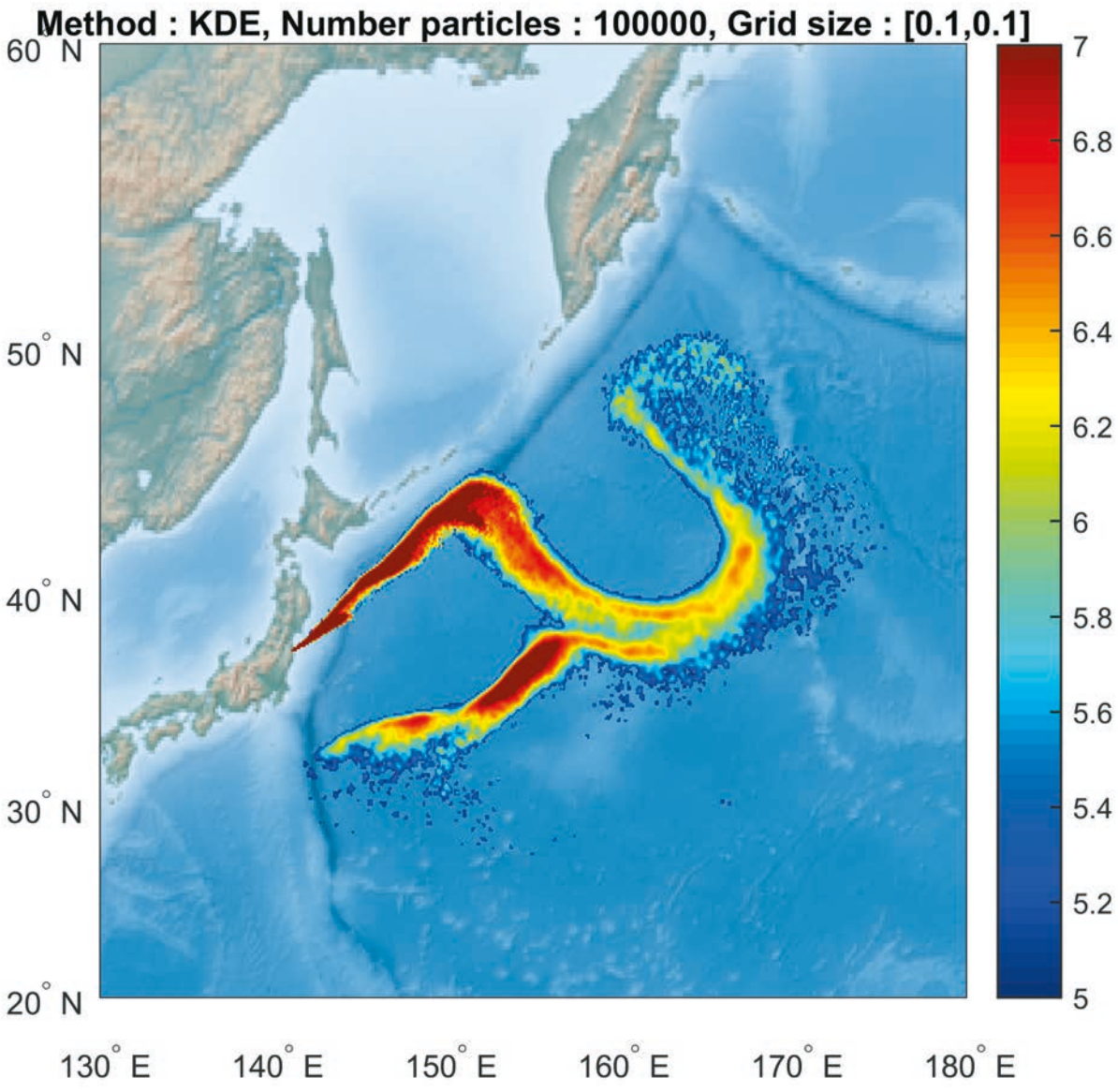
FOI

Post-processing of results from a particle dispersion model by employing kernel density estimation

OSCAR BJÖRNHAM, NIKLAS BRÄNNSTRÖM,
HÅKAN GRAHN, PETTER LINDGREN, PONTUS VON SCHOENBERG



FOI-R--4135--SE
ISSN 1650-1942 November 2015

Oscar Björnham, Niklas Brännström, Håkan Grahn, Petter Lindgren, Pontus von Schoenberg

Post-processing of results from a particle dispersion model by employing kernel density estimation

Titel	Databehandling med kernel density estimators av resultat från en partikelspridningsmodell
Title	Post-processing of results from a particle dispersion model by employing kernel density estimation
Rapportnr/Report no	FOI-R--4135--SE
Månad/Month	November
Utgivningsår/Year	2015
Antal sidor/Pages	35 p
ISSN	1650-1942
Kund/Customer	SSM
Forskningsområde	2. CBRN-frågor och icke-spridning
FoT-område	
Projektnr/Project no	B40089
Godkänd av/Approved by	Mats Strömqvist
Ansvarig avdelning	CBRN-skydd och säkerhet

Cover picture: Oscar Björnham

Sammanfattning

I strålskyddsberedskapen ingår spridningsmodellen Pello framför allt för att beskriva transport av radioaktiva partiklar efter en kärnvapendetonation, Pello är implementerad i beslutstödssystemet ARGOS via Matchskalet hos SMHI. Pello är en partikelspridningsmodell som innebär att en stor mängd modellpartiklar släpps ut för att representera en källa, dessa sprids därefter med de advekterande vindarna och späds av turbulens. För att visualisera resultatet räknas partiklarna traditionellt in i gridboxar, boxcounting, och koncentrationsfält beräknas och visualiseras på en karta.

Databehandling kan göras på många olika sätt, varav boxcounting är en metod, för olika syften. I den här rapporten har vi undersökt om kernel density estimators (KDE) istället kan användas för att databehandla resultaten från partikelspridningsmodellen Pello. Databehandlingsmetoder fördelar om massan från varje partikel till dess omgivning. Fördelen med KDE:er är att denna omfördelning kan göras med större urskiljning, vilket gör att såväl brus som överutslätning av modellresultat kan reduceras.

I rapporten presenteras några olika alternativa metoder för att beräkna KDE:er. Två metoder ”Integrerad turbulens” respektive ”Partitionsvarierande bandbredd” används sedan för att databehandla resultat från en och samma körning med Pello (ett Fukushima Daiichi-scenario). De databehandlade resultaten, både depositionsfält och luftkoncentrationsfält, jämförs sedan visuellt såväl som statistiskt (medelkvadratfel).

Givet ett fixt antal partiklar genererar KDE-metoderna resultat som är bättre (såväl reducerat brus som överutslätning) än boxcountingmetoden. Vi visar även att KDE-metoderna kan minska mängden partiklar som behöver släppas ut för att generera resultat av viss given kvalitet: KDE metoden kan generera likvärdiga resultat som boxcounting fast med färre partiklar. Resultaten tyder på att antalet partiklar kan minskas med åtminstone en storleksordning. En minskning av antalet partiklar leder i sin tur till en tidsvinst då simuleringstiden är beroende av antalet partiklar i körningen.

Nyckelord: Partikelmodell, Pello, kernel density estimation, depositionsfält, koncentrationsfält, databehandling

Summary

The dispersion model Pello is used, amongst other applications, for estimating and tracking dispersion of radioactive nuclides and gases. Pello is a stochastic particle model, where the source is represented by emission of model particles which are then transported by the wind field and diluted by turbulence. Today Pello is accessible to the Swedish radiation emergency preparedness system via an implementation in ARGOS through the Match framework at SMHI.

To visualise the result from Pello, the particles are traditionally counted grid-boxwise, box-counting, and thus the concentration field is estimated and then visualised on a map. Visualisation is one of several purposes of post-processing the result from the dispersion model, and box-counting is one post-processing method amongst many. In this report we have investigated whether kernel density functions (KDE) may serve as a good alternative method for post-processing of particle model dispersion results. All post-processing methods aims at redistributing the mass of each particle to its neighbourhood. The advantage with KDEs is that this redistribution can be done more delicately in order to reduce both noise as well as over smoothing in the model results.

In this report we present a number of different alternative algorithms to compute KDEs. Two methods “Integrated turbulence” and “Partition varying bandwidth” are then singled out for benchmarking against box-counting. The test case is a model run with Pello of the Fukushima Daiichi accident. The post-processed results, deposition fields and air concentration fields, are then compared both visually and statistically (mean square error).

Given a fixed number of particles, the KDE-methods generate results that are better (less noise, less over smoothing) than box-counting. We also show that, given a certain quality threshold, the KDE methods may reduce the relative number of particles that need to be simulated: KDE methods can yield equivalent results as box-counting, but with fewer particles. Our results indicate that the number of particles can be reduced by at least one order of magnitude. A reduction in the number of released particles will in turn reduce the time it takes to run the model.

Keywords: Particle model, Pello, kernel density estimation, field of deposition, field of concentration, post-processing

Content

1 Introduction

The dispersion model Pello is used, amongst other applications, for estimating and tracking dispersion of radioactive nuclides and gases. Pello is a stochastic particle model, where the source is represented by emission of model particles which are then transported by the wind field and diluted by turbulence. The output from the dispersion simulation is typically either a concentration field or field of deposited material. To aid the comprehension of these fields they are typically visualised on a map, see Figure 1.

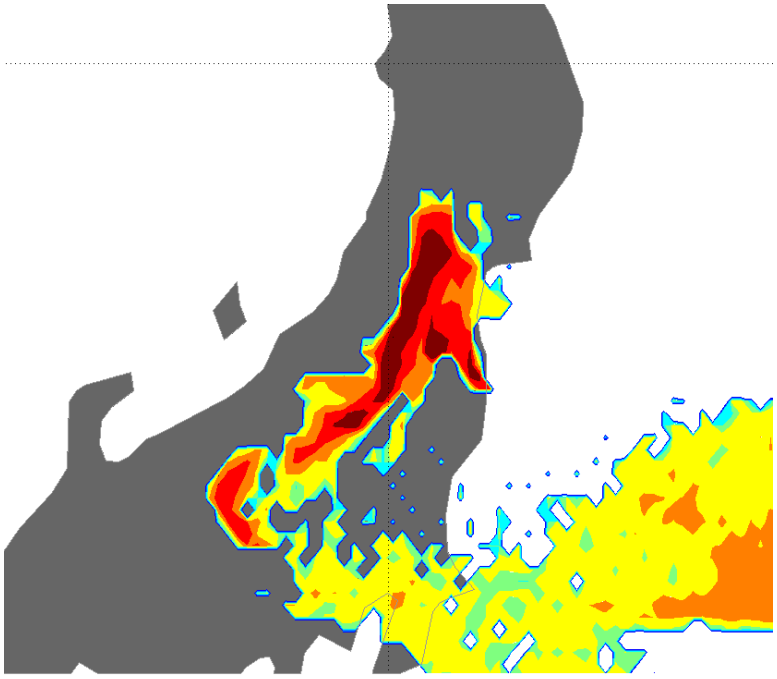


Figure 1. Example of how data from a stochastic particle model can be post-processed with box counting and visualised on a map.

