 50 replicates. To determine the precision of the results, a basic bootstrap interval (Davison, 1997) is calculated for each pair of polygons at a confidence level equal to 99%, from 10,000 bootstrap replicates. The absolute error is estimated by half the width of this interval. The results are shown in Table 1. The errors are rather small, supporting our hypothesis that the results calculated with 0.25 m-tiled grids are precise enough to be considered as reference values.

It should be observed that the usual way of estimating the confidence interval using the quantiles of the Student distribution is not appropriate here because the distribution of the replicates is not symmetric, as illustrated by Figure 8 on the pair of polygons (1,1), in the case of 100 replicates of 1 m-tiled grids.

The number of evaluations increases as the polygon shape becomes more irregular and has to be decomposed into many convex subpolygons. For example, the highly irregular polygon 6 is decomposed into 7 convex subpolygons. To calculate the integrated flow onto itself with a 0.25 m-tiled grid, the number of evaluations is more than ten times greater than the number of evaluations required for the four-sided polygon 1 onto itself.

5.3 Comparison of the grid and cubature precisions

The precision of the cubature method is assessed from the absolute difference between the integral approximation calculated using the cubature method and the reference value. Assessing the precision of the grid method is not as straightforward since the result is stochastic. The criterion used for comparison is the 99th percentile of the absolute difference between the integral estimate and the reference value. The computation of the 99th percentile was performed using a Monte-Carlo procedure based on independent grid replicates. In order to compare the efficiencies of the cubature and the grid methods, the number of integrand evaluations required to obtain given precisions was computed for both methods. The results are shown in Figure 9.

As for running time, adaptive cubature is much more effective: to reach the same error bound, it executes about ten times less evaluations than the grid method for the pair of polygons (1,1), and up to a hundred times less for the other pairs.

5.4 Assessment of error estimation by the cubature method

The cubature method provides integral approximations and error estimates. The error estimates are based on empirical rules. In order to determine whether these errors tend to be over or under-estimated, we compared different values of the relative errors estimated by the cubature method with the relative differences between the results and the reference values. The result of this comparison is illustrated by Figure 10: all the points are above the diagonal. The errors estimated by the cubature method are greater than the expected values, and we can reasonably hypothesize that the required precision is reached with a safe margin.

5.5 Practical limitations

The running time rapidly increases at the same time as the number of polygon vertices. Figure 11 shows two polygons with similar global shapes, but polygon 1bis has 12 extra vertices compared to polygon 1. The particle

flow on polygon 1 (respectively 1bis) was computed using the cubature method with a required relative precision equal to 0.1%. The number of integrand evaluations is 10 times greater for the "noisy" polygon 1bis than for polygon 1. It is therefore recommended to remove any spurious polygon vertices before particle flow computation.

The required precision is another critical factor for computation speed. For instance, flow computation between polygons 4 and 5, using the cubature method is two times as long for a required relative precision of 0.1% than for a required relative precision of 5% (see Figure 9).

Integration by adaptive cubature is quite efficient (compared to integration based on grid sampling). However, it should be observed that adaptive cubature may fail to converge, as discussed in Section 4. It is therefore recommended to test flow computation based on adaptive cubature before intensive use. In particular, it is possible to compare flows computed by adaptive cubature and by grid sampling.

5.6 Application with non-isotropic dispersal

For airborne pollen plants, the dispersal is non-isotropic; pollination is dependent on the wind direction and intensity. The grid and cubature methods can be applied in this case as well.

To illustrate this, we chose the maize pollen dispersal model, BR-IG, described in Grimaud (2007), with Linear Quasilikelihood estimation of the parameters. The main wind direction is orientated from west to east (see Figure 12).

The results of the dispersal from the convex polygon 4 to the non-convex polygon 5 are illustrated by Figures 13 (a) and 13 (b), calculated in the same way as Figures 9 in Section 5.3 and 10 in Section 5.4. The same observations can be made when dispersal is isotropic:

- the adaptive cubature method is more effective than the grid method in terms of running time, and increasingly effective as the need for precision increases: it is nine times faster with a required precision of 5%, and 75 times faster with a required precision of 0.2% (see Figure 13 (a));
- the required precision has a strong impact on the computation speed for both methods: it takes 1.2 times longer to calculate results with a precision of 0.2% than with a precision of 5% by the cubature method, and 10 times longer by the grid method (see Figure 13 (a));
- while the cubature method provides error estimates only, we can be reasonably confident that it reaches the required precision with a safe margin: the errors resulting from this method tend to be over-estimated, as illustrated by Figure 13 (b).

