ESAIM: PROCEEDINGS, August 2008, Vol. 24, p. 97-110
C. Dobrzynski, P. Frey, Ph. Pebay, Editors

SOLVING THE UNIFORM DENSITY CONSTRAINT IN A STOCHASTIC DOWNSCALING MODEL

C. Chauvin[1], S. A. Hirstoaga[2], P. Kabelikova[3], A. Rousseau[4], F. Bernardin[5] and M. Bossy[6]

Abstract. The present work aims to contribute to the development of a numerical method to compute small scale phenomena in atmospheric models, getting rid of any mesh refinement. In a domain, typically a mesh of a numerical weather prediction model, we simulate some particles that are moved thanks to a Stochastic Lagrangian model adapted from the PDF methods proposed by S.B. Pope. We estimate the Eulerian values of the required fields, thanks to the computation of a local mean value over an ensemble of particles. We are thus using a stochastic particle method. At small scale, our atmospheric model imposes that the mass density ρ is constant in the domain. As a consequence, the particles have to be uniformly distributed at every time step of the particle method. We aim to use D.P. Bertsekas *Auction Algorithm* in order to move a given cloud of particles to a new position, which is also given in advance, and that realizes the constraint $\rho = cst$. Naturally, the transport cost will have to be minimum. This is a problem of 3D optimal transport, which is known to be difficult.

Résumé. Ce travail a pour cadre le développement d'une méthode de simulation d'écoulement atmosphérique aux petites échelles, réalisant des calculs sous-maille sans avoir besoin de raffiner le maillage. Dans un domaine, typiquement une maille d'un modèle numérique d'écoulement géophysique, on simule un ensemble de particules dont la dynamique suit un modèle stochastique lagrangien adapté des méthodes PDF proposées par S.B. Pope. On estime les valeurs eulériennes en moyennant localement les valeurs sur les particules. Il s'agit donc d'une méthode particulaire stochastique qui, on l'espère, sera comparable ou même meilleure, à des méthodes plus classiques de raffinement de maillage. Dans le domaine considéré, le modèle contraint la densité massique à rester constante. Cette contrainte se reporte sur la position des particules dont la distribution doit rester uniforme à chaque étape de la méthode particulaire. Le travaille présenté ici consiste à mettre en place l'algorithme des enchères (*Auction Algorithm*) développé par D.P. Bertsekas, pour permettre de déplacer les particules dans le domaine de calcul, en conservant leur distribution uniforme et en minimisant le coût de transport. Autrement dit, c'est un problème numérique de transport optimal 3D.

[1] Grenoble High Magnetic Field Laboratory, 25 ave des Martyrs, 38042 Grenoble, Claire.Chauvin@grenoble.cnrs.fr.

[2] CEREMADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris, France, sever@ceremade.dauphine.fr

[3] Dpt of Applied Mathematics, Faculty of electrical engineering and informatics (FEI), VSB-Technical University of Ostrava, Tr. 17. listopadu 15, CZ 708 33 Ostrava-Poruba, Czech Republic, pavla.kabelikova@vsb.cz

[4] INRIA, Team MOISE-LJK, 51 rue des mathématiques, 38041 Grenoble, France, Antoine.Rousseau@inria.fr

[5] Centre d'Études Techniques de l'Équipement de Lyon, Laboratoire régional des ponts et chaussées, 8-10 rue Bernard Palissy Z.I. du Brézet, 63017 CLERMONT-FERRAND Cédex 2,Frederic.Bernardin@equipement.gouv.fr

[6] INRIA, TOSCA Project, 2004 route des lucioles, 06902 Sophia Antipolis, France, Mireille.Bossy@inria.fr.

Contents

Introduction

This work aims to contribute to the development of a numerical tool, called *Stochastic Downscaling Method* (*SDM*), that was initially proposed in [19]. It consists in introducing a new method for the simulation of the wind at small scales.
The large scale behavior of the ground wind is known to be of major importance for our society, in particular in the context of global warming. But the local wind variability is also a key mechanism, for instance when one wishes to build a bridge, or optimize the location of a wind turbine. Ground winds also vary at small time and space scales, that are unfortunately far smaller than those solved by classical weather/climate forecasting models. Thus, there is a need to imagine and propose some new numerical techniques that would ensure a satisfying prediction of the wind at sufficiently small scales: this is the general objective of the *downscaling* techniques.

Whereas deterministic refinement methods, based on *a posteriori* estimates [20], are already well developed, some of the authors introduced *SDM* [19], an algorithm of a totally new type in the framework of downscaling methods, based on a stochastic Lagrangian model, solved by a particle method (see Section 1.1 below). Indeed, given the coarse prediction provided by a Numerical Weather Prediction model, *SDM* simulates a set of fluid particles in a chosen domain of computation, in order to refine the numerical prediction over it. For the time being, the vertical size of the *SDM* domain is sufficiently small to consider a constant mass density. This implies that the numerical particles must satisfy the uniform density constraint in the domain. At each time step, the rearrangement of the particles under minimal displacement is the most expensive procedure, in term of number of operations. To this aim, we evaluate in this paper the performances of the *Auction Algorithm*, introduced in [6], and its ability to be coupled with *SDM*.
At this time, *SDM* uses methods expected to be nearly optimal whereas the *Auction Algorithm* is optimal. This tool will be our reference algorithm; in this paper, we validate the heuristic parameters of *Auction Algorithm*, and test its complexity on several configurations of particles.

This article is organized as follows: we start in Section 1.1 with a general description of Lagrangian stochastic models, before introducing in 1.2 the constraint of uniform density, required by the model. Then, in Section 2, we present the *Auction Algorithm*. Finally, Section 3 provides some numerical results, and analyzes the claimed properties of the *Auction Algorithm* thanks to different experiments.

1. Framework of the Stochastic Downscaling Method

1.1. **Stochastic Lagrangian Models**

We now give a general description of Lagrangian stochastic models, in order to introduce the specific problem tackled in this paper, that is the uniform density constraint. For more details, the reader is referred to [16, 17] and the related bibliography.