The compilation of a concentration field is the result of post-processing the model results, and in this sense it is distinguished/separated from the model and any simulation using the model. The field may be compiled in numerous ways, but currently box counting is the method implemented in the emergency preparedness system. While being a simple and straightforward method it has a shortcoming: at locations with a small number of model particles, the box counting method gives large variations. As a consequence, the contour plot of the estimated concentration are scattered (“jumpy”) at these locations, see Figure 1. Kernel density estimation (KDE) is designed to decrease the variance and thus render contour plots smoother at positions with low density [1]. This report aims at investigating whether KDEs are suitable for post-processing of Pello model results. KDE should also perform as well as box counting for regions with high particle density, that is, there should be no relative loss in resolution. This is an important point as smoothing could also be achieved simply by picking larger boxes for the box counting, but that would also render a loss in resolution in areas with a large number of particles. Indeed, the criterion that KDEs should yield as good as, or rather better, concentration estimates than the box counting method has another implication: in the KDE case fewer model particles has to be released to obtain results of the same

quality as using box counting. If the KDE is appropriately chosen the number of particles can be reduced by an order of magnitude according to de Haan [1]. Particle models suffer from long simulation times, hence reducing the number of particles will yield in a welcome reduction in simulation time.

In this report we give a background to KDEs and we discuss different KDE algorithms to motivate our choice of algorithm to implement. The objective of the work described in this report is to find a better algorithm for calculating concentration fields from Lagrangian particle models by making the concentration fields smoother while keeping high resolution in areas containing high concentration.

The developed algorithm will also be benchmarked against deposition data from the Fukushima Daiichi Power Plant accident [2, 3].

1.1 Model particles

Pello uses model particles to simulate the atmospheric dispersion of a plume. A model particle represents a fraction of the cloud and each such model particle has properties such as sedimentation velocity and radioactive activity. To represent an entire plume, a large number of model particles are required. In Pello the model particles span the plumes total activity, actual particle size distribution, plume spatial size, and plume age. One model particle does in reality represent an entire cluster of real particles with equal properties. Each model particle is transported with the prevailing wind and diffused with a stochastic contribution representing atmospheric turbulence.

1.2 Compilation of concentration fields

Since a model particle only represents a point in space, information from nearby model particles has to be used to generate a concentration field. In the current implementation of Pello this is done by means of box counting where the space is divided into a grid with three-dimensional cells. In each cell the activity is a sum from all model particles inside the cell and the concentration is then the activity divided with cell volume. This method is fast and straightforward. However, it also generates discrete irregularities in the fields and depends on the choice of grid.

1.3 MatchPello and ARGOS

Today Pello is accessible to the Swedish radiation emergency preparedness system via an implementation in ARGOS. In this system the model is known as MatchPello as it resides at SMHI’s servers under their Match framework.

2 Kernel density estimation (KDE)

Kernel functions regard each particle position as a density distribution (a smoothing kernel) instead of just a point. The distribution could be of any type but are typical normal (Gaussian). At a local scale, dispersion processes typically leads to a normal distribution which suggests that this distribution is suitable as kernels in this project. The density estimation at any given position is then the sum of each smoothing kernel at that position. Figure 2 shows how KDEs are calculated for one-dimensional data and compares it with the method of box-counting (in the one dimensional case box-counting is the same as drawing histograms).

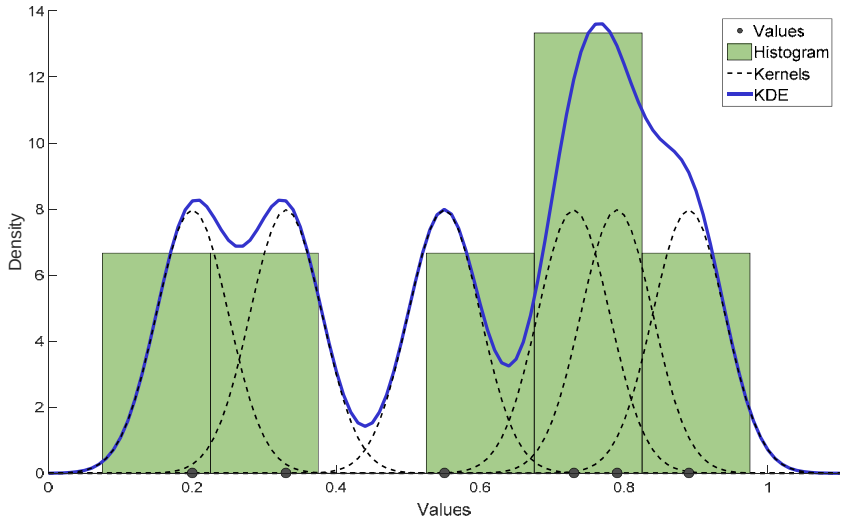


Figure 2. KDE versus box-counting. Box-counting of six data points (market as black points on the x-axis). They are collected into a histogram depicted in green. To obtain a KDE each point is represented by a kernel function shown with dashed black line. These kernels are summed up to obtain the collective KDE of the six data points, depicted by the thick blue line. The difference between box-counting and a KDE representation is illustrated here.

As a remark, before proceeding, we notice that box-counting is in fact a KDE-method as well: the KDE-distribution that corresponds to box-counting is a uniform distribution in the box in which the model particle is located. However, note that this type of kernels are not self-centred, which inherently gives rise to a decreased accuracy in the concentration field. The fact that this distribution is binary, i.e., only contribute to one box, strongly contributes to the discontinuities in the resulting concentration/deposition field. While keeping the simplicity of the box-counting method but alleviating some of the problems with discontinuity the method may be augmented to a neighbouring-box-counting method. Here each model particle shares some of its weight (uniformly) with the immediate neighbouring boxes. Neighbouring-box-counting is also a KDE-method (a superposition of uniform distributions with varying support).

As motivated above we will only consider normal distribution in the kernel function in the remainder of this report. The kernel will furthermore be self-centred, thus we are left with determining the standard deviation of this (multi-

dimensional) distribution. In the literature the standard deviation of a smoothing kernel is called the bandwidth of the smoothing kernel. The selection of bandwidth has a strong influence on the resulting density estimation.

2.1 Bandwidth selection of an individual kernel

Deciding the bandwidth is very important for getting good estimations. One can use a single bandwidth in all dimensions (Type A), different bandwidths in each dimension (Type B) or use a d x d-matrix (Type C), where d = number of dimensions, for getting the shape of the kernel in any direction. Figure 3 shows the 2D-kernel shape for the three types. Type A can also be considered as a scalar times the d x d identity matrix, Type B as a diagonal d x d matrix with positive numbers at the main diagonal and Type C as symmetric positive definite d x d matrix. The two first types are special cases of the general Type C.

There are competing views on how to pick the ultimate bandwidth on kernels depending on which measure you use to estimate the error (the *Integrated Squared Error* or the *Mean Integrated Squared Error*). The bandwidth also depends on the number of particles, *n*, in the data set. In a survey paper Turlach [4] found that the bandwidth should scale with a factor of $n^{-0.2}$.

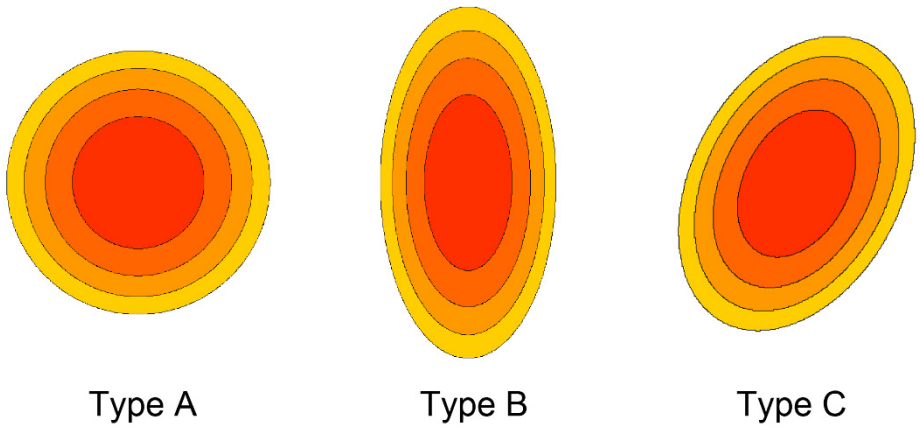


Figure 3. 2D-kernel shapes for different bandwidth types. The left picture, type (A), shows a 2D-kernel shape of a constant bandwidth, the picture in middle, type (B), shows a 2D-kernel with different single bandwidths in both dimensions, and the picture to the right, type (C), shows a 2D-kernel that can take any direction due properties of a symmetric positive definite bandwidth matrix.

There are two main school of thoughts on how to choose the bandwidth: a prior choice and a posterior choice.

2.2 Choosing the bandwidth a priori

By choosing the bandwidth a priori we mean that the bandwidth is determined at run-time by the dispersion model. Each model particle will be assigned its

own kernel bandwidth depending on the physical processes it undergoes in the dispersion model, hence the bandwidth will directly depend on the scenario, including meteorological conditions, at hand. In essence the bandwidth will increase with the amount of turbulence that the particle has endured during its flight through the atmosphere. This approach has for example been adopted in the DERMA particle model (thereby making it a puff-particle model) [5].

2.3 Choosing the bandwidth a posteriori

If the bandwidth is chosen after the dispersion model has finished and delivered its result (a swarm of dispersed particles) we say that the bandwidth is chosen a posteriori. Since the model run at this point is *a compli fait* there is only an indirect influence of the physics involved. In essence the bandwidth will be chosen after an initial assessment of how the particles are distributed, see e.g. [1] and [6].

2.3.1 Variable bandwidths

Often the same bandwidth is used for all smoothing kernels (particle positions). KDE with constant bandwidth is often good enough but for data with large variation in density the method are over-smoothing at high densities and und-smoothing at low densities. The use of variable kernel bandwidth on the other hand, allows for more adequate smoothing where particle density is high and are minimizing the variance where particle density are low.

To utilize variable bandwidths it is necessary to calculate, or estimate, the local density for each individual model particle. This requires an extra initial calculation which can be computationally intensive and become a severe drawback of this approach. Even so, in this report we opt to use KDEs with variable bandwidth while trying to find algorithms that reduce the computational cost. Indeed, we will divide the space into a number of partitions of equivalent density and use KDEs of constant bandwidth inside each partition: partition varying bandwidth.

2.3.1.1 Binned kernel

As mentioned, for a large number of particles, field compilation using a normal KDE may be very computer intensive. Therefore, alternative methods have been devolved to handle this problem.

To apply a varying bandwidth it is necessary to first establish the density at the location of each model particle. This can be done by, for each model particle, summing up the contribution from each other individual model particle. This is very computational demanding. Another way that are substantially faster is to use binned kernels.

Binned kernel is a method where all model particles are binned to a pre-defined grid. The contribution of a model particle to the calculation of the density at an arbitrary position depends on the distance between the sample point and the position.