6 Conclusion

The algorithm presented in this paper makes it possible to integrate a large number of stationary dispersal functions between pairs of convex or non-convex polygons. It includes a reliable estimation of its own accuracy so that the computing load can be adjusted to a required level of precision. In practice, the more well-behaved the dispersal function, the faster and more accurate the algorithm. However, it is quite general and may be used for a wide range of situations.

The algorithm has been coded in a program known as Califlopp, which has recently been integrated into the GENESYS model on oilseed rape gene flow, allowing this model to run on more general and larger agricultural landscapes. It is hoped that Califlopp may be useful for many other models operating at the landscape scale in the agricultural, ecological and environmental sciences.

7 Software availability

The Califlopp program that implements the algorithm is freely available from the first author and on the site: http://www.inra.fr/miaj/public/logiciels/califlopp, under the GPL-like CeCILL-C licence.

It is written in C++ and the size of the source-files is 473K.

8 Acknowledgements

This work was partially supported by the European SIGMEA project on the modelling of gene flow at the landscape scale. The authors wish to thank Ms. Ying Fu for preliminary programming work during a training course in our unit and Dr. Nathalie Colbach (INRA) and Mr. Christophe Sausse (CETIOM) for extensive testing on real landscapes. They are also grateful to Ms. Gail Wagman for a careful rereading and rewording of their English.

References

Angevin, F., Klein, E.K., Choimet, C., Gauffreteau, A., Lavigne, C., Messéan, A., Meynard, J.-M., 2008. Modelling impacts of cropping systems and climate on maize cross-pollination in agricultural landscpaes: the MAPOD model. European Journal of Agronomy 28 471-484.

Berntsen, J., Espelid, T.O., 1992. Algorithm 706, DCUTRI: An algorithm for adaptative cubature over a collection of triangles. ACM Transactions on Mathematical Software 18 (3) 329–342.

Colbach, N., Clermont-Dauphin, C., Meynard, J.M., 2001. GeneSys: a model of the influence of cropping system on gene escape from herbicide tolerant rapeseed crops to rape volunteers. I. Temporal evolution of a population of rapeseed volunteers in a field. Agriculture, Ecosystems and Environnement 83 235-253.

Colbach, N., Clermont-Dauphin, C., Meynard, J.M., 2001. GeneSys: a model of the influence of cropping system on gene escape from herbicide tolerant rapeseed crops to rape volunteers. II. Genetic exchanges among volunteer and cropped populations in a small region. Agriculture, Ecosystems and Environnement 83 255-270.

Davison, A.C., Hinkley,D.V., 1997. Bootstrap methods and their application. Cambridge University Press.

Devaux, C., Lavigne, C., Austerlitz, F., Klein, E.K., 2006. Modelling and estimating pollen movement in oilseed rape (*Brassica napus*) at the landscape scale using genetic markers. Molecular Ecology 16 (3) 487-499.

Dragoni, M., Bonafede, M., Boschi, E., 1986. Downslope flow models of a Bingham liquid: implications for lava flows. Journal of Volcanology and Geothermal Research 30, 305-325.

Grimaud, A., Larédo, C., 2007. Stochastic Models and Statistical Inference for Plant Pollen Dispersal. Journal de la Société Française de Statistique 148 77-106.

Guibas, L.J., Ramshaw, L., Stolfi, J., 1983. A kinetic framework for computational geometry. In: Proc. 24th Annu. IEEE Sympos. Found. Comput. Sci. 100-111.

Hertel, S., Mehlhorn, K., 1983. Fast Triangulation of Simple Polygons. In: Proceedings of the 1983 International Fundamentals of Computing Theory Conference, Lecture Notes in Computer Science. Springer-Verlag:Borgholm-Sweden 158 207–218.

Klein, E.K., Lavigne, C., Picault, H., Renard, M., Gouyon, P.H., 2006. Pollen dispersal of oilseed rape: estimation of the dispersal function and effects of field dimensions. Applied Ecology 43 1141–1151.