We are interested in the behavior of a *turbulent incompressible fluid* in an open set $\mathcal{D}$ of $\mathbb{R}^3$, with constant mass density ρ. In the statistical description of turbulence, all the physical fields (here the velocity U and the pressure $\mathcal{P}$) are considered as random fields: they can be written, following the so-called Reynolds decomposition, as the sum of a deterministic part and a random turbulent part, namely:

$$\begin{aligned} U(t,x,\omega) &= \langle U\rangle(t,x) + u(t,x,\omega), \\ \mathcal{P}(t,x,\omega) &= \langle \mathcal{P}\rangle(t,x) + p(t,x,\omega). \end{aligned}$$

Applying the Reynolds operator to the incompressible Navier Stokes equations, we obtain the so-called Reynolds Averaged Navier Stokes (RANS) equations:

$$\begin{aligned} \partial_t\langle U^{(i)}\rangle(t,x) &+ \left(\langle U\rangle(t,x).\nabla\right)\langle U^{(i)}\rangle(t,x) \\ &= \nu\triangle\langle U^{(i)}\rangle(t,x) - \frac{1}{\rho}\partial_{x_i}\langle\mathcal{P}\rangle(t,x) - \sum_{k=1}^{3}\partial_{x_k}\langle u^{(i)}u^{(k)}\rangle(t,x), \qquad (1.1a) \\ \nabla\cdot\langle U\rangle(t,x) &= 0, \qquad (1.1b) \end{aligned}$$

where $U^{(i)}$ stands for the ith component of U, with $i = 1..3$, and $\langle\mathcal{P}\rangle$ is solution of the following Poisson equation:

$$-\frac{1}{\rho}\Delta_x\langle\mathcal{P}\rangle = \sum_{i,j=1}^{3}\left(\partial_{x_j}\langle U^{(i)}\rangle\partial_{x_i}\langle U^{(j)}\rangle + \partial^2_{x_ix_j}\langle u^{(i)}u^{(j)}\rangle\right). \qquad (1.2)$$

The matrix $\{\langle u^{(i)}u^{(j)}\rangle\}_{1\leq i,j\leq 3}$, called the Reynolds tensor, is defined by:

$$\langle u^{(i)}u^{(j)}\rangle = \langle U^{(i)}U^{(j)}\rangle - \langle U^{(i)}\rangle\langle U^{(j)}\rangle \;\forall 1\leq i,j\leq 3.$$

The system (1.1) is not closed, and hence one needs to provide additional information on the Reynolds tensor, in order to close the system and compute some solution. Different ways of closing system (1.1) have been introduced over the years: for example $k-\varepsilon$ closure, or second order Rotta closure, etc. The interested reader is referred to [14] for further details on the closure of the RANS equations. If we denote by $f_E(t,x;\cdot)$ the law of the random variable $U(t,x,\omega)$, it is to be noticed that a closure on f_E is equivalent to the RANS closure.

From 1985, S.B. Pope started to introduce some completely different way of closing the RANS equations. His idea consists in describing the fluid as an ensemble of Lagrangian particles, introducing a stochastic model, whose state variable is $(X_t, \mathcal{U}_t, \phi_t)$, that respectively correspond to the position, the velocity and, possibly, some other physical quantities attached to a fluid particle, such as a temperature or a turbulent frequency. The dynamics of the fluid particle is ruled by a system of stochastic differential equations (SDEs) of the form

$$dX_t = \mathcal{U}_t dt, \qquad (1.3a)$$

$$d\mathcal{U}_t = -\frac{1}{\rho}\nabla_x\langle\mathcal{P}\rangle(t,X_t)dt + D_{\mathcal{U}}(t,X_t,\mathcal{U}_t,\phi_t)dt + B_{\mathcal{U}}(t,X_t,\mathcal{U}_t,\phi_t)dW_t, \qquad (1.3b)$$

$$d\phi_t = D_\phi(t,X_t,\mathcal{U}_t,\phi_t)dt + B_\phi(t,X_t,\mathcal{U}_t,\phi_t)d\tilde{W}_t, \qquad (1.3c)$$

where $(W_t, t \in \mathbb{R}^+)$ and $(\tilde{W}_t, t \in \mathbb{R}^+)$ are two independent Brownian motions. The system (1.3) intends to model f_E and the laws governing the Eulerian fields $U(t,x,\omega)$ and $\phi(t,x,\omega)$ in the sense that $\mathcal{U}_t = U(t, X_t, \omega)$ and $\phi_t = \phi(t, X_t, \omega)$.

The design of the coefficients D_* and B_* (the subscript $*$ holds for $\mathcal{U}$ and ϕ) in equations (1.3b) and (1.3c) determines the modelling context and its closure: for *SDM*, their expressions are given in [3]; for engineering applications such as turbulent fluids in combustion, or models including the temperature, see [10, 11]. These coefficients D_* and B_* depend on statistics of the state variables $(X_t, \mathcal{U}_t, \phi_t)$ (for instance the mean velocity, the turbulent kinetic energy, the dissipation), according to the model closure. Hence those coefficients D_*, B_* are non linear in the sense of McKean [12].

Assume the existence of the density $f_L(t; x, V, \theta)$ for the Lagrangian variables, then at each time t, the measure $f_L(t; x, V, \theta)dxdVd\theta$ is the law of $(X_t, \mathcal{U}_t, \phi_t)$ solution of (1.3). Considering an incompressible fluid, with constant density ρ, the link between $f_E(t, x; V, \theta)$ and $f_L(t; x, V, \theta)$ (see *e.g.* [15] for details) is:

$$f_E(t, x; V, \theta) = \frac{f_L(t; x, V, \theta)}{\int_{\mathbb{R}^3 \times \mathbb{R}^3} f_L(t; x, V, \theta) dV d\theta}, \tag{1.4}$$

which allows to compute any mean fields required in D_* and B_* by the formula:

$$\langle Q(U, \phi) \rangle (t, x) = \int_{\mathbb{R}^3 \times \mathbb{R}^3} Q(V, \theta) f_E(t, x; V, \theta) dV d\theta.$$

Moreover, and to end this part, a fluid particle satisfying (1.3) and (1.2) also satisfies [8] :

$$\int_{\mathbb{R}^3 \times \mathbb{R}^3} f_L(t; x, V, \theta)\ dV\ d\theta = \rho, \tag{1.5a}$$

$$\nabla_x \cdot \langle U \rangle (t, x) = 0. \tag{1.5b}$$

The pressure term in (1.3b) produces these two equations (1.5). In *SDM*, we simulate the equations (1.3) without the pressure term (prediction step), and then correct the fluid density and the mean velocity according to (1.5). Since the coefficients D_* and B_* depend on the law governing the fluid particle in a non-local way (mean field interaction), a stochastic particle method is required to discretize (1.3). Next section firstly introduces the *Particle in Cell* method, and the splitting scheme (prediction + correction). More particularly, the aim of this work is to present a method to study and solve the constraint (1.5a), called the uniform density constraint.