2.3.2 Partition varying bandwidth

To acquire a relatively fast field compilation the FFT-approach is the best choice. However, it does not support variable bandwidths which is also a preferred feature to obtain the desired resolution and smoothness of the field.

One way of overcoming the problem with algorithms that does not handle varying bandwidths is to split data into partitions and use different constant bandwidths within the partition. The combined KDE will be a partition varying bandwidth. When this method is utilized, it is important to use the same grid points (the points where the density estimation is to take place) in all partitions in order to be able to combine the results into a partially varying bandwidth KDE. Figure 4 illustrates how data can be split into three parts according to sample point densities.

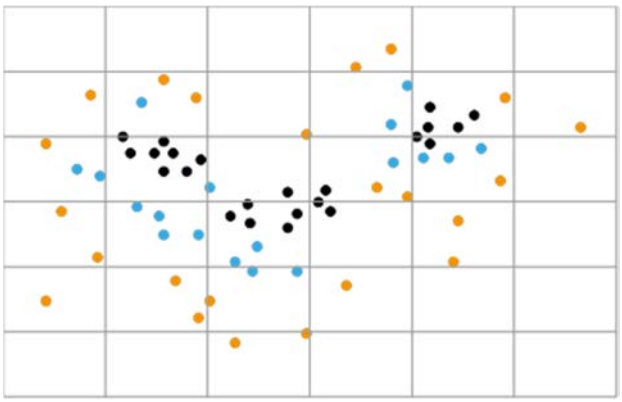


Figure 4. Partition based on sample point densities. Example of how data can be split into three parts according to density. Black means high density, blue means medium density and orange means low density.

3 Methods

In this chapter we explore several methods for choosing the bandwidth. We begin with describing one method for making a prior choice of bandwidths, and this is then followed by a presentation of several methods of making a posterior choice. Seemingly there are more degrees of freedom when making sound posterior choices of bandwidth.

3.1 A priori choice of bandwidth

In the dispersion model Pello each model particle is inert and independent. At every timestep in a Lagrangian random displacement model each model particle, if still airborne, will be translated. The length and direction of this translation is determined by two components: the main movement will take place along flow lines of the wind field, and then there will be diffusive (turbulence induced) jumps along or between flow lines. Typically we would expect the advective part to dominate the turbulent component. The turbulent component is modelled as a stochastic process, and it is this process that yields a spread amongst particles (in particular among particles released simultaneously in the same position). It is therefore natural to consider each model particle as the centre point of a swarm of nearby particles (i.e. a distributed particle) where the size of the swarm depends on how much dispersing turbulence that the particle has experienced. We model the stochastic turbulence by a random walk, thus a normally distributed (Gaussian) KDE is a natural choice, and we choose the bandwidth, a priori, to be the integral of the turbulence encountered.

Let us now consider the diffusive component mentioned above in more detail to see how the kernel bandwidth is estimated.

3.1.1 RDM – Random displacement model

For simplicity we only consider turbulence in one direction, the x-direction, but in the actual model turbulence acts in all three direction. At each small time step, Δt, the model particles takes a stochastic step given by

$$\Delta x = \sqrt{2K_x}\,dW \qquad (1)$$

where the diffusion coefficient K_x is given by

$$K_x = \sigma_x^2 \tau \qquad (2)$$

and σ_x is the variance in wind velocity in the x-direction and τ is the turbulent time scale. The stochastic contribution comes from dW which a Wiener process with variance Δt, this means that the process generates normally distributed steps with variance Δt or equivalently standard deviation $\sigma_W = \sqrt{\Delta t}$. The turbulence model is described in Lindqvist 1999 [9].

3.1.2 A random walk

The time discrete version of a Wiener process is a random walk. A random walk is the trajectory of particle that, on random, takes a step to the left or right. Each step is independent of the previous ones. The variance of the sum of two steps is

$$Var_2 = Var\left(a_1 X_1 + a_2 X_2\right) = a_1^2 Var\left(X_1\right) + a_2^2 Var\left(X_2\right) \quad (3)$$

and for N steps

$$Var_N = Var\left(\sum_{i=1}^{N} a_i X_i\right) = \sum_{i=1}^{N} a_i^2 Var\left(X_i\right). \quad (4)$$

3.1.3 Constant turbulence K - normally distributed KDE

If we make the simplifying assumption that K_x is constant then equation (4) implies that a set of particles performing this random walk will be normally distributed with variance

$$\sigma_R^2 = 2nK_x\sigma_W^2 = 2nK_x\Delta t = 2K_x t \quad (5)$$

where *n* is the number of time steps in the random walk. The assumption that K_x is constant is required to obtain an analytic expression for the bandwidth of the KDE. We choose to set K_x equal to the mean turbulence experienced by each particle, thus we integrate the turbulence along the trajectory of the given particle and divide by the total time. This choice of K_x together with equation (5) yields the KDE we are looking for. The choice (5) coincides with the choice made by Sørensen et al. [5] to describe the turbulent diffusion affecting the puffs in the particle-puff model DERMA.

3.2 A posteriori choice of bandwidth

Several methods were tested for being able to calculate KDEs with variable bandwidths on large data sets. These are described below. All methods are based on a two-step procedure, first a pilot run to calculate densities at all sample points and then a final step run with varying bandwidths derived from the densities estimated in the pilot run. The test data used are from Pello model runs for the Fukushima accident 2011, see Figure 5a for the particle distribution from this model run and Figure 5b for the corresponding box-counting estimation of the concentration field. Now, when computing KDEs for a large number of particles computational efficiency becomes a problem. A computationally effective method of computing KDE is to use the Fast Fourier Transform, see e.g. Silverman [7]. Botev et al. [8] presents a very fast FFT based KDE estimation using a linear diffusion processes and claims it outperforms existing methods in terms of accuracy and reliability. See Figure 5c for the resulting concentration estimation. A disadvantage of the FFT based method is that variable bandwidths is not applicable.

3.2.1 In-house program

No computer software was found that calculates two-dimensional KDE with variable bandwidths for each sample point and different bandwidths in each dimension. Therefore, a program was developed in-house that uses products of two one-dimensional KDEs (product kernel). The algorithm for calculating variable bandwidths is based on a pilot run with constant bandwidths for estimating density at each sample point. The sample point density estimation from the pilot run is then used to calculate the variable bandwidth according to [1]. This method works well for moderate data sizes, see Figure 5d. For large number of data-points (>10 000) the program is slow, especially due to the pilot run.

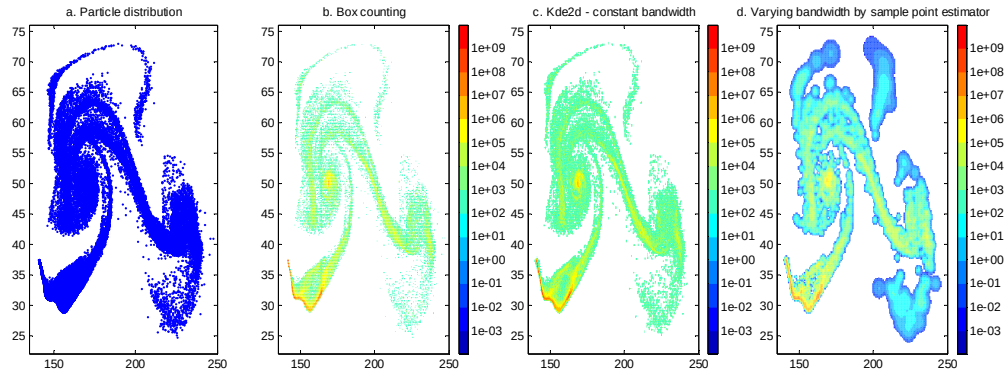


Figure 5. Contour plots for different estimation methods. a) Distribution of particles. b) Concentration by "box counting". c) Concentration by FFT-generated KDE (Botev et al. [8]) with constant bandwidth using all 62286 particles. The density is estimated on a 512x512 grid. d) Sample point estimator (individual bandwidth for all sample points) using product kernel (in-house program). 5000 particles out of 62286 are used.

3.2.2 Binned kernel + in-house program

An alternative method was to do the pilot run using a binned kernel technique and then using the in-house program for calculating the concentrations. The pilot run can for example be done using a KDE binning algorithm developed by Duong [10]. This method is faster than the FFT-algorithm used to generate Figure 5c and the in-house program used to generate Figure 5d, but for extremely large data sets (>100 000 data points), the second step is still slow.

3.2.3 Binned KDE + binned KDE

A very fast method is to use the binned KDE both in the pilot run and the second step. The results from such a run is illustrated Figure 6. The binning grid is defined in advance and often symmetrical in all directions (dimensions). If the variation in the particle positions mainly occurs in one direction, the

binning method might be a too big approximation

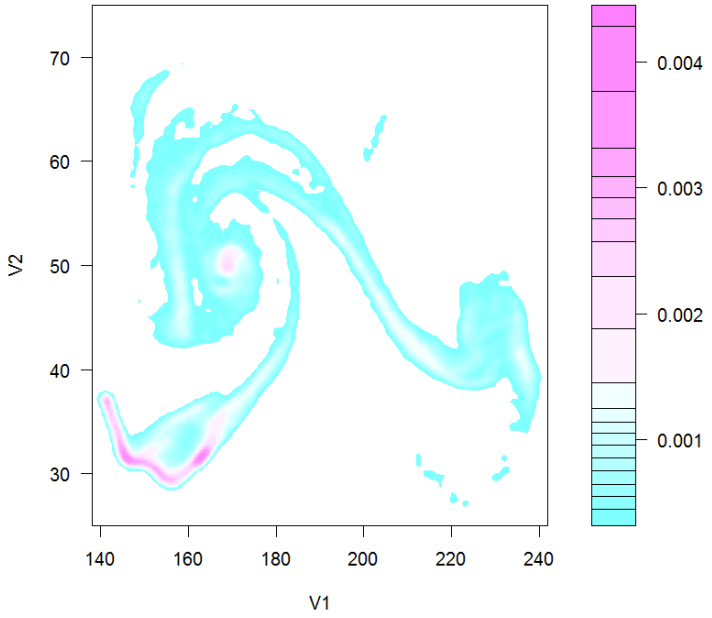


Figure 6. Binned KDE with varying bandwidths. KDE with sample point varying bandwidth for all data using binned option. First a pilot run with constant bandwidth and estimations at each sample point are performed using a KDE binning algorithm developed by Duong [10]. Individual sample point bandwidths are being calculated as a function of the density at the pilot run. These bandwidths are used in a new Binned KDE with the weight option.

3.2.4 Binned kernel + partition varying bandwidth KDE

The last method examined is a KDE with partition varying bandwidths. The pilot run can be done in the same manner in the two previous methods. The final step, with separate constant bandwidths KDE runs on each partition can for example be done using a FFT KDE. Figure 7 shows the results from such a run.