Lavigne, C., Klein, E.K., Vallée, P., Pierre, J., Godelle, B., Renard, M., 1998. A pollen-dispersal experiment with transgenic oilseed rape. Estimation of the average pollen dispersal of an individual plant within a field. Theoretical and Applied Genetics 96 886–896.

Lenser, T., Constable, A., 2007. A nonparametric algorithm to model movement between polygon subdomains in a spatially explicit ecosystem model. Ecological Modelling 206 79–92.

Moreno-Jiménez, A., Hodgart, R.L., 2003. Modelling a single type environmental impact from an obnoxious transport activity: implementing locational analysis with GIS. Environment and Planning A 35, 931-946.

O'Rourke, J., Chien, C.-B., Olson, T., Naddor, D., 1982. A new linear algorithm for intersecting convex polygons. Computer Graphics and Image Processing 19 384-391.

O'Rourke, J., 1998. Computational Geometry in C. Cambridge University Press.

Ojo, T.O., Bonner, J.S., Page, C.A., 2007. Simulation of constituent transport using a reduced 3D constituent transport model (CTM) driven by HF radar: Model application and error analysis. Environmental Modelling & Software 22, 488-501.

Skelsey, P., van der Werf, W., Kessel, G.J.T., Rossing, W.A.H., Holtslag, A.A.M., 2007. Multi-scale modelling of infection pressure from *Phythophthora infestans*. Bulletin OEPP/EPPO Bulletin 37, 313-316.

Vicari, A., Alexis, H., Del Negro, C., Coltelli, M., Marsella, M., Proietti, C., 2007. Modeling of the 2001 lava flow at Etna volcano by a cellular automata approach. Environmental Modelling & Software 22, 1465-1471.

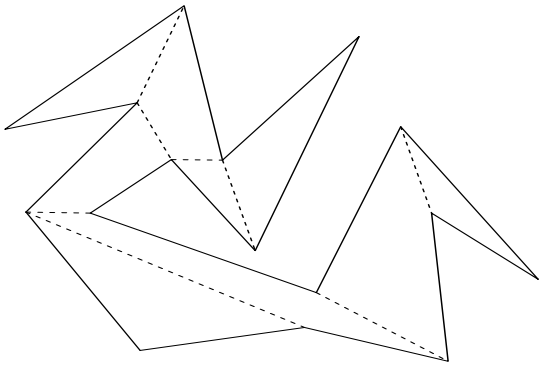


Figure 1: Convex decomposition of a polygon.

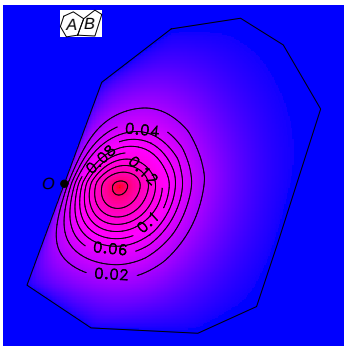


Figure 2: Example of integrand involved in the computation of the flow between two adjacent convex polygons. Both polygons A and B are shown in the inset. The dispersal function is Gaussian. The domain of integration is a convex polygon. The integrand at a given point t of the domain is the amount of particles exchanged by pairs of points in A and B, separated by t. It is represented by colors ranging from blue (low values) to red (high values).

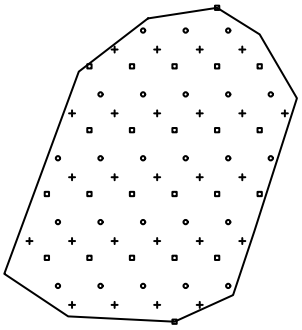


Figure 3: Randomized grids used to sample the integration domain. Several replicates (each represented with a given symbol) can be generated in order to assess variability.

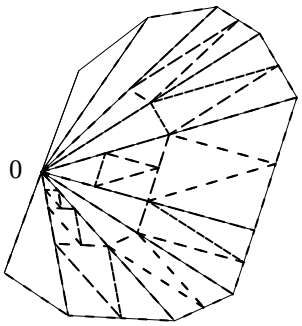


Figure 4: Integration by adaptive cubature. The initial triangles are shown in solid lines. The sampling of the domain of integration is adapted to the integrand.