1.2. Uniform Density Constraint

In a parallelepipedic domain $\mathcal{D}$, we present the discretization scheme to solve the splitted problem:

(1) The dynamics of each particle k is characterized at time $t_n = t_0 + n\Delta t$ by the discrete variables $(X_k^n, \mathcal{U}_k^n, \phi_k^n)$. The coefficients D_* and B_* are non-linear, and depend on each particle k, and on the statistics given by the N particles.
(2) Prediction step: solve the discretized version of (1.3) without the pressure term
(3) Correction step: modify the state variables in order to satisfy (1.5).

In the context of building the numerical tool *SDM* that was mentioned above, we have to deal with a parallelepipedic domain $\mathcal{D}$.

In this domain, we describe equations (1.3) (without the pressure term) thanks to a system of N interacting particles. The dynamics of each particle k is characterized at time $t_n = t_0 + n\Delta t$ by the discrete variables $(X_k^n, \mathcal{U}_k^n, \phi_k^n)$. The operators $D_{\mathcal{U}}$ and $B_{\mathcal{U}}$ of equation (1.3b) are then determined at time t_n by $D_{\mathcal{U}}^n = D_{\mathcal{U}}(t_n, \{X_k^n\}, \{\mathcal{U}_k^n\}, \{\phi_k^n\})$, $B_{\mathcal{U}}^n = B_{\mathcal{U}}(t_n, \{X_k^n\}, \{\mathcal{U}_k^n\}, \{\phi_k^n\})$; the same holds for operators D_ϕ^n and B_ϕ^n of equation (1.3c). The discretized system at time t_{n+1} is the following :

$$X_k^{n+1} = X_k^n + \mathcal{U}_k^n \, \Delta t, \tag{1.6a}$$
$$d\mathcal{U}_k^{n+1} = D_{\mathcal{U}}^n \, dt + B_{\mathcal{U}}^n \, dW, \tag{1.6b}$$
$$d\phi_k^{n+1} = D_\phi^n \, dt + B_\phi^n \, d\widetilde{W}, \tag{1.6c}$$

where $(W_t, t \in \mathbb{R}^+)$ and $(\widetilde{W}_t, t \in \mathbb{R}^+)$ are two independent Brownian motions. The non-linear coefficients B^{n+1} and D^{n+1} are determined by the solutions $\{X_k^{n+1}\}, \{\mathcal{U}_k^{n+1}\}, \{\phi_k^{n+1}\}$ and their associated averaged Eulerian quantities $\langle \mathcal{U} \rangle^{n+1}, \langle \phi \rangle^{n+1}$. Without going into details, these quantities are computed from the Lagrangian ones $\{\mathcal{U}_k^{n+1}\}, \{\phi_k^{n+1}\}$ using a local approximation, as in the *Particle in Cell* (PIC) method, see [18].

Among the available PIC interpolations, we use the so-called *Nearest Grid Point* (NGP) method, which consists in dividing the domain $\mathcal{D}$ into cells (see Figure 1), and in evaluating the Eulerian quantities, as well as the fluid density, at a point $x \in \mathcal{D}$ by a Monte-Carlo approximation which uses all the particles k belonging to the cell $\mathcal{C}, x \in \mathcal{C}$.

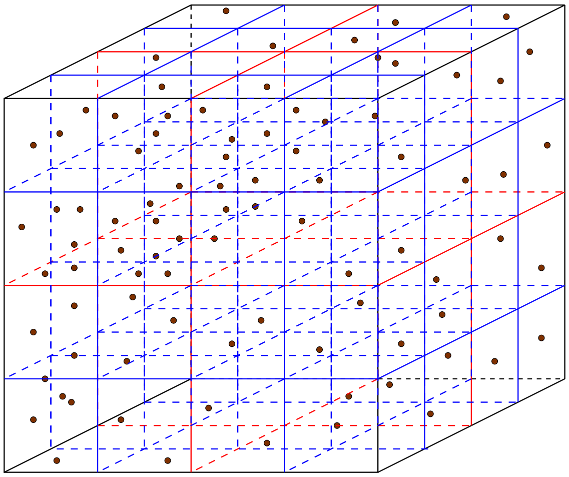

Figure 1. Fluid particles in different cells of $\mathcal{D}$.

After proceeding to the advancement step (1.6), the equations (1.5) have to be fulfilled by the system of particles. We focus here on the the uniform density constraint (1.5a), which consists in ensuring that each cell of the discretized domain (see Figure 1) has the same number of particles. To ensure a good convergence of the approximation in each cell, the number of particles in each cell N_{pc} has to be important (greater than 1000). That is, given a set of N particles of positions $(X_i)_{1\leq i\leq N}$, we have to move them to some new positions $(Q_j)_{1\leq j\leq N}$, such that every cell of the domain contains N_{pc} particles.

The choice of final positions $(Q_j)_{1\leq j\leq N}$ is obviously not unique, but they have to follow the uniform law: in *SDM*, the positions $(Q_j)_{1\leq j\leq N}$ are constructed using the uniform law in each cell, such that all the cells contain exactly N_{pc} particles.

In addition, we use the physical properties of the Lagrangian particles $(\mathcal{U}_k, \phi_k)$ to compute the Eulerian fields (e.g. $\langle U \rangle^n$) (see [19] and Equation (1.4) above). Consequently, the particles should not move too far from their initial position during the transportation problem, in order to maintain locally the physical information they transport.