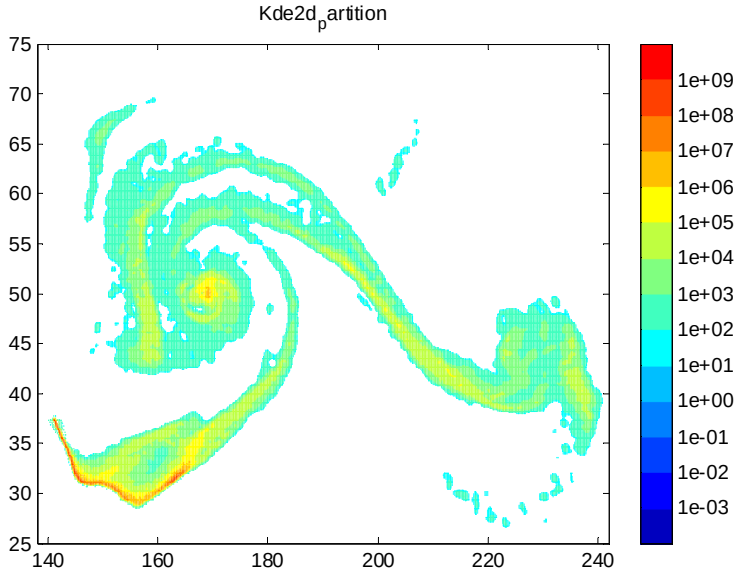


Figure 7. KDE with partition varying bandwidth. First a pilot run with constant bandwidth and estimations at each sample point are performed using a KDE binning algorithm developed by Duong [10]. The data are then being split into five parts depending on the KDE from the pilot run. KDEs with constant diagonal bandwidth are being calculated for each part (the bandwidths differ between the KDEs) on a predefined grid (same grid for each KDE) using a FFT-algorithm (Botev et al. [8]). The KDEs are then summed to form a total KDE with partition varying bandwidth.

4 Implementation

The current version of Pello holds a box-counting method to estimate concentration and deposition fields. This box-counting method is implemented in Fortran. The ambition in this project has been to make an equivalent implementation of one, or several, KDE methods. Due to limitations in resources we had to stop short of this and we mainly used MATLAB as our development environment.

4.1 A priori chosen bandwidth, integrated turbulence

Each particle experiences different amount of turbulence, therefore the turbulence is integrated separately for each particle. Pello has been updated to perform this integration (summation) at run-time. The integrated turbulence is then stored in the particle properties struct. This information is thereby made available to an external application handling the output from the dispersion model.

Generating the corresponding KDEs and plotting the result has been done in MATLAB using built in libraries.

4.2 A posteriori chosen bandwidth, partition varying bandwidth

We deemed that the method “binned kernel + partition varying bandwidth” was the most promising alternative for choosing the a posteriori bandwidth. This method has been implemented in a development environment in MATLAB and R. The pilot run that bins particles into regions of differing levels of concentration is done using an R package called ks developed by Duong [10], but modified to allow for the options “binned=true” and “eval.point = data” to be set simultaneously. The output from the R package is then read by MATLAB and in each partition (bin) the optimal bandwidth is computed using the FFT based KDE script KDE2d.mat by Botev et al. [8]. The final result is then visualised in MATLAB using build in libraries.

4.3 Cartesian grid

For long-range dispersion models it is natural to use lat-long coordinates, not least because weather data is supplied on that format. The scripts we have used for determining the a posteriori chosen bandwidth is however written for a Cartesian grid. Converting the fast fourier transform scripts from handling KDEs on flat Cartesian grids to instead coping with spherical surfaces in lat-long coordinates is outside the scope of this project. Then as a consequence, for the two KDE methods to be comparable, the a priori KDE was also written for a Cartesian grid. Converting the latter to a lat-long grid is however straightforward.

5 Results

To compare the different KDE methods (a priori and a posteriori chosen bandwidth) and box-counting we picked a well-studied test bench: the release of radioactive material following the Fukushima accident.

5.1 Case study

During the nuclear power plant accident in Fukushima 2011 radioactive material was released into the atmosphere. This case has since then been used to study behaviour of atmospheric dispersion models. Since this event is familiar to us and we have worked with is since the accident, e.g. [3, 11, 12], it provides a good case to study how KDEs behave. We study the release of Caesium 137, ^{137}Cs, into the atmosphere which is assumed to attach to surrounding aerosols (assumed to be a rural aerosol described by von Schoenberg and Grahn [11]). The variation of the source term in time comes from the latest official source term from Katata et al [13]. We have chosen to simulate the first 20 days of the accident for deposition studies and 3 days for air concentration fields.

5.2 Concentration in the boundary layer

We released different number of particles and post-processed the resulting concentration using box-counting and turbulence integrated KDEs respectively. Note that the partitioning varying bandwidth KDE is not used in this comparison. In this analysis we have taken into account all particles that are airborne within the planetary boundary layer at the time of comparison. In Figure 8, Figure 9 and Figure 10 we compare turbulence integrated KDEs with box-counting for 1 000 000, 100 000 and 10 000 model particles. The visualisation grid has a resolution of 0.1*0.1 degrees in latitude and longitude.

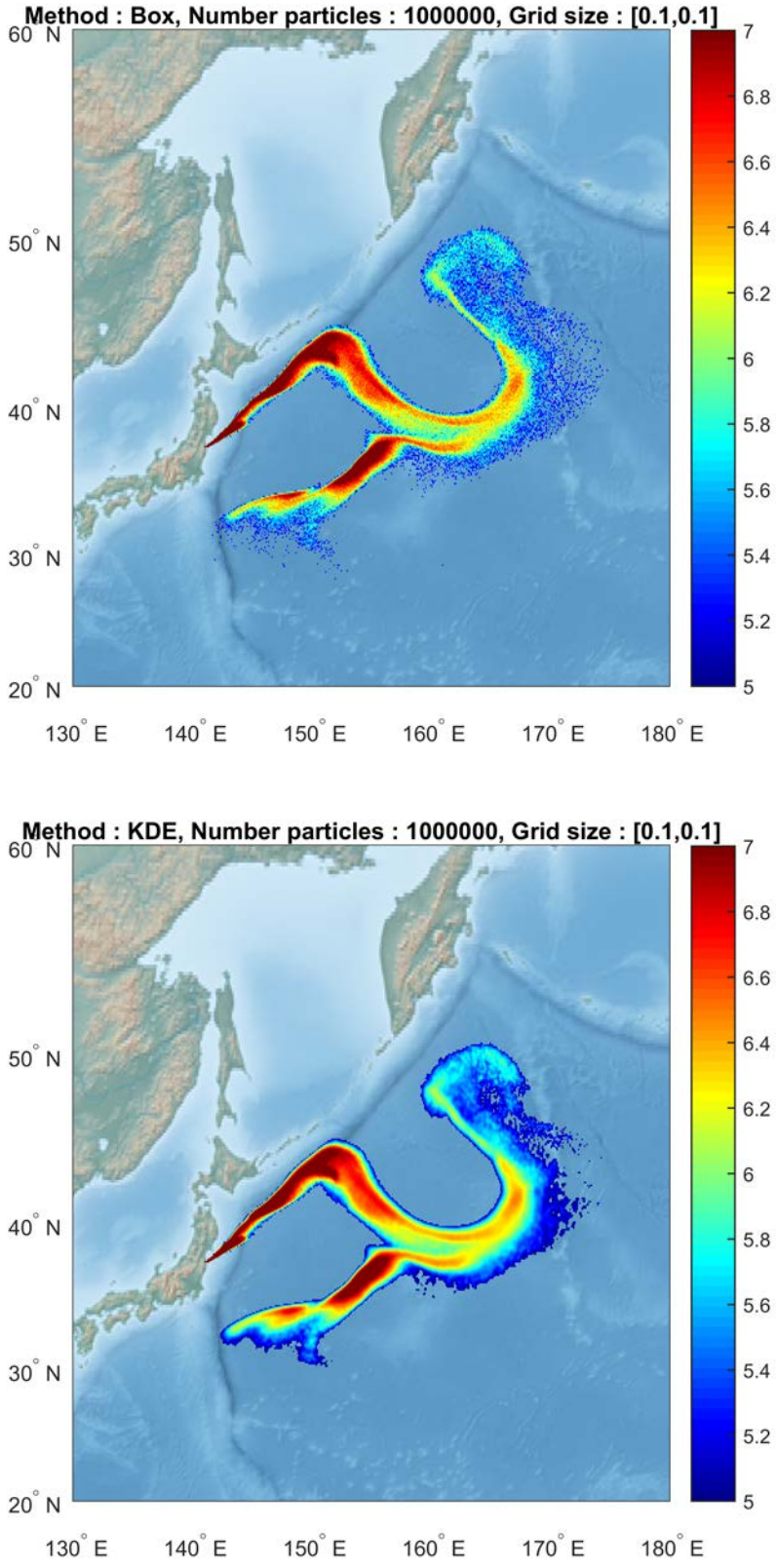


Figure 8. Box-counting (upper image, labelled BOX) versus integrated turbulence KDE (lower image, labelled KDE). Both images show post-processing of the same model run with 1 000 000 released particles.

