Wind Simulation Refinement: Some New Challenges for Particle Methods

C. Chauvin[1], F. Bernardin[2], M. Bossy[1], and A. Rousseau[3]

[1] INRIA, Sophia Antipolis, France
Claire.Chauvin@inria.fr, Mireille.Bossy@inria.fr
[2] CETE de Lyon, LRPC, Clermont-Ferrand, France
Frederic.Bernardin@developpement-durable.gouv.fr
[3] INRIA, Laboratoire Jean Kuntzmann, Grenoble, France
Antoine.Rousseau@inria.fr

Summary. We present two new challenges related to the stochastic downscaling method (SDM) that we applied to wind simulation refinement in Bernardin et al. (Stoch. Environ. Res. Risk Assess. 23:851–859, 2009). After setting the framework, we introduce the boundary forcing issue, and propose a numerical scheme adapted to *Particle in Cell* methods. Then we turn to the uniform density constraint raised by SDM and propose some new methods that rely on optimization algorithms.

1 The Stochastic Downscaling Method

We are interested in the behaviour of an *incompressible fluid* in a domain $\mathcal{D}$ of $\mathbb{R}^3$; $\mathcal{D}$ is such that the mass density ρ is supposed constant. We decompose the unknown functions as the sum of a large-scale component and a turbulent one. Rather than solving the Reynolds Averaged Navier Stokes (RANS) equations on the mean velocity $\langle U \rangle$ and pressure $\langle \mathcal{P} \rangle$, we consider some stochastic differential equations (SDEs) that describe the stochastic dynamics of a fluid particle with state variables $(X_t,\ \mathcal{U}_t)_{t \geqslant 0}$:

$$dX_t = \mathcal{U}_t dt, \tag{1a}$$

$$\begin{aligned} d\mathcal{U}_t = &- \frac{1}{\rho} \nabla_x \langle \mathcal{P} \rangle (t, X_t) dt - \left(\frac{1}{2} + \frac{3}{4} C_0 \right) \langle \omega \rangle (t, X_t) \left(\mathcal{U}_t - \langle U \rangle (t, X_t) \right) dt \\ &+ \sqrt{C_0 \varepsilon (t, X_t)} dW_t \\ &- \sum_{0 \leq s \leq t} 2\, \mathcal{U}_{s-}\, \mathbb{1}_{\{X_s \in \partial \mathcal{D}\}} + \sum_{0 \leq s \leq t} 2\, V_{\text{ext}}(s, X_s)\, \mathbb{1}_{\{X_s \in \partial \mathcal{D}\}}, \end{aligned} \tag{1b}$$

where ε is the turbulent kinetic energy dissipation rate, $\langle \omega \rangle$ the turbulent frequency, and $(W_t)_{t \geq 0}$ is a three dimensional Brownian motion. The foundation

A.D. Fitt et al. (eds.), *Progress in Industrial Mathematics at ECMI 2008*,
Mathematics in Industry 15, DOI 10.1007/978-3-642-12110-4_123,

of such a model can be found in [1] and was inspired from [7]. The last two terms of (1b) model a Dirichlet condition (see [3]):

$$\langle U\rangle(t,x) = V_{\text{ext}}(t,x),\ \ x \in \partial\mathcal{D}, \tag{2}$$

V_{ext} denoting a known external velocity field (provided e.g. by measures, large scale simulations, or statistics). In the general RANS equations, $\langle\mathcal{P}\rangle$ is recovered thanks to the following Poisson equation:

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

which requires the knowledge of the second order moments of the velocity; this can be done thanks to turbulent closures, see [5] for a review of these models.

Assume that there exists a Lagrangian density f_L, such that at every time t the measure $f_L(t;x,V)dxdV$ is the law of the random process $(X_t,\mathcal{U}_t)$ solution of (1); a fluid particle satisfying (1) and (3) also satisfies (at least formerly), for almost all $x \in \mathcal{D}$

$$\int_{\mathbb{R}^3} f_L(t;x,V)\ dV = \rho, \tag{4a}$$

$$\nabla_x \cdot \langle U\rangle(t,x) = 0. \tag{4b}$$

The method that we define, called the *Stochastic Downscaling Method (SDM)*, is of a totally new type: its consists in simulating a solution of (1), (2), (4) with a given V_{ext}.