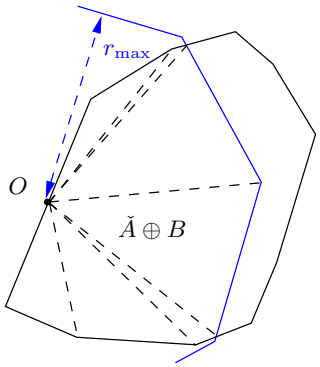


Figure 5: Reduction and triangulation of the domain of integration. When the dispersal function is considered as negligible beyond a given distance $r_{\max}$ from the origin, integration is restricted to the intersection of $\check{A} \oplus B$ with a disk, approximated by an octogon (blue), of radius $r_{\max}$ and centered at the origin.

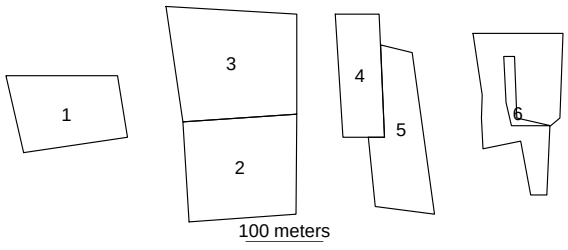


Figure 6: The tested polygonal fields.

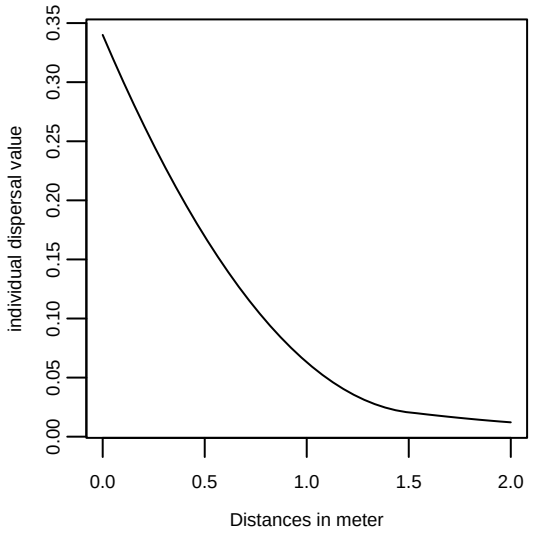


Figure 7: The tested individual dispersal function for distances between the source and target points up to 2 meters

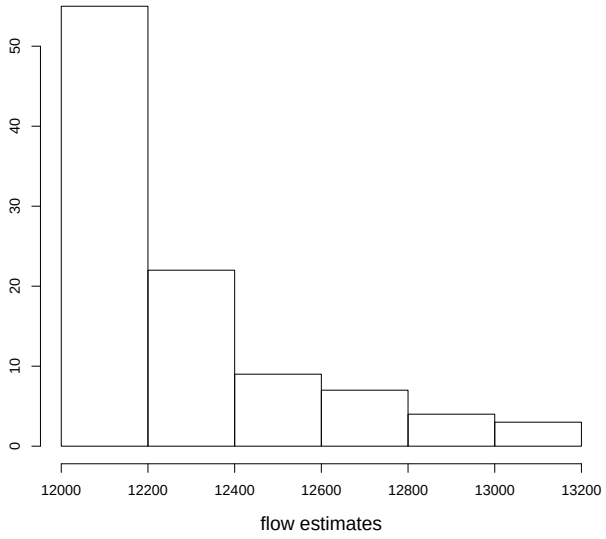


Figure 8: Histograms of the results calculated by the grid method over 100 replicates for the pair of polygons (1,1); the grid step is equal to 1 m.