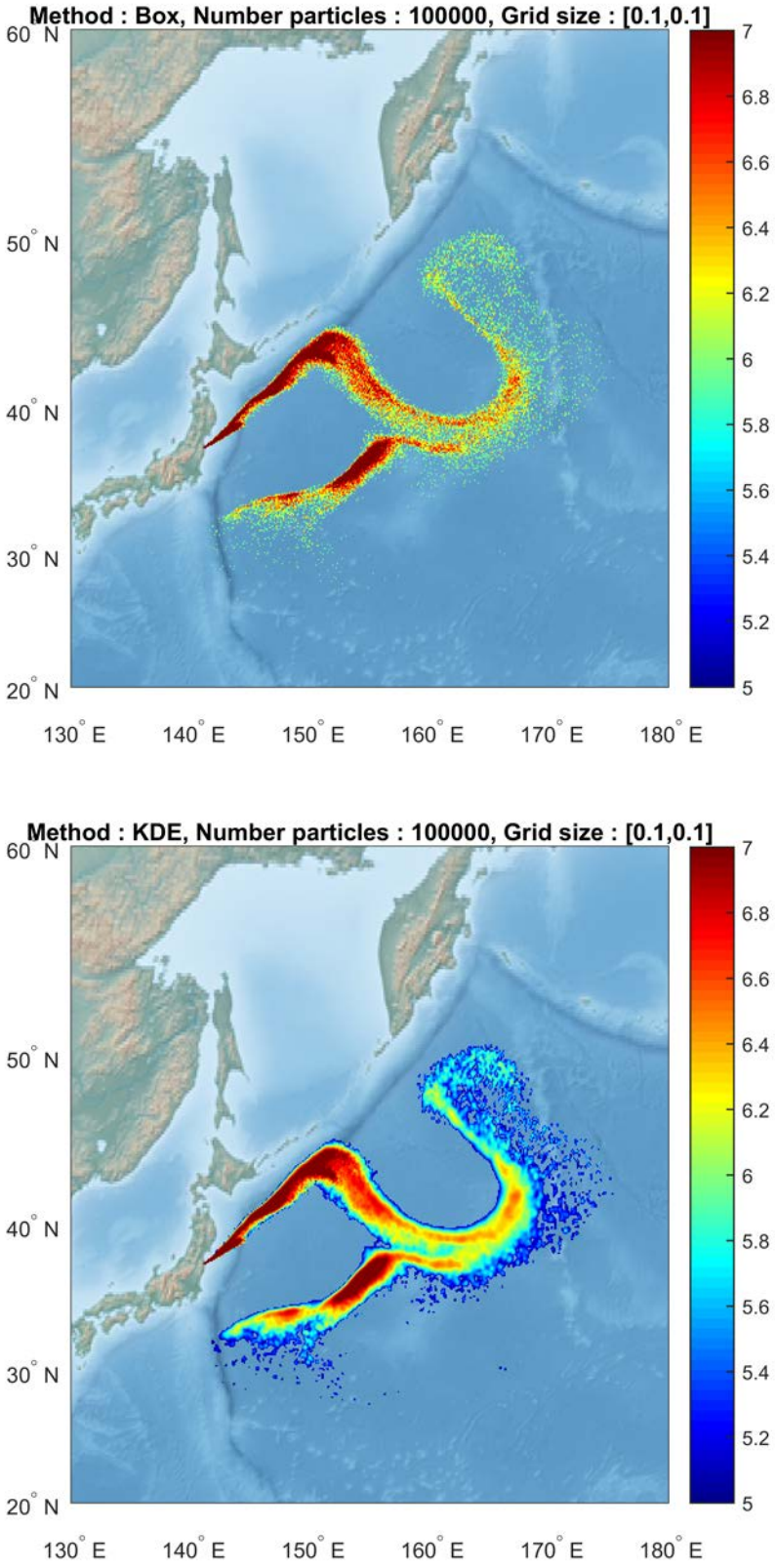


Figure 9. Box-counting (upper image, labelled BOX) versus integrated turbulence KDE (lower image, labelled KDE). Both images show post-processing of the same model run with 100 000 released particles.

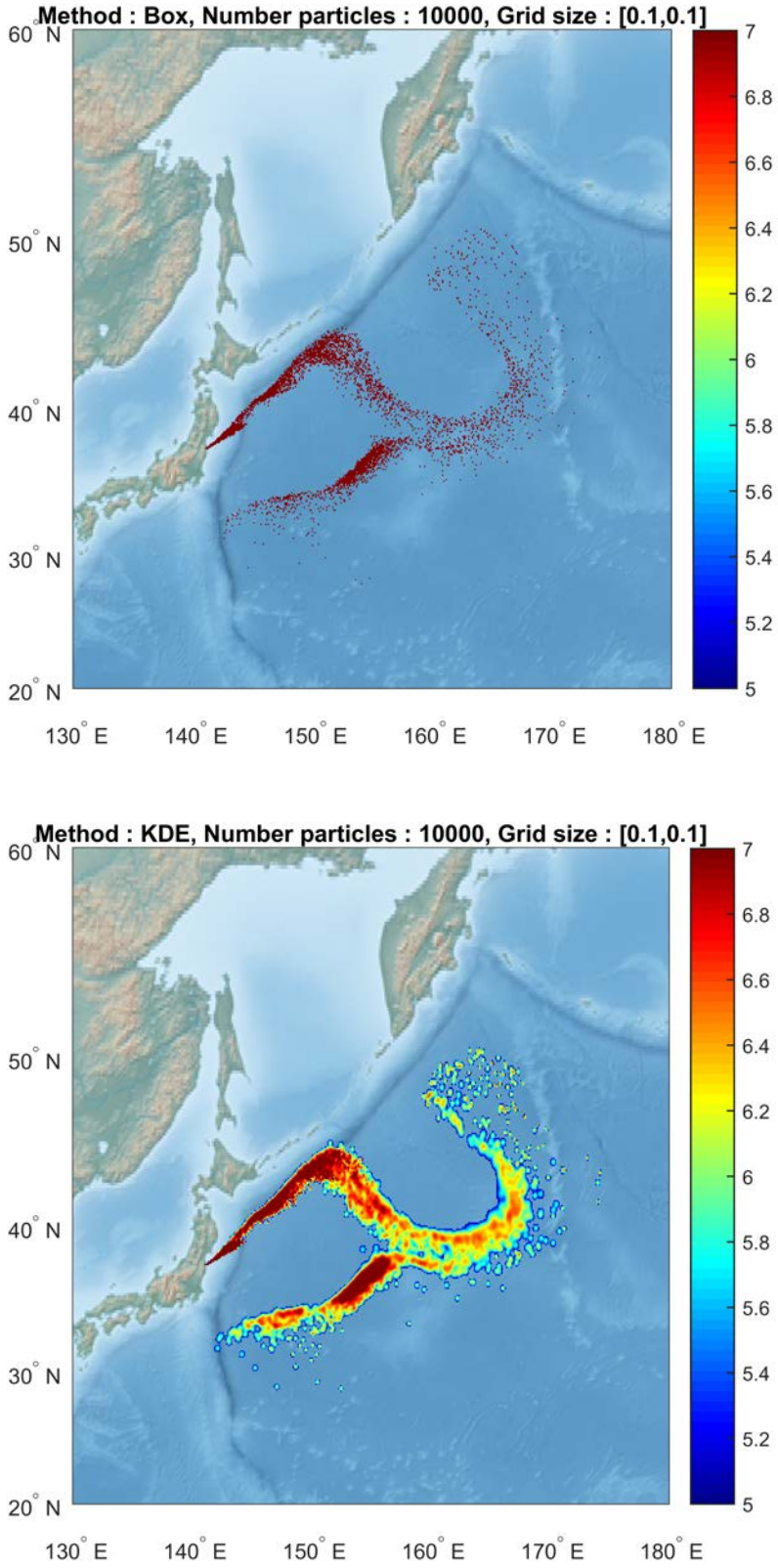


Figure 10. Box-counting (upper image, labelled BOX) versus integrated turbulence KDE (lower image, labelled KDE). Both images show post-processing of the same model run with 10 000 released particles.

5.2.1 The resolution of the visualization grid matters

We note that decreasing the resolution of the visualisation grid has two effects, the KDE will be less smooth, looking more like a box counting method, while the box-counting method looks smoother as the mass is distributed over a larger grid cell, see Figure 11.

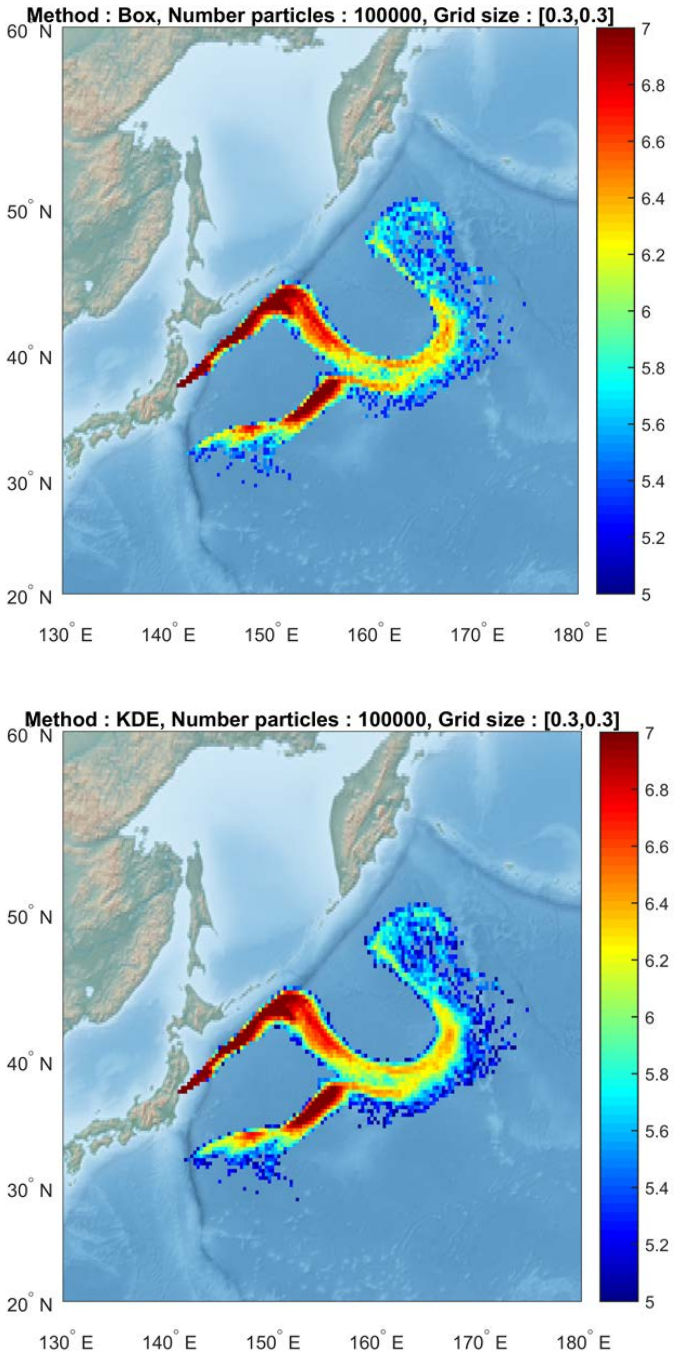


Figure 11. Box-counting (upper image, labelled BOX) versus integrated turbulence KDE (lower image, labelled KDE), with a coarser visualisation grid. Both images show post-processing of the same model run with 100 000 released particles. Comparing with Figure 9 we note that coarsening the visualisation grid has a smoothing effect on the box-counting method, while at the same time the behaviour of the KDE approaches that of box-counting.

5.2.2 Contour plots

Once the concentration fields have been estimated, regardless of method, there are several plotting algorithms which can be employed. Since contour plotting is popular we include such an example for reference, see Figure 12.