2 Numerical Description of SDM

2.1 The Stochastic Particle Method

The time is discretized with a sequence $t_k = k\Delta t, k = 0,\dots,K,\ \Delta t = T/K$. The stochastic dynamics is approximated at time t_k by the discrete random variables $(X_k^n, \mathcal{U}_k^n, 1 \le n \le N)$ associated to N fluid particles dropped inside $\mathcal{D}$. The statistics on these variables are defined using a local approximation, as in the *Particle in Cell* method (see [8]). More precisely, in the *Nearest Grid Point* method, a partition of $\mathcal{D}$ into N_c cells is defined: $\mathcal{D} = \cup_{i=1}^{N_c}\mathcal{C}_i$, associating N_i particles to each cell $\mathcal{C}_i$. A statistics $Q(U)$ is defined on each cell $\mathcal{C}_i$ by

$$\langle Q(U)\rangle_k(x) = \frac{1}{N_i}\sum_{n=1}^{N_i} Q\left(\mathcal{U}_k^n\right), \qquad x \in \mathcal{C}_i. \tag{5}$$

Notice that the method we construct is not hybrid. In particular, inside $\mathcal{D}$ the pressure gradient term $-\frac{1}{\rho}\nabla_x\langle\mathcal{P}\rangle$ is not computed by mean of a PDE solver.

Moreover, the computation of the right-hand-side of (3) is far too costly since it requires a very fine cell subdivision. Instead, we proceed to a fractional step algorithm inspired from Pope (see [7]): at each step, we solve (1b) without the term $-\frac{1}{\rho}\nabla_x\langle\mathcal{P}\rangle$. We simulate the pressure effect by solving the constraints (4), more adapted to a particle method [4].

2.2 Two New Numerical Challenges

In this paper, we focus our work on two issues: first, the confinement scheme required by (2). To the best of our knowledge, the case of (inhomogeneous) imposed boundary conditions in the framework of stochastic particle methods has not been formerly studied in the literature. Second, we focus on the transportation problem raised by (4a) (see [4] for some first studies in the SDM context).

Solving the Boundary Condition (2)

The external velocity $V_{\rm ext}$ is imposed at the boundaries of $\mathcal{D}$. The guidance is modelled by the two last terms of (1b). For robustness considerations (see [6]), we introduce the exponential version of the explicit Euler scheme. Hereafter, we sketch the main steps of the algorithm. After a prediction step, the dynamics of the outgoing particles is treated by the following reflection scheme:

At time t_k, for each particle n:

1. *Prediction.* Predict the position $\widetilde{X}_k^n = X_{k-1}^n + \Delta t\ \mathcal{U}_{k-1}^n$ and the velocity $\widetilde{\mathcal{U}}_k^n$ using an exponential scheme [10]:
$$d\mathcal{U}_t^n = -\left(\frac{1}{2}+\frac{3}{4}C_0\right)\langle\omega\rangle_{k-1}\left(\mathcal{U}_t^n - \langle U\rangle_{k-1}\right)dt + \sqrt{C_0\varepsilon_{k-1}}dW_t, \quad t\in[t_{k-1},t_k], \tag{6}$$
where $\langle U\rangle_{k-1}$, $\langle\omega\rangle_{k-1}$ and ε_{k-1} are evaluated in the cell containing X_{k-1}^n. If $\widetilde{X}_k^n\in\mathcal{D}$, then set $X_k^n = \widetilde{X}_k^n$ and $\mathcal{U}_k^n = \widetilde{\mathcal{U}}_k^n$.
2. *Reflection.* When $\widetilde{X}_k^n \notin \mathcal{D}$; let $\delta_{out} = \lambda\Delta t$ be the boundary hitting time, and $x_{out} = X_{k-1}^n + \delta_{out}\mathcal{U}_{k-1}^n$ be the hitting position, then the reflected position reads
$$X_k^n = x_{out} + (\Delta t - \delta_{out})\left(2V_{\rm ext}(t_{k-1},x_{out}) - \mathcal{U}_{k-1}^n\right). \tag{7}$$

   The reflected velocity is constructed by two successive steps. First, we simulate Equation (6) between t_{k-1} and t_{out^-} with an exponential scheme to obtain the velocity $\mathcal{U}_{t_{out^-}}$. Then, in order to match the boundary conditions, a *jump* is imposed to the velocity at $t = t_{out}$, leading to $\mathcal{U}_{t_{out^+}} = 2V_{\rm ext}(t_{k-1},x_{out}) - \mathcal{U}_{t_{out^-}}$. The second advancement is done between t_{out^+} and t_k.