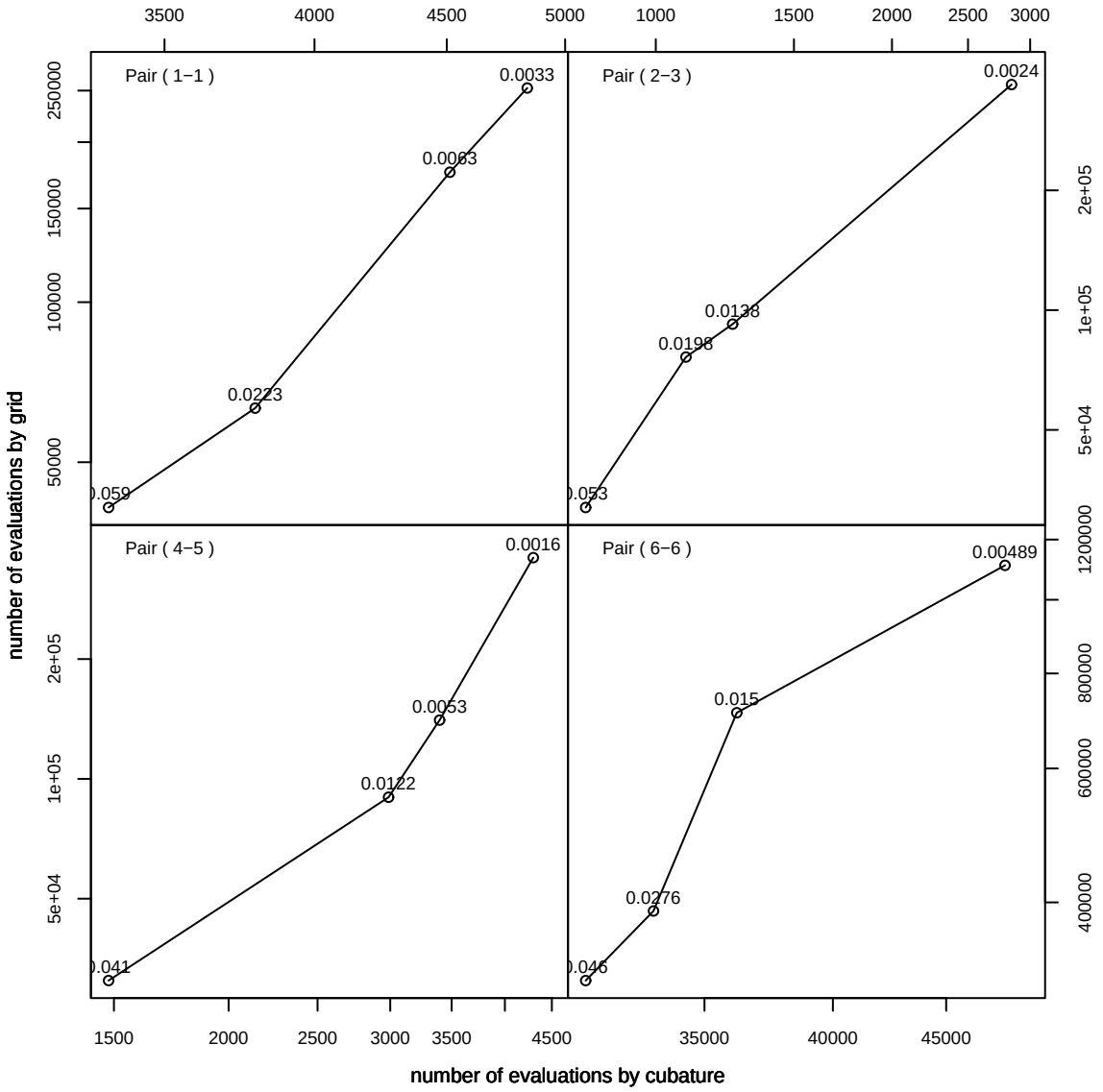


Figure 9: Number of evaluations executed by the grid and cubature methods to calculate results with the same precision. Different values of the relative errors are plotted. Each point is labelled with the corresponding error value. Its x-coordinate, (respectively, y-coordinate) is the number of evaluations executed by the cubature method (respectively, grid method) to calculate results with the corresponding precision. Coordinates are logarithmic.

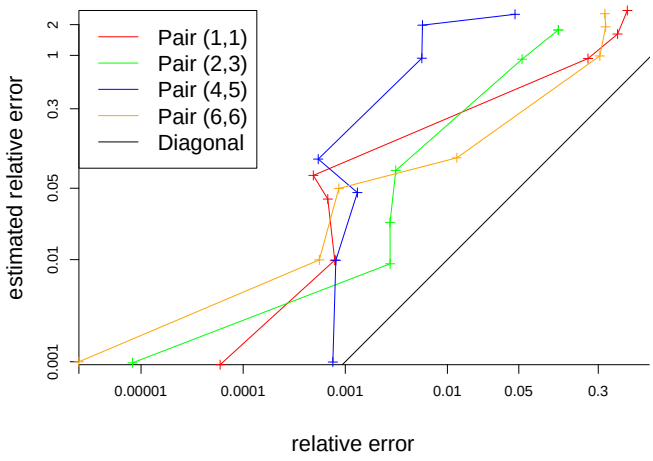


Figure 10: Relative errors as calculated by the cubature method against true relative errors. Coordinates are logarithmic.