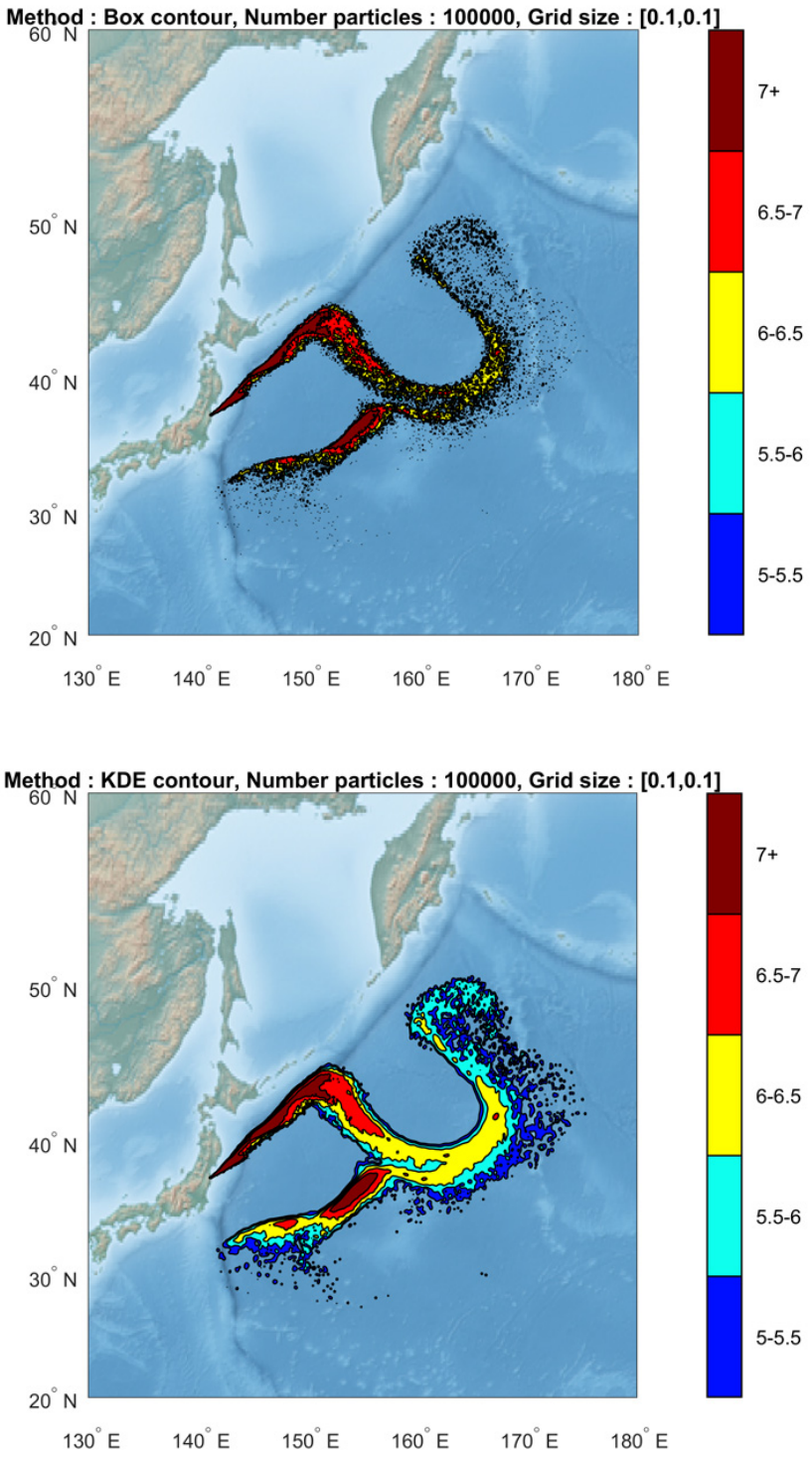


Figure 12. Box-counting (upper image, labelled BOX) versus integrated turbulence KDE (lower image, labelled KDE), both plotted using a contour function. Both images show post-processing of the same model run with 100 000 released particles.

And as previously, the resolution of the visualization grid also affects how smooth the output looks, see Figure 13.

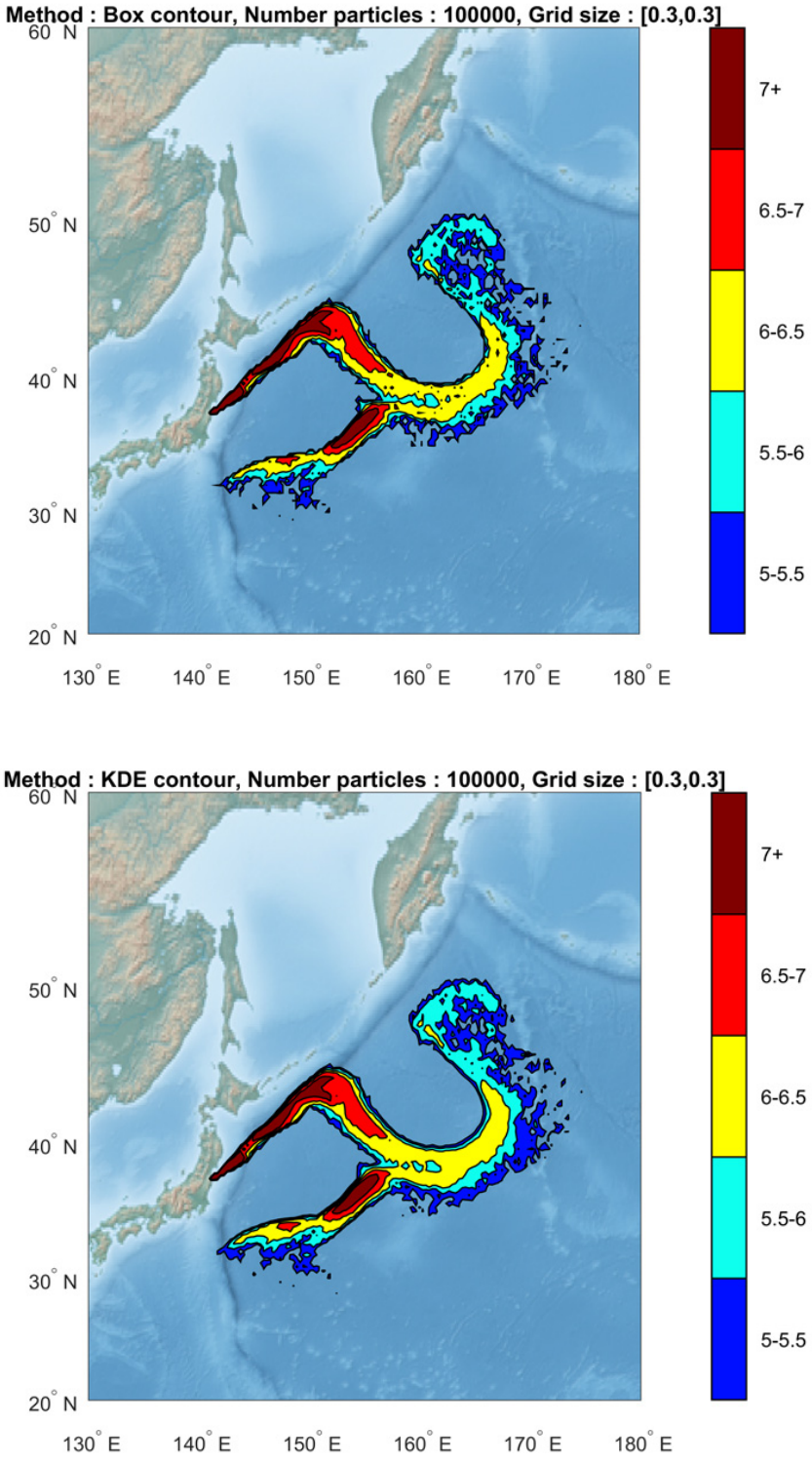


Figure 13. Box-counting (upper image, labelled BOX) versus integrated turbulence KDE (lower image, labelled KDE), both plotted using a contour function but on a coarser visualization grid. Both images show post-processing of the same model run with 100 000 released particles.

5.3 Deposition fields

In this comparison we study the particles from the Fukushima simulations that deposited over the portion of mainland Japan where radioactivity measurements were collected by airborne measurements [3] . We simulated releases of different number of particles and post-processed the resulting deposition fields using box-counting, integrated turbulence KDEs and partition varying bandwidth KDEs respectively. Instead of making a number of simulations with a different amount of released particles we made a single model run with 9 670 558 particles released over 20 days. From this set of particles we picked, randomly and unbiased, six subsets of particles to represent six simulations with fewer released particles. The choice was calibrated to yield an even number of deposited particles. Of the deposited particles a fraction (roughly 24%) deposited over the area of mainland Japan that was studied. The total number of deposited particles and the corresponding amount of deposited particles in the area are presented in Table 1.

Table 1. Total number of model particles deposited and the corresponding number of particles that deposited in the area of mainland Japan that we studied.

Number of released particles	Total number of deposited particles	Number of deposited particles in the area of interest
13 900	10 000	2 428
69 500	50 000	12 026
139 000	100 000	24 001
695 000	500 000	120 447
1 390 000	1 000 000	241 434
6 950 000	5 000 000	1 206 888

To have a reference field, “a truth”, to compare with we have also made a simulation with 700 000 000 particles, out of which nearly 500 000 000 particles deposited in the area. We consider this model result to be the correct answer. It is shown in Figure 14.

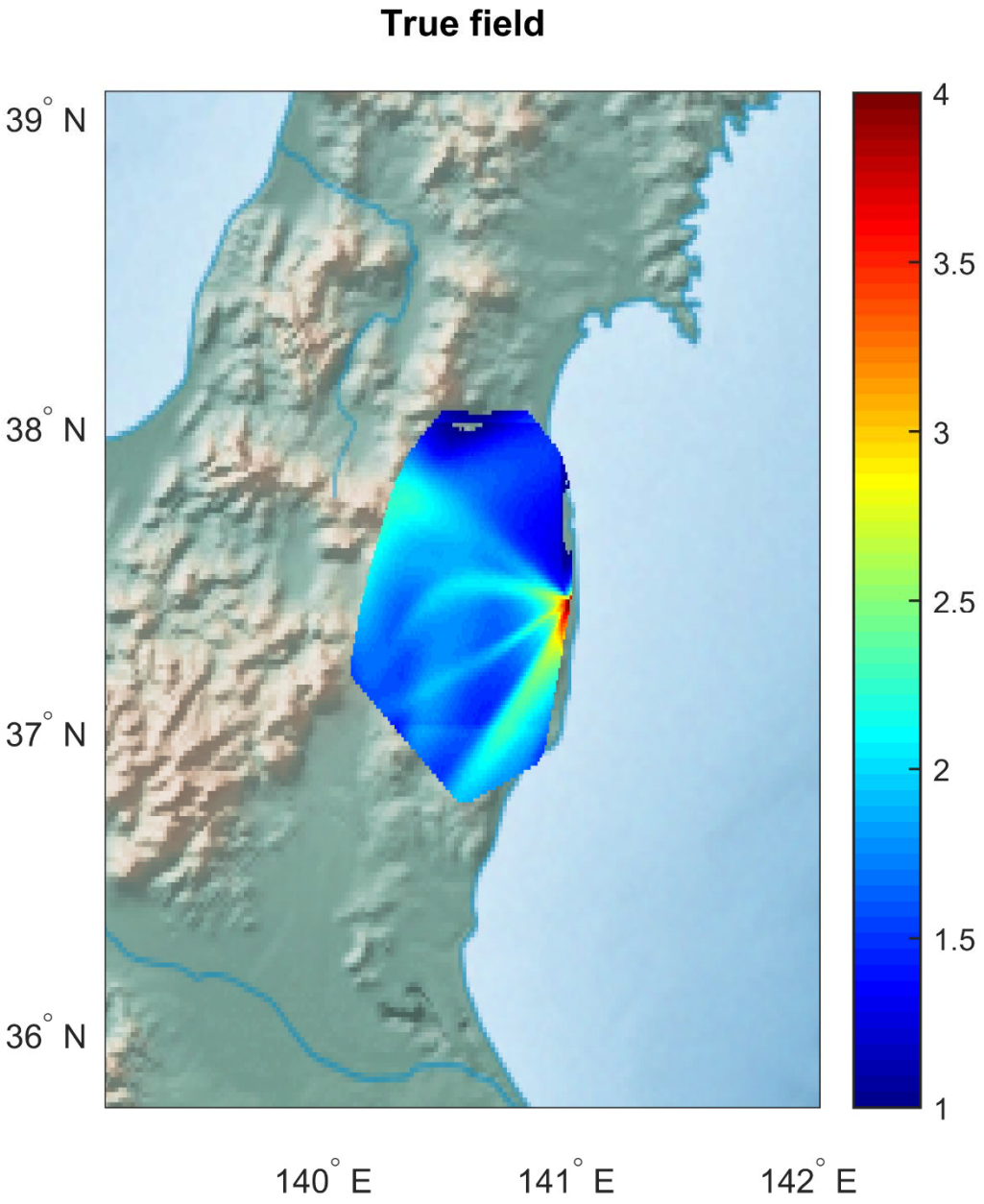


Figure 14. In the figure 500 000 000 model particles have deposited in the area plotted, out of which 24% in the area shown. We consider this result to be the true deposition field.