3. *Kill-Build Procedure.* It may happen that the reflected position (7) does not belong to $\mathcal{D}$. In this case, the particle is *killed, and created* in a boundary cell with incoming velocity V_{ext}.

The linear equation (6) is exactly solved in step 1; the same holds for the two velocity advancements in step 2, knowing the boundary hitting time t_{out}, and the velocity jump.

Solving the Constant Mass Density Constraint (4a)

We come now to the second difficulty of this paper. The condition (4a) implies that the number of particles per cell has to be constant: for each cell C_i, $N_i = N_{pc}$, and thus the total number of particles is $N = N_c N_{pc}$. After steps 1–3 above, this condition may not hold anymore. Let us denote x_i the particle locations at the end of step 3. When $N_i < N_{pc}$, locations are randomly created in $\mathcal{C}_i$, and the set $\{q_j\}_{1 \leq j \leq N}$ is constructed by taking N_{pc} particles per cell.

At this point, the constant mass density problem can be interpreted (at least formally) as an *optimal transport problem* (see [1, 4]): defining the cost $p_{ij} = \|x_i - q_j\|_{L_2}^2$ of transporting a particle from x_i to q_j, the problem consists of finding an element σ of the set of permutations $\mathcal{S}_N$ of $\{1, \dots, N\}$ which minimizes the overall transport cost:

$$(P) \quad \text{Find } \sigma^* \in \mathcal{S}_N \text{ such that } D^* := \sum_{i=1}^{N} p_{i\sigma^*(i)} = \min_{\sigma \in \mathcal{S}_N} \sum_{i=1}^{N} p_{i\sigma(i)}. \qquad (8)$$

This so-called *Assignment Problem* has been tackled by D. Bertsekas in [2], introducing the *Auction Algorithm*, where the optimality condition (8) is ε-relaxed:

$$D^* \leqslant \sum_{i=1}^{N} p_{i\sigma^*(i)} \leqslant D^* + N\varepsilon. \qquad (9)$$

The overall cost of the final assignment is within $N\varepsilon$ of being optimal. Numerical tests (see [4]) have shown that in our specific configuration, the optimal solution is obtained when $\varepsilon \backsimeq \frac{C}{N}$, with a complexity of order N^2. Such a computational cost involves a very slow execution of SDM, since we need a large number of particles N for the Monte Carlo method to converge.

Hereafter, in the SDM framework, we present our strategies to reduce the number of objects involved in the Auction Algorithm.

3 Benchmarks

In order to decrease the number of particles involved in the Auction Algorithm, we consider the supernumerary particles and possibly a set of particles coming from tanks, defined in each cell. Let be the sets $\mathbb{X}$, containing the particles to be transported, and $\mathbb{Q}$, the final locations, constructed as follows:

Table 1. Comparison of several transportation algorithms: Auction Algorithm with Tank (AAT), for several tank sizes, and Triangular Transport (TT)

		Run time (s)	D	c_{max}	c_{move}
AAT	$\alpha = 1$	467,183	8.8	1.5	4,846
AAT	$\alpha = 0.01$	1,646	66	1.7	3,012
AAT	$\alpha = 0$	557	85	2.3	2,414
TT		0.38	110	1.4	9,843

Initialization: $\mathbb{X} = \mathbb{Q} = \varnothing$*, and the tank size* $N_{tank} = \alpha N_{pc} \in \mathbb{N}$*,* $0 \leq \alpha \leq 1$*.*
For all $\mathcal{C}_i$*:*

If $N_i > N_{pc}$*: add* $N_i - N_{pc}$ *particles to* $\mathbb{X}$*, and add* N_{tank} *other particles of* $\mathcal{C}_i$ *to* $\mathbb{X}$ *and* $\mathbb{Q}$*.*

If $N_i < N_{pc}$*: create* $N_{pc} - N_i$ *particles in* $\mathcal{C}_i$*, and add them to* $\mathbb{Q}$*. If* $N_{pc} - N_i < N_{tank}$ *then add* $N_{tank} - (N_{pc} - N_i)$ *other particles to* $\mathbb{X}$ *and* $\mathbb{Q}$*.*

If $N_i = N_{pc}$ *then add* N_{tank} *particles to* $\mathbb{X}$ *and* $\mathbb{Q}$*.*

The Auction Algorithm with Tank (AAT) is applied to $(\mathbb{X}, \mathbb{Q})$, and by then the particles of $\mathbb{X}$ are assigned to the final locations of $\mathbb{Q}$, leading to the global transport cost $D = \sum_{i=1}^{|\mathbb{X}|} p_{i\sigma^*(i)}$.