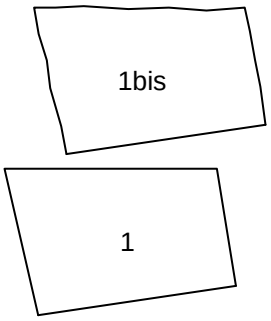


Figure 11: A polygon and its noisy version. Polygon 1bis is obtained by adding vertices to polygon 1 without modifying its general shape. Execution is longer with polygon 1bis onto itself than with polygon 1 onto itself with both methods.

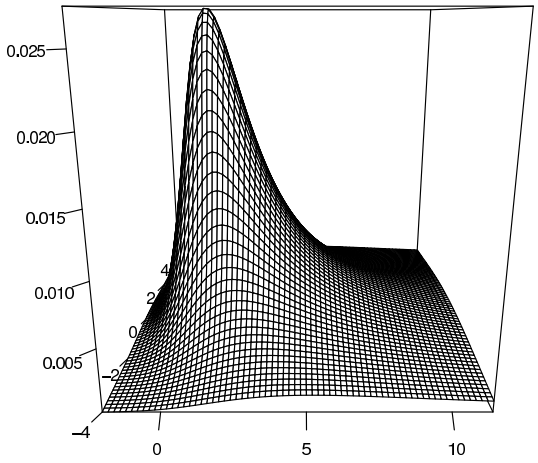
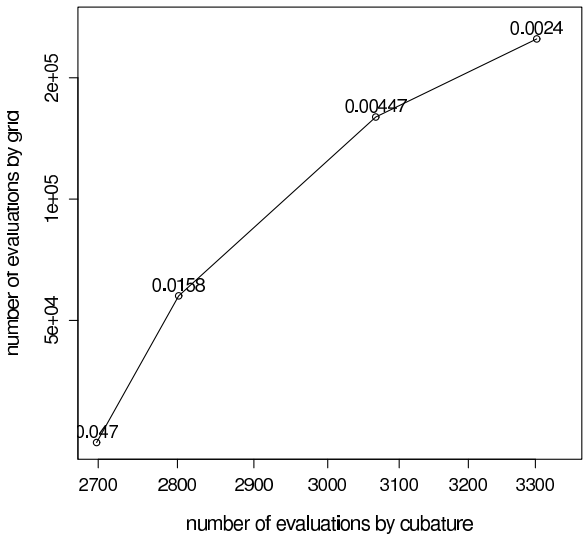
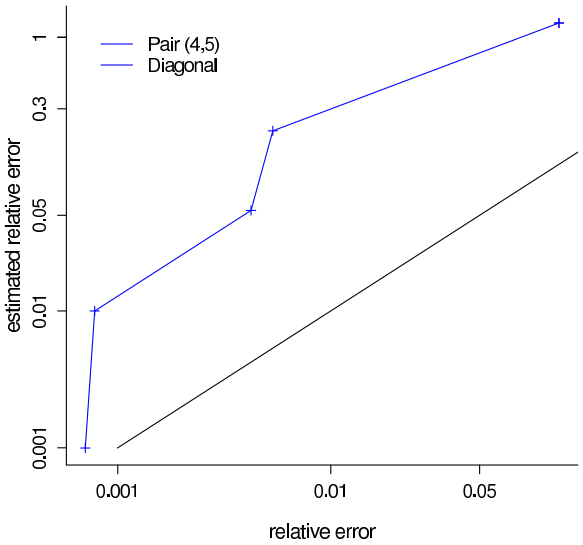


Figure 12: The non-isotropic dispersion function on a point grid: dispersion is greater along the direction of the main wind, here WE.



(a) Number of evaluations executed by the grid and cubature methods to calculate results with the same precision (see Figure 9).



(b) Relative errors calculated by the cubature method against true relative errors (see Figure 10). All the points are above the diagonal; the errors estimated by the cubature method tend to be over-evaluated.

Figure 13: Results from polygon 4 to polygon 5 with the non-isotropic dispersion function. Coordinates are logarithmic.