In Figure 15, Figure 16 and Figure 17 we compare box-counting, integrated turbulence KDEs and partition varying bandwidth KDEs for deposited particles. In the model runs there were 5 000 000, 500 000 and 50 000 model particles that deposited, out of which roughly 24% in the area of interest.

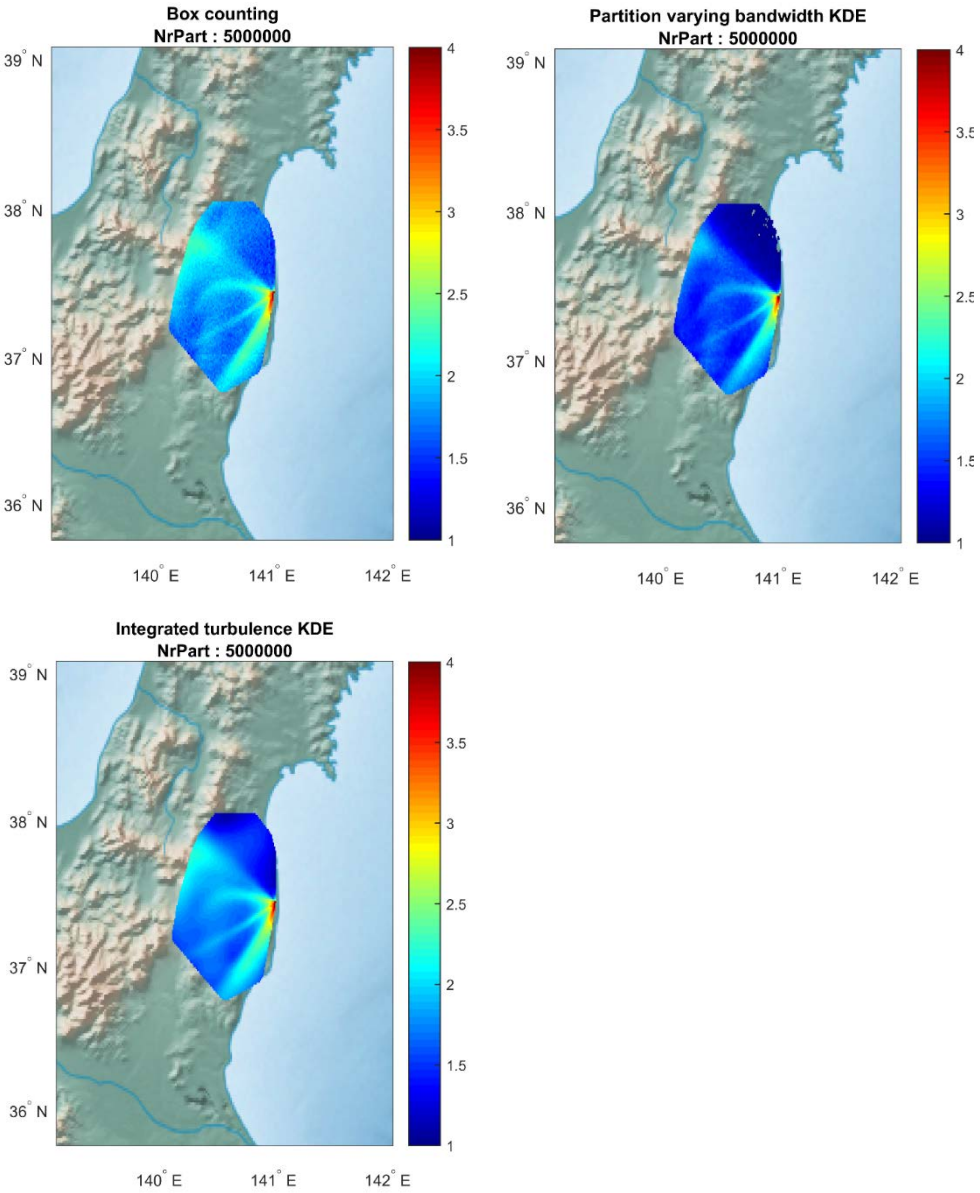


Figure 15. Post-processing of deposition field. There are 5 000 000 deposited model particles, out which 1 206 888 in the shown area. The fields are post-processed using box-counting, partition varying bandwidth KDE and integrated turbulence KDE respectively.

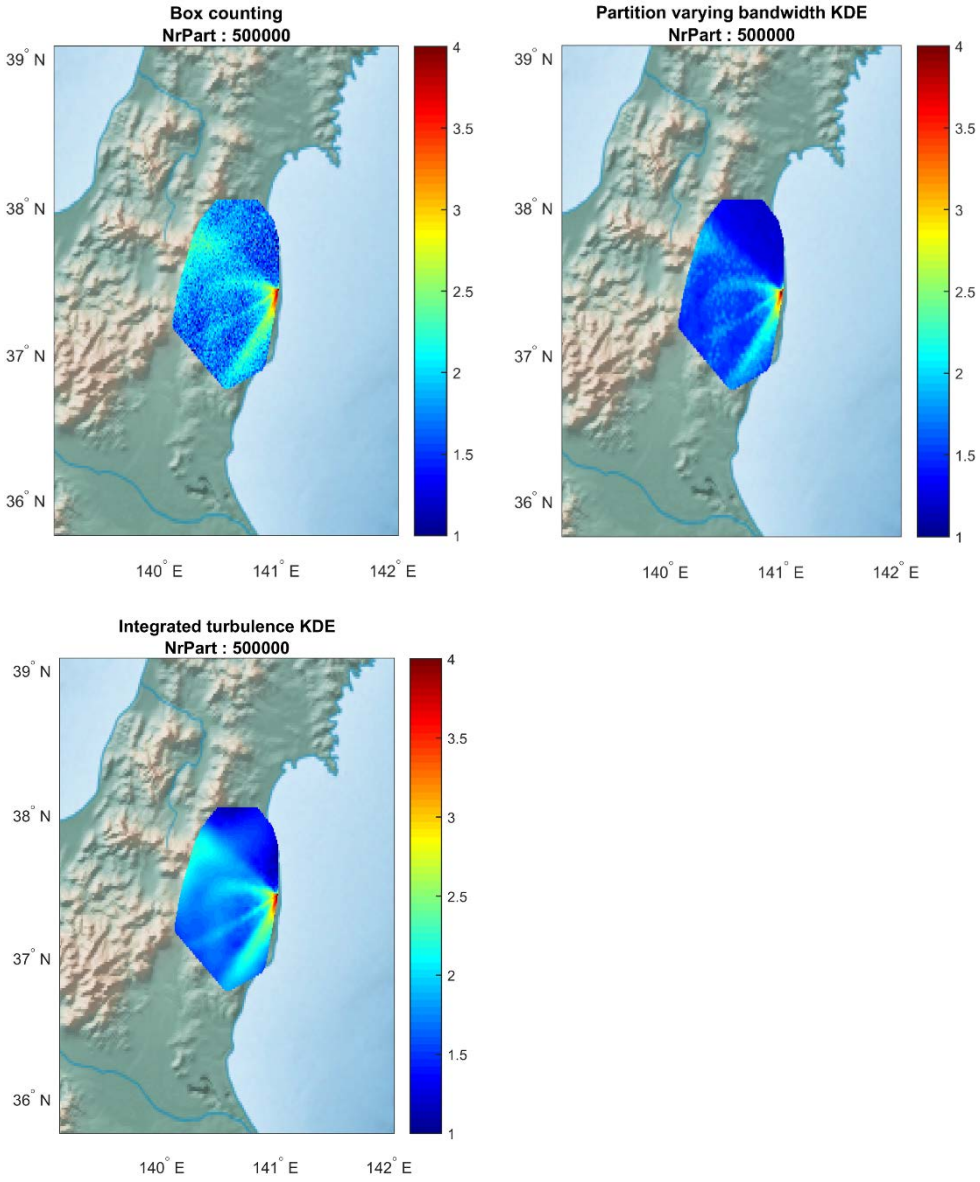


Figure 16. Post-processing of deposition field. There are 500 000 deposited model particles, out which 120 477 in the shown area. The fields are post-processed using box-counting, partition varying bandwidth KDE and integrated turbulence KDE respectively.