In a previous work [9], the triangular transport procedure (TT) was presented as a competitive method for the uniformization of the mass density: in the case of dimension one, the transport cost is known to be optimal, with a very cheap complexity of $\mathcal{O}(n \log n)$.

Table 1 compares the AAT procedure with several tank sizes to the TT procedure. The initial locations $\{x_i\}_{1 \leq i \leq N}$ are randomly created inside $\mathcal{D}$, and $\mathcal{D}$ is partitioned into $6 \times 6 \times 6$ cells, with $N_{pc} = 800$ particles per cell. The four columns correspond to the mean of the following quantities: the computational time on a work station (run time (s)), the transport cost D, the largest number c_{max} of cells crossed by the particles during their transportation (expected to be close to 1), and finally the number c_{move} of particles which have leaved their initial cell. The variable c_{max} plays a crucial role in SDM: particles transport physical information, and hence we look for an optimization procedure that preserves the physics inside each cell.

When $N_{tank} = N_{pc}$ (AAT with $\alpha = 1$, full tank), the Auction Algorithm is applied to the N particles: an optimality condition can be written (see (9)). This test is taken as a reference in terms of transport cost D and c_{max}. Nevertheless the computational time is far too large, and unsuitable for SDM since the mass density uniformization has to be done at *every* time step. Setting $\alpha = 0$ in AAT (empty tank) consists in transporting the supernumerary particles towards the cells that miss particles. With 1% particles in the tank, we obtain a satisfying trade-off between computational and transport costs (see Table 1). Although this procedure does not lead to an optimal transport cost, the number c_{max} of crossed cells is surprisingly small; this is precisely what matters in our application. The introduction of a better-adapted metric to define $p_{i\sigma^*(i)}$ for our application is still an open problem. Meanwhile, our preferred method remains TT as it both minimizes the computational cost and the number of crossed cells.

4 Conclusion

We have introduced a new numerical scheme which ensures that the Dirichlet condition (2) is satisfied. Then, we have presented a new adaptation of the Auction Algorithm, that improves the resolution of the optimal transport problem in the context of SDM: the computational cost is reduced, involving few particles in the process, with a satisfying transport cost.

References

1. Bernardin, F., Bossy, M., Chauvin, C., Drobinski, P., Rousseau, A., Salameh, T.: Stochastic downscaling method: application to wind refinment. Stoch. Environ. Res. Risk Assess. **23**(6) 851–859 (2009)
2. Bertsekas, D.P.: Auction algorithms. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, vol. 1, pp. 73–77. Kluwer, London (2001)
3. Bossy, M., Jabir, J.-F.: Confined Langevin processes and mean no-permeability condition. Preprint (2008)
4. Chauvin, C., Hirstoaga, S., Kabelikova, P., Rousseau, A., Bernardin, F., Bossy, M.: Solving the uniform density constraint in a downscaling stochastic model. ESAIM Proc. **24**, 97–110 (2007)
5. Mohammadi, B., Pironneau, O.: Analysis of the k-Epsilon Turbulence Model. Masson, Paris, (1994)
6. Mora, C.M.: Weak exponential schemes for stochastic differential equations with additive noise. IMA J. Numer. Anal. **25**(3), 486–506 (2005)
7. Pope, S.B.: P.D.F. methods for turbulent reactive flows. Prog. Energy Comb. Sci. **11**, 119–192 (1985)
8. Raviart, P.-A.: An analysis of particle methods. In: Numerical Methods in Fluid Dynamics (Como, 1983), vol. 1127, pp. 243–324. Springer, Berlin (1985)
9. Rousseau, A., Bernardin, F., Bossy, M., Drobinski, P., Salameh, T.: Stochastic particle method applied to local wind simulation. In: IEEE International Conference on Clean Electrical Power, pp. 526–528. Springer, Berlin (2007)
10. Talay, D.: Probabilistic numerical methods for partial differential equations: elements of analysis. In: Talay, D., Tubaro, L. (eds.) Probabilistic Models for Nonlinear Partial Differential Equations, vol. 1627, pp. 148–196. Springer, Berlin (1996)