That is why we need the *transport cost* from $(X_i)_{1\leq i\leq N}$ to $(Q_j)_{1\leq j\leq N}$ to be minimum; if we define the distance between an initial position and a final one:

$$d_{ij} = |X_i - Q_j|^2,$$

where $|\cdot|$ is the Euclidean norm of $\mathbb{R}^3$, then we are faced with a classical problem of optimal transport:

$$(P) \quad \text{Find } \ j \in \mathcal{S}_N \ \text{ such that} \sum_{i=1}^{N} d_{ij(i)} = \min_{k \in \mathcal{S}_N} \sum_{i=1}^{N} d_{ik(i)}, \tag{1.7}$$

where $\mathcal{S}_N$ stands for the set of permutations over $\{1,\ldots,N\}$. More explicitly, we want to find a *feasible assignment*, i.e., a set of pairs $\{(X_1, Q_{j(1)}),\ldots,(X_N, Q_{j(N)})\}$ such that $j(1),\ldots,j(N)$ are all distinct, that minimizes the overall cost $\sum_{i=1}^{N} d_{ij(i)}$. This is an instance of the classical *assignment problem* appearing in network optimization. Both from the continuous and discrete viewpoints, the optimal transport problem in dimension 3 is known to be challenging (see [2,9,13]), whereas in dimension 1, it simply consists (in the discrete case) to sort the particles [2], with a computational cost of $\mathcal{O}(N\log(N))$, where N is the number of particles.
The assignment problem (P) can be understood as a search, among all the $N!$ permutations, of one permutation (possibly not unique) satisfying (1.7). Obviously, the aim is to solve the problem using a method with an efficient computational cost rather than factorial. In this direction, several algorithms have been developed and the general upper bound $\mathcal{O}(N^3)$ for the complexity was shown (see the simplex method of [1] or the so-called Hungarian method).

The method we use to reach the solution of our particular assignment problem is the *Auction Algorithm* developed by D.P. Bertsekas, since its practical performance is competitive with state-of-the-art codes (see [6] for a short presentation). The aim of this work is to study the *Auction Algorithm* in our case and its complexity (we call complexity the total number of unitary assignments (X_i, Q_j) during the auction process), and to see in which conditions it is an interesting algorithm for our application.

2. Correction of Particle Positions

In this section, we present the *Auction Algorithm* and some of its improvements, introduced by Bertsekas.

2.1. Naive Auction Algorithm

Having the form of a maximization problem, the classical assignment problem deals with benefits rather than costs. More precisely, there are N persons and N objects that we have to match on a one-to-one basis supposing that there is a benefit a_{ij} for matching person i with object j. The aim is to assign persons to objects so as to maximize the total benefit (see [6]). Mathematically, the problem is:

$$\text{find } \ j \in \mathcal{S}_N \ \text{ such that } \ \sum_{i=1}^{N} a_{ij(i)} = \max_{k \in \mathcal{S}_N} \sum_{i=1}^{N} a_{ik(i)}. \tag{2.1}$$

Setting by $d_{ij} = -a_{ij}$, i.e., passing from benefits to costs, the assignment problem (2.1) is equivalent to our problem (1.7).

Suppose that position Q_j has a price p_j in the sense that we must pay p_j so as one particle can acquire this position. Thus, we define the total cost for moving one particle from initial position X_i to final position Q_j as

$$p_j + d_{ij}, \tag{2.2}$$

where d_{ij} is the transport cost. The algorithm seeks to assign each X_i to some $Q_{j(i)}$ with minimum total cost, i.e.,

$$\begin{aligned} &\text{for every } \ i \in \{1,\ldots,N\} \ \text{ find } \ j(i) \in \{1,\ldots,N\} \\ &\qquad\qquad \text{such that } \quad p_{j(i)} + d_{ij(i)} = \min_{k \in \{1,\ldots,N\}} \big(p_k + d_{ik}\big). \end{aligned} \tag{2.3}$$

The algorithm starts with the empty assignment and the zero price vector. At the beginning of each iteration there are a partial assignment and a price vector. The algorithm is applied until all $(X_i)_{1\le i\le N}$ become assigned.

Algorithm 2.1.

```
Step 1. Select randomly an unassigned initial position $X_i$.
Step 2. Find a position $Q_{j(i)}$ which offers minimum cost for $X_i$, i.e.,
        $$\textit{find } j(i) \in \{1,\dots,N\} \textit{ such that } j(i) \in \operatorname*{Argmin}_{k\in\{1,\dots,N\}} \big(p_k + d_{ik}\big).$$
Step 3. Compute the bidding increment
        $$\overline{\gamma}_i = -\overline{v}_i + \overline{w}_i \qquad (2.4)$$
    where
        $$\overline{v}_i = \min_{k\in\{1,\dots,N\}} \big(p_k + d_{ik}\big) \quad \textit{and} \quad \overline{w}_i = \min_{k\in\{1,\dots,N\}, k\neq j(i)} \big(p_k + d_{ik}\big). \qquad (2.5)$$
Step 4. Increase the price of position $Q_{j(i)}$ by $\overline{\gamma}_i$ and assign $Q_{j(i)}$ to $X_i$.
Step 5. The initial position that were assigned to $Q_{j(i)}$ at Step 2 becomes unassigned and then go
        to Step 1.
```

Each iteration of the algorithm can be seen as a real auction where the bidder X_i searches for his preferred object $Q_{j(i)}$ and the second best object. Then X_i raises the price of $Q_{j(i)}$ by $\overline{\gamma}_i$ since at this stage $Q_{j(i)}$ rests his preferred object but it becomes less attractive to the others bidders.

2.2. Auction Process

Unfortunately, as shown in [5,6], the *Naive Auction Algorithm* (Algorithm 2.1) does not work when two or more final positions offer the same total cost for some given particle X_i. In this case, the bidding increment $\overline{\gamma}_i$ is zero, thereby creating a never ending cycle (see [5, Section 1.3.3]). To break such cycles, a perturbation mechanism is needed. Precisely, for a given small $\varepsilon > 0$, we will not look for fulfilling the conditions (2.3) any more, but the weaker ones, the so-called ε-complementary slackness,

$$\begin{aligned} &\text{for every } \ i \in \{1,\dots,N\} \ \text{ find } \ j(i) \in \{1,\dots,N\} \\ &\qquad\qquad\qquad\qquad \text{such that } \quad p_{j(i)} + d_{ij(i)} \le \min_{k\in\{1,\dots,N\}} \big(p_k + d_{ik}\big) + \varepsilon. \end{aligned} \qquad (2.6)$$

The *Auction Algorithm* is the same as the *Naive Auction Algorithm*, except that the bidding increment is

$$\overline{\gamma}_i = -\overline{v}_i + \overline{w}_i + \varepsilon, \qquad (2.7)$$

rather than $\overline{\gamma}_i = -\overline{v}_i + \overline{w}_i$ as in (2.4).