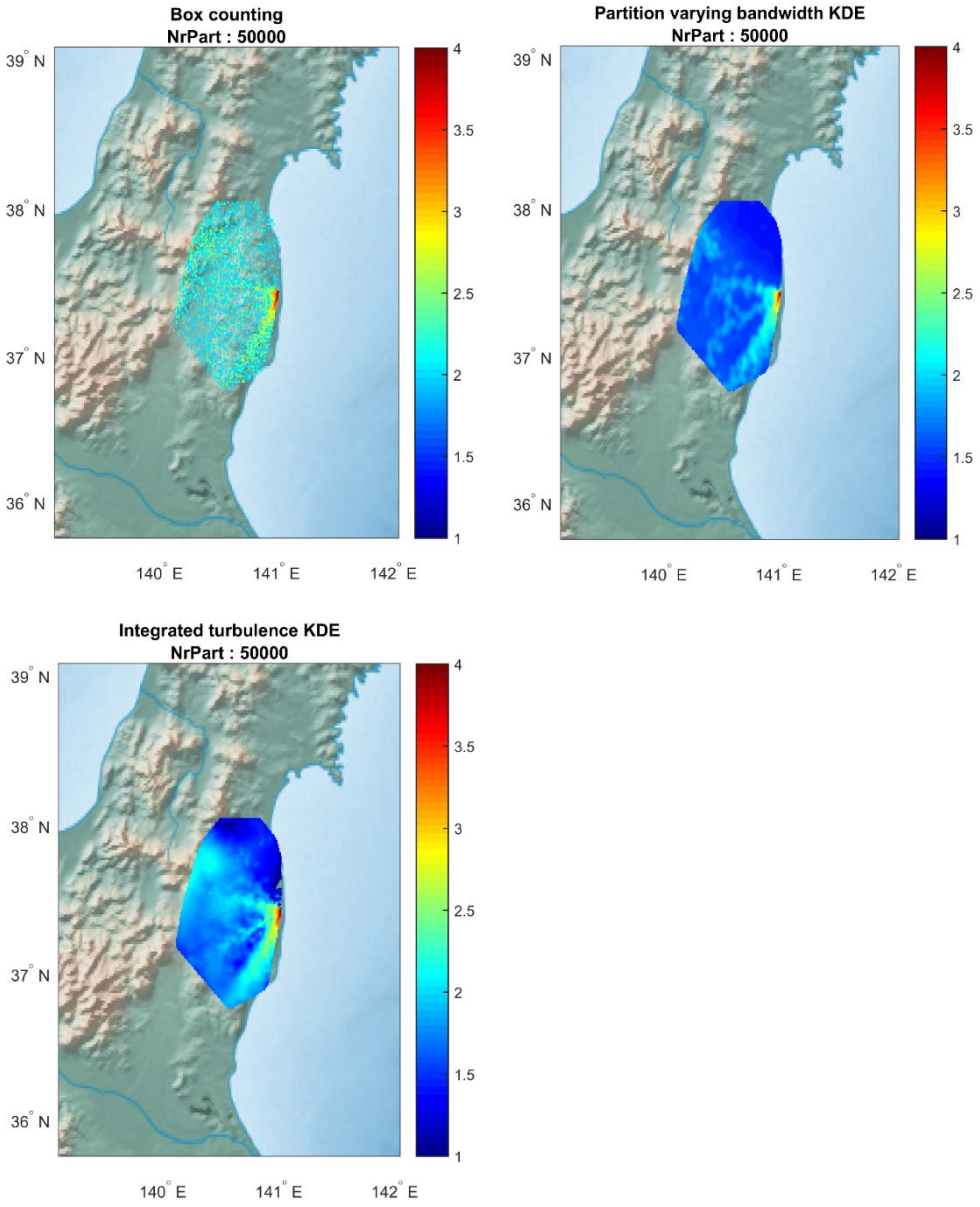


Figure 17. Post-processing of deposition field. There are 50 000 deposited model particles, out which 12 026 in the shown area. The fields are post-processed using box-counting, partition varying bandwidth KDE and integrated turbulence KDE respectively.

5.4 Statistical comparison: integrated mean square error

Visual comparison of concentration fields and depositions fields, like in Figure 8 to Figure 18, gives an intuition of how the different post-processing methods perform. That type of comparison can be complemented by quantitative statistical measures. In this case we have chosen the integrated mean standard error (MSE) as our statistical measure. As the level of deposited material spans a large interval, with the area around the source completely dominating the picture, we have taken the logarithm (base 10) of the estimated deposition values to get a fairer comparison (otherwise, the area around the source would also dominate the statistical measure making the rest of the deposition field more or less irrelevant).

The integrated mean square error of the logarithm (base 10) of the estimated deposition field is presented in Table 2 for each post-processing method for each model run.

Table 2. Integrated MSE of the logarithm (base 10) of the estimated deposition field as a function of released model particles for box-counting, integrated tubulence KDE, partition varying bandwidth KDE.

No dep particles	log(Box)	log(Int turbulence)	log(Partition varying)
10 000	0.492313	-1.66144	-0.0963
50 000	0.209166	-2.17792	-0.11871
100 000	-0.08152	-2.31715	-0.12139
500 000	-1.20793	-2.4375	-0.14601
1 000 000	-1.3886	-2.47413	-0.15342
5 000 000	-1.4797	-2.52612	-0.16364

This data is plotted in Figure 18.

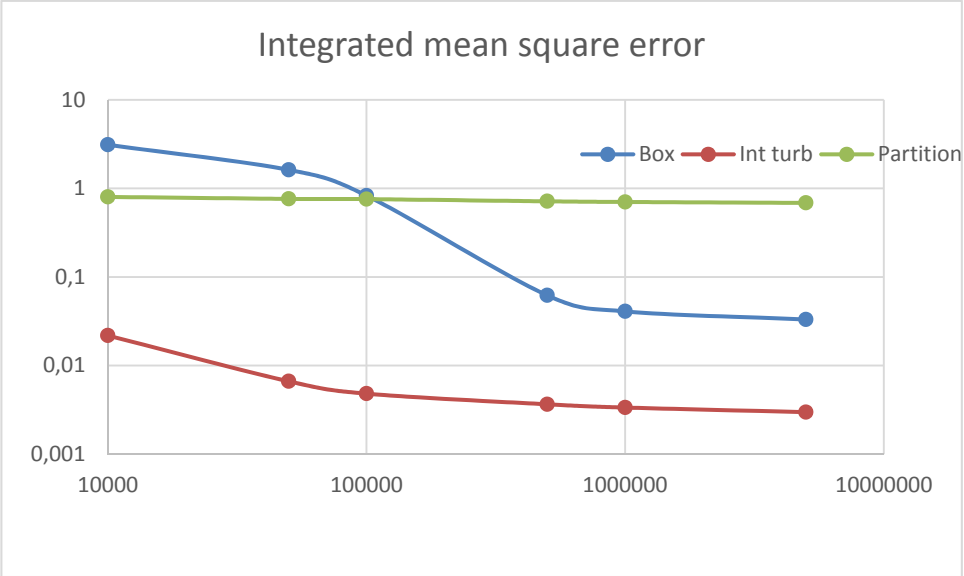


Figure 18. In the plot the logarithm (base 10) of the integrated mean square error of the estimated deposition field is plotted for the three post-processing methods box-counting (Box), integrated turbulence KDE (Int turb) and partition varying bandwidth KDE (Partition).

5.5 Comparison with field data

We originally intended to compare a KDE post-processed deposition field and a box-counting post-processed field with measurements of deposited material from Fukushima [11]. But given that the deposition field around Fukushima depends on both the local geometry and the local weather during the accident and that none of these are resolved by the dispersion model combined with the used weather forecast data [3] we decided that such a comparison is to be postponed until the dispersion model has been developed further. In other words, the deposition measurements from around Fukushima is not a good data set to validate any post-processing method against.

6 Discussion and conclusions

Different methods to post-process data from a Lagrangian dispersion model have been examined quantitatively using the Fukushima Daiichi accident as source term. Box-counting is a commonly used method to compile a concentration field from particle data and is currently used in Pello. Box-counting is a fast method that suffers from noisy results that often can appear unphysical. Two different kernel density estimation methods has been investigated in this work. By the utilization of KDEs it is possible to obtain smoother fields and thereby avoid discrete hotspots. The kernels have spatial widths that determines the smoothness of the concentration field. The two versions of KDE presented here employ two completely different ways of determining the bandwidths.

In the method referred to as *partition varying bandwidth KDE* the widths of the kernels are determined from the distribution of particles. The width scales inversely proportional towards the local density of particles. The advantage of this approach is that the field becomes smooth in low density areas which reduces the noise while the impact of the KDE-treatment is reduced in high density areas to maintain the small-scale details in the field. The drawback is that it is nontrivial to compute this method in a time efficient manner. However, by introducing binned kernel + partition varying bandwidth the method becomes significantly faster and thereby also useful. Note that this method of determining the bandwidth is exclusively based on the particle distribution and has no further correlation to the physics of the dispersion process.

The second method investigated is labelled *integrated turbulence KDE*. In contrast to the *partition varying bandwidth KDE* the physics of the dispersion model is directly implemented to calculate the individual bandwidths based on the path of the particles. The bandwidths are therefore readily derived with no regard to the other particles. This implies that the widths are quickly determined. On the other hand, the concentration field may be more time-consuming to calculate since FFT-schemes are not possible to apply when all particles have different widths.

The three post-processing methods were compared towards an extremely high resolved field that served as a reference field (also called *True field*). There was a distinct difference between box-counting and the two KDE-methods. Box-counting performs poorly especially when there are few particles. In this case the concentration field becomes noisy and difficult to analyze. As the number of particle increases the box-counting method improves as expected. When comparing the two KDE-methods it is clear that the *integrated turbulence* method outperform the *partition varying bandwidth* method. The mean square error in the deposited field shows significantly better results for the former method. The coupling to the physics in the dispersion model also suggests that each model particle is represented more adequately by this approach.

In conclusion, it has been shown that the use of KDE improves the compilation of concentration fields from discrete particle data. The method that yielded the best results is the *integrated turbulence* which in all cases performed well. This

method will be implemented as a standard post-processing method in the dispersion systems at the Swedish Defence Research Agency (FOI). Even though smoother fields in themselves may be a good enough reason to use KDE, the benefit may also take other forms. For instance, the dispersion simulation may be performed with fewer particles and thereby be conducted faster which is a highly desirable outcome. Moreover, given this improvement in performance the model could utilize more advanced physics schemes which could answer new questions and result in a more precise dispersion modelling.

7 References

1. Haan, P.d., *On the use of density kernels for concentration estimations within particle and puff dispersion models.* Atmospheric Environment, 1999. **33**(13): p. 2007-2021.
2. von Schoenberg, P. and L. Thaning, *Våtdeposition av radioaktiva partiklar*, in *FOI Report FOI-R--3818--SE*. 2013, CBRN Environment and protection: FOI.
3. von Schoenberg, P. and H. Grahn, *Våtdeposition av radioaktiva partiklar. Del 2. Implementation*, in *FOI Report, FOI-R--3972--SE*. 2014, FOI.
4. Turlach, B.A., *Bandwidth Selection in Kernel Density Estimation: A Review.* Discussion Paper 9307, 1993.
5. Sørensen, J.H., A. Baklanov, and S. Hoe, *The Danish emergency response model of the atmosphere (DERMA).* Journal of Environmental Radioactivity, 2007. **96**(1–3): p. 122-129.
6. Haan, P.d. and M.W. Rotach, *A novel approach to atmospheric dispersion modelling: The Puff-Particle Model.* Quarterly Journal of the Royal Meteorological Society, 1998. **124**(552): p. 2771-2792.
7. Silverman, B.W., *Density estimation for statistics and data analysis.* Vol. 26. 1986: CRC press.
8. Botev, Z.I., J.F. Grotowski, and D.P. Kroese, *Kernel density estimation via diffusion.* 2010: p. 2916-2957.
9. Lindqvist, J., *En stokastisk partikelmodel I ett ickemetriskt koordinatsystem*, in *FOI Report, FOI-R—99-01086-862-SE*. 1999: FOI.
10. Duong, T., *ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R.* Journal of Statistical Software, 2007. **21**(7).
11. von Schoenberg, P. and H. Grahn, *Dispersion of radioactive material across the northern hemisphere from Fukushima Daiichi Power Plant accident modeled with a random displacement stochastic particle model*, in *FOI Report, FOI-R--3746--SE*. 2013: FOI.
12. Schoenberg v, P., et al., *Atmospheric Dispersion of Radioactive Material from the Fukushima Daiichi NuclearPower Plant*, in *Air Pollution Modeling and its Application XXII*, D.G. Steyn, P.J.H. Builtjes, and R.M.A. Timmermans, Editors. 2014, Springer Netherlands. p. 345-349.
13. Katata, G., et al., *Detailed source term estimation of the atmospheric release for the Fukushima Daiichi Nuclear Power Station accident by coupling simulations of an atmospheric dispersion model with an improved deposition scheme and oceanic dispersion model.* Atmos. Chem. Phys., 2015. **15**(2): p. 1029-1070.

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Defence Research Agency
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se