

Stochastic particle method applied to local wind simulation

Antoine Rousseau, Frédéric Bernardin, Mireille Bossy

FirstName.LastName@inria.fr

INRIA, OMEGA Project, Sophia Antipolis, France

Tamara Salameh, Philippe Drobinski

LMD, Ecole Polytechnique, Palaiseau, France.

Abstract

A Lagrangian stochastic model is introduced in order to describe the local behavior of the wind. Based on some MM5 data, and thanks to particles driven by stochastic differential equations, we propose a numerical method allowing to improve the MM5 simulations at small scales, without requiring too much additional computational cost.

1. Introduction

The aim of this work is to propose a numerical method to improve the wind forecasting at small scales. Up to now, solvers like *MM5* (developed at NCAR and PSU, see [3] and [4]) have a computational limit in both horizontal and vertical ranges. Namely, for a computation on the whole Mediterranean basin, it is quite difficult, if not hopeless, to run *MM5* with an horizontal resolution finer than $1km$. Thanks to some numerical tools based on particle methods, we wish to improve significantly this resolution. We recall that the computations that we consider are local and concern a small part of the *MM5* computational domain. For this reason and since particle methods do not require any stability (CFL) condition, the CPU time should remain sufficiently small (compared to what *MM5* requires).

2. Numerical algorithm and related issues

The numerical tool that we consider is based on the usual *MM5* computations. Namely, *MM5* will guide our simulations, since it both provides initial data and boundary conditions. This is a *one way* coupling so far, but we think of plugging the local information back into *MM5* in future works.

Consider a 3D box (for example a cell of *MM5* mesh) where the three components (u, v, w) of the velocity field are given on each side. We use a Lagrangian approach, and adopt the point of view of a generic fluid particle, to get some information at smaller scales (inside the box). To this aim, we run a *Particle In Cell* (PIC) numerical algorithm