Remark 2.2. To make the connection with the assignment problem and the auction algorithm as they are presented in [6], we recall that we pass from our formulation to the classical one by taking $a_{ij} = -d_{ij}$.

- The cost formula (2.2) corresponds to the net value of object j for person i.
- The min-equation in (2.3) is exactly [6, Equation (1)].
- The costs defined in (2.5) correspond respectively to the best object value v_i and second best object value w_i introduced in [6, Equation (5)&(6)]. Precisely,

$$\overline{v}_i = -v_i \quad \text{and} \quad \overline{w}_i = -w_i.$$

Therefore, the increment $\overline{\gamma}_i$ in (2.7) is equal to that in [6, Equation (8)].

- The condition in (2.6) is equivalent to its classical counterpart [6, Equation (7)].

The next result establishes the finite convergence of the *Auction Algorithm* (Algorithm 2.1 with (2.7) instead of (2.4)) to an approximate solution of problem (1.7).

Proposition 2.3. *The Auction Algorithm terminates in a finite number of iterations with a feasible assignment and a price vector that satisfy the ε-complementary slackness (2.6). Moreover, the overall cost of this assignment is within $N\varepsilon$ of being optimal, i.e., if $\{(X_1, Q_{j(1)}), \dots, (X_N, Q_{j(N)})\}$ is the final assignment of the algorithm and $D^* = \min_{k \in \mathcal{S}_N} \sum_{i=1}^{N} d_{ik(i)}$ is the optimal cost, then*

$$D^* \leq \sum_{i=1}^{N} d_{ij(i)} \leq D^* + N\varepsilon. \tag{2.8}$$

Proof. By Remark 2.2, Algorithm 2.1 with (2.7) instead of (2.4) takes the form of the auction process described in [6]. Then, it suffices to apply [5, Proposition 7.1], [5, Proposition 1.4], and [5, Proposition 7.2]. □

Remark 2.4. Suppose that the costs d_{ij} are all integers. In this case, if

$$\varepsilon < \frac{1}{N}, \tag{2.9}$$

it follows from (2.8) that the assignment obtained at the end of the *Auction Algorithm* is optimal, i.e., it is the solution of problem (1.7).

2.3. An Improved Algorithm : the ε-Scaling

As explained in [5], one can improve the complexity of the *Auction Algorithm* by introducing the ε-scaling: rather than applying the algorithm for a fixed ε, one starts with a large value ε_0 and then successively reduce it up to some final ε_{K+1}, sufficiently small (like that in (2.9) for example). In addition, for each $k \in \{1, \dots, K\}$, successive auctions with $\varepsilon = \varepsilon_k$ are done for $N - N_k$ particles among the N such that $(N_k)_k$ is decreasing until 0.

Algorithm 2.5.

Step 0. All the particles are unassigned, the price vector is zero, ε_0 and $\theta > 1$ are fixed. Let $k = 1$.

Step 1. Evaluate the number of particles to be assigned : $N - N_k$, where $N_k = \left[\frac{N}{\theta^k}\right]$.

Step 2. Use the Auction Algorithm with $\varepsilon_k = \frac{\varepsilon_0}{\theta^k}$, until $N - N_k$ particles are assigned.

Step 3. Keep the price vector $\{p_j^k\}_{1 \leq j \leq N - N_k}$ associated to the set of $N - N_k$ assignments among the N. These prices are used as input of iteration $k+1$. While $N_k > 0$ go to Step 1.

Setting $\theta > 1$, let $K > 0$ be the greatest integer such that $\theta^K \leq N$. Hence there exists $q \in \mathbb{N}$ such that $q < \theta^{K+1} - \theta^K$ and

$$N = \theta^K + q. \tag{2.10}$$

Then, it is easy to see that Algorithm 2.5 stops at iteration $K+1$ and that (the overall cost of) the final assignment is optimal within $N\varepsilon_{K+1}$. Therefore, the optimal solution is found for $\varepsilon_{K+1} < \frac{1}{N}$ in the case of integer costs d_{ij} (see Remark 2.4). The case of real distances will be discussed in Section 3.

The main idea of the ε-scaling algorithm is to find an approximated solution for a coarse ε_0 (for which the prices increase rapidly at the beginning of the algorithm), and then to successively refine the solution with new (smaller) values of ε_k. The final assignment, obtained with ε_K, is exactly the same with or without considering the ε-scaling algorithm. But the claim is that the new algorithm should lower the total number of computational operations.

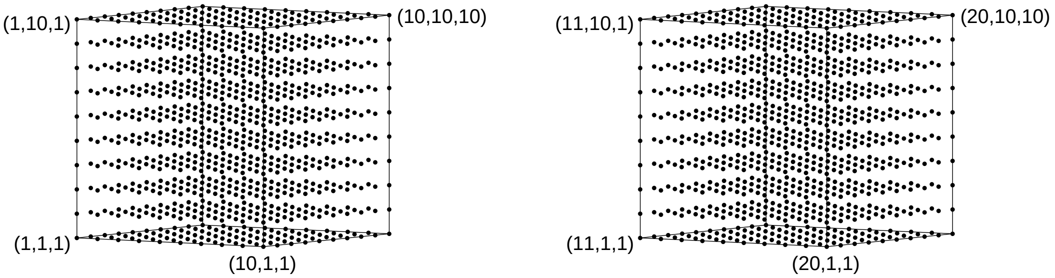

FIGURE 2. Test case 1. Both input set $(X_i)_{1\leq i\leq N}$ and output set $(Q_j)_{1\leq j\leq N}$ are uniform, for 1000 particles. $(Q_j)_{1\leq j\leq N}$ is the result of a shift of $(X_i)_{1\leq i\leq N}$along the $x-$coordinates by 10.

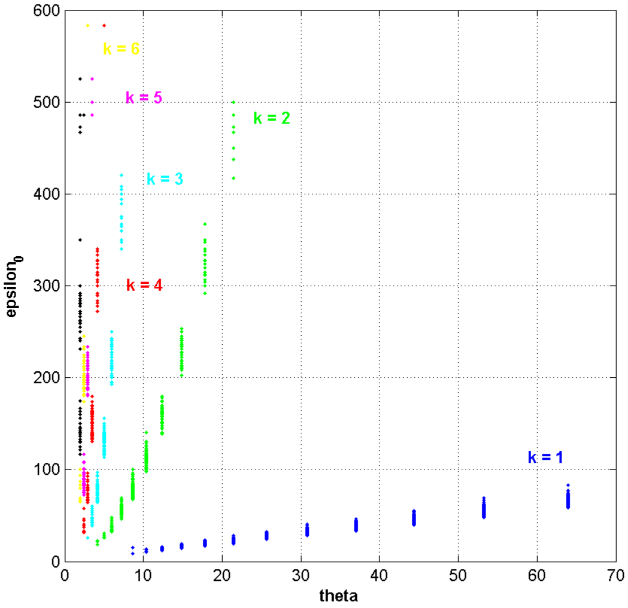

FIGURE 3. Test case 1: For several N, values of ε_0 and θ leading to the optimal solution. Integer K corresponds to the exponent in (2.10). Points in blue ($K = 1$) correspond to $\theta \leq N < \theta^2$, points in green ($K = 2$) correspond to $\theta^2 \leq N < \theta^3$.

3. Numerical experiments

Thanks to the forthcoming numerical experiments, we aim to study the influence of the following parameters, that have been introduced in Section 2:

- ε which leads to the optimality condition (2.8). One can easily check that $\varepsilon < 1/N$ is sufficient to obtain an optimal assignment in the case of integer distances d_{ij}, but we need to investigate the case of real distances.
- θ which is heuristic: is there a domain in $\mathbb{R}^2$ of couples (θ, ε_0) for which the transport cost is minimum, leading to the solution of the assignment problem? Several criteria will be presented here through some experiments on integer and real data.

Other properties that have to be considered are the complexity of the algorithm and its CPU cost (e.g. its computational cost on a workstation). For such an algorithm, which is strongly dependent on data configuration, we talk about the *worse-case complexity*: an upper bound for the complexity, obtained for bad-conditioned data. There is no theoretical study available so far, but it was evidenced (see e.g. [4,7]), that the worse-case complexity of the *Auction Algorithm* with ε-scaling and integer data is $\mathcal{O}(N^3 \log(NC))$, where

$$C = \max_{i,j=1,\dots,N} d_{ij}.$$

Hereafter, we present the complexity study of our specific configuration, and then conclude on the efficiency and possible improvements of this algorithm.

3.1. Validation of heuristic parameters

3.1.1. *The case of integer distances*

The configurations of initial and final positions, called Test case 1, and shown in Figure 2, are the following: the initial particles are located at points $(i,j,k) \in \mathbb{N}^3$ with $1 \leq i,j,k \leq 10$, and the final one are translated by the vector $(10,0,0)$. The assignment with minimal transport cost is then $\{((i,j,k),(i+10,j,k))\}_{i,j,k=1,2,\dots,10}$.
In the configuration Test Case 1, we look for the couples (ε_0, θ) leading to the optimal assignment. The results are given in Figure 3. This is done by running the *Auction Algorithm* on several number of particles N, and several couples (ε_0, θ). If the numerical solution is exactly the optimal one, then a point is added in Figure 3.
These points are classified with respect to colors, representing several values of K. The same N may appear several times, for instance $N = 64^2$ can also be written $N = 8^4$. Several parameters may then possibly lead to the optimal solution, and the aim is to find an interval in which we are sure the optimal solution is always found.

As discussed in [5], a sufficient condition to reach the optimal assignment in the case of integer distances $d_{ij} = |X_i - Q_j|^2$ is $\varepsilon_0 < 1$ (see Remark 2.4 and Equation (2.10)). This condition is not necessary, as can be seen in Figure 3.

According to [5], the choice of θ is heuristic (for integer distances as well as for real distances): $4 \leq \theta \leq 10$. For large N (e.g. for large K), the points are located in the interval $0 < \theta \leq 10$, which means that for biggest θ, the algorithm did not find the optimal solution. In our tests, we thus found that the interval $4 \leq \theta \leq 10$ is not really well adapted: $4 \leq \theta \leq 6$ seems to be a more confident interval.

3.1.2. *The case of real distances*

The ε-scaling version of the *Auction Algorithm* aims only at accelerating the convergence of the algorithm, and does not change the optimality criterion of the solution. That is why we only deal in this part with the initial auction process for fixed ε (see Section 2.2 above).

This behavior is experimented in the configuration Test case 2, presented in Figure 4. The goal of this part is to find a link between N and the largest $\varepsilon < 1$ leading to the smallest transport cost. The interval $4 \leq \theta \leq 6$ previously deduced from the integer distances is also validated here, and in the following we take $\theta = 4$.

Since we want to understand precisely the behavior of the algorithm in function of the disorder of the configuration, we only change the initial positions from Test case 1, taking initial random positions.

We first define an **approximated optimal solution**, using the following algorithm:

(1) Choose a value for N.
(2) Consider a finite set of values of ε, compute the ε-optimal solution for each of them; we obtain a set of assignments S_ε indexed by ε.
(3) For each assignment S_ε, compute the associated transport cost $D_\varepsilon = \sum_{i=1}^{N} d_{ij(i)}$.
(4) Look for the minimal cost $\widetilde{D}^* = \min_{\varepsilon} D_\varepsilon$. This minimal cost is such that $\widetilde{D}^* \geq D^*$.

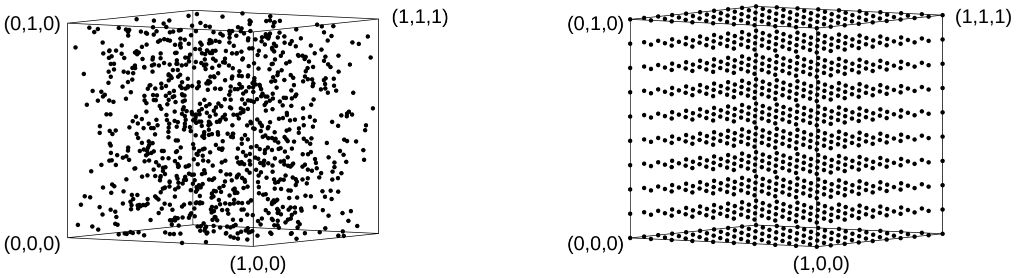

Figure 4. Test case 2. Input set $(X_i)_{1\leq i\leq N}$ is randomly generated, whereas the output set $(Q_j)_{1\leq j\leq N}$ is uniform, for 1000 particles. This configuration is similar to the one met in the SDM calculations.

(5) Compare each D_ε to $\widetilde{D}^*$ and keep the values of ε that realize $D_\varepsilon = \widetilde{D}^*$. We denote by $\widetilde{\varepsilon} = \widetilde{\varepsilon}(N)$ the largest one.

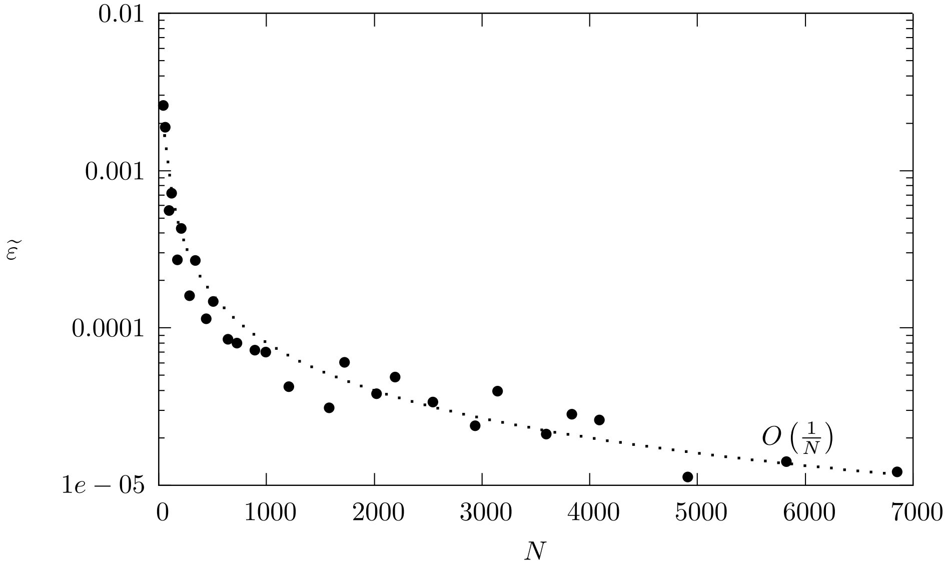

Figure 5. Number of particles N versus the maximum $\widetilde{\varepsilon}$ leading to the smaller cost.

Figure 5 describes the behavior of $\widetilde{\varepsilon}$, for a special configuration of the data: the final positions are contained in a cubic or almost cubic configuration (like Figure 4). Each value of $\widetilde{\varepsilon}$ of Figure 5 is a mean of ten realizations for a given N. For a given N, most of the $\widetilde{\varepsilon}$ are equal, and the confident interval is $\mathcal{O}(\widetilde{\varepsilon})$. In this case, $\widetilde{\varepsilon}$ seems to follow a law like $\widetilde{\varepsilon} = \mathcal{O}(\frac{1}{N})$, which is the same behavior found in the integer case (see Remark 2.4).

Compared to figure 3, several remarks have to be made: first, in the case of integer distances, the optimal configuration is found even for high ε_0, because of the simplicity of our test case. Second, the configuration of Test Case 2, which contains more disorder, implies lower values for $\widetilde{\varepsilon}$. Finally, it is not sure at all that for all $\varepsilon_0' < \varepsilon_0$, the algorithm reaches the optimal solution. In figure 5, $\widetilde{\varepsilon}$ is the largest one found, ensuring that for all ε taken lower than it, the optimal configuration is found.

3.2. Complexity and CPU time

The worse-case complexity for the *Auction Algorithm* (with ε-scaling, in the case of integer data) is said to be in $\mathcal{O}(N^3 \log(NC))$ [4,7]. Surprisingly, Figure 6 shows a better result for Test case 2: the complexity in our case (according to $\widetilde{\varepsilon}$, for a fixed θ), tends to be ruled by a law in $\mathcal{O}(N^2)$.

It is noteworthy that, in our application, the configuration and the regularity of the final positions of Test case 2 (shown in Figure 4) may improve the prefactor appearing in front of $\mathcal{O}(N^2)$. Indeed, taking a random repartition of the final positions rather than the regular one in Test Case 2 leads to the same behavior of the complexity, except that the prefactor is higher.

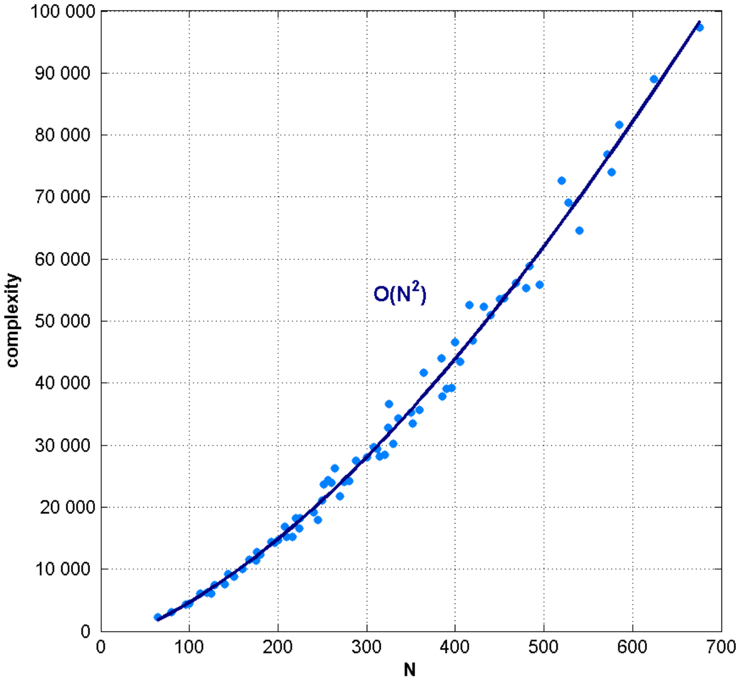

Figure 6. Complexity of the algorithm in the configuration of Figure 4.

The implementation of the *Auction Algorithm* uses efficient structures, but the time of execution in a sequential processor remains high: 3 *min* for 10^4 particles. The main limitation appears in the *Step 2.* of Algorithm 2.5: for large N, and a given number $N - N_k$ of particles to assign, the number of unassigned particles is not strictly decreasing, but can remain constant during numerous iterations of the loop 2.1. The reasons of such a phenomena are not well understood now, and we expect that some modifications (introduction of virtual source and well, for instance) may improve the *Auction Algorithm*.

Conclusion and forecast

After introducing the framework of the *Stochastic Downscaling Method*, we have presented an algorithm that corrects the particle positions. This algorithm, introduced by D.P. Bertsekas, is based on an auction process, and on an ε-scaling refinement; it contains several parameters. Our aim was to understand this algorithm, and use it as a reference method for the optimal transport problem.

We have proposed a way to characterize the optimality of the solution in the space of parameters, and a criterion on parameters ε_0 and θ which ensures to find a quasi-optimal solution in the case of real distances: $4 \leq \theta \leq 6$, and $\varepsilon = \mathcal{O}(\frac{1}{N})$. It is noteworthy that, even if the algorithm does not necessarily provide the optimal solution, Equation (2.8) provides a control of the optimality error with the parameter ε.

There is no theoretical result available on the complexity of the algorithm, since its efficiency is strongly dependent on the data. However, in our case, the complexity is surprisingly low: $\mathcal{O}(N^2)$. Although the complexity behaves rather goodly, some work has to be done to bring down the CPU time, using improvements of the *Auction Algorithm* proposed in [4].

As a conclusion, this work validates the *Auction Algorithm* as a powerful tool for optimal transport problem in dimension three. This algorithm will be the reference method, that will be compared to other cheaper strategies solving the uniform density constraint, as the one proposed in [19]. Our future work thus consists in coupling *SDM* with the *Auction Algorithm*, and in adapting the *Auction Algorithm* to our application. Indeed, the resolution of the stochastic Lagrangian model is based on a *Particle in Cell* method, and this specific structure of cells should be integrated in the *Auction Algorithm*. This could lead to a cheaper solution in term of number of operations, and a more adapted solution, according to our physical requirements.

Acknowledgments

The authors are gratefull to A. Sobolevskiĭ for providing precious information and programs. They are also greatly indebted to the anonymous reviewers for their numerous helpful suggestions. This work was partially supported by ADEME, CEMRACS, CNRS and INRIA (formation par la recherche).

References

[1] M.L. Balinski. A competitive (dual) simplex method for the assignment problem. *Math. Programming*, 34(2):125–141, 1986.

[2] J.D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numer. Math.*, 84(3):375–393, 2000.

[3] F. Bernardin, M. Bossy, C. Chauvin, A. Rousseau, P. Drobinski, and T. Salameh. Stochastic Downscaling Method: Application to wind refinment. *Stochastic Environmental Research and Risk Assessment*, 2008. To appear.

[4] D.P. Bertsekas. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications, Springer, Netherlands*, 1(1):1–66, 1992.

[5] D.P. Bertsekas. Network Optimization: Continuous and Discrete Models. *Athena Scientific, Belmont, Massachusetts*, 1998.

[6] D.P. Bertsekas. Auction Algorithms. *Encyclopedia of Optimization (C.A. Floudas and P.M. Pardalos, Eds), Kluwer Academic Publishers, Vol. 1, pp. 73-77*, 2001.

[7] D.P. Bertsekas and J. Eckstein. Dual Coordinate Step Methods for Linear Network Flow Problems. *Mathematical Programming, Series B*, 42:203–243, 1988.

[8] M. Bossy and J. Fontbona. Private communication.

[9] Yann Brenier. Optimal transportation theory via geometric partial differential equations. In *Proceedings of the 5th International Congress on Industrial and Applied Mathematics, Chap. III (Sydney, 2003)*. SIAM, 2003.

[10] S.K. Das and P.A. Durbin. A lagrangian stochastic model for dispersion in stratified turbulence. *Physics of Fluids*, 17(2):025109, 2005.

[11] Thomas D. Dreeben and Stephen B. Pope. Probability density function/Monte Carlo simulation of near-wall turbulent flows. *J. Fluid Mech.*, 357:141–166, 1998.

[12] J.-F. Jabir. Modèles stochastiques lagrangiens de type Mckean-Vlasov conditionnel et leur confinement. *PhD thesis*, 2008.

[13] Robert J. McCann. Existence and uniqueness of monotone measure-preserving maps. *Duke Math. J.*, 80(2):309–323, 1995.

[14] B. Mohammadi and O. Pironneau. *Analysis of the k-epsilon turbulence model.* Masson, Paris, 1994.

[15] S.B. Pope. On the relationship between stochastic Lagrangian models of turbulence and second-moment closures. *Phyics of Fluids*, 6:973–985, 1993.

[16] S.B. Pope. Lagrangian pdf methods for turbulent flows. *Annu. Rev. Fluid Mech.*, 26:23–63, 1994.

[17] S.B. Pope. *Turbulent flows.* Cambridge Univ. Press, 2003.

[18] P.-A. Raviart. An analysis of particle methods. In *Numerical methods in fluid dynamics (Como, 1983)*, volume 1127 of *Lecture Notes in Math.*, pages 243–324. Springer, Berlin, 1985.

[19] A. Rousseau, F. Bernardin, M. Bossy, T. Salameh, and P. Drobinski. Stochastic particle method applied to local wind simulation. *IEEE*, 2007.

[20] T. Salameh, P. Drobinski, L. Menut, B. Bessagnet, C. Flamant, A. Hodzic, and R. Vautard. Aerosol distribution over the western mediterranean basin during a tramontane/mistral event. *Ann. Geophys.*, 11:2271–2291, 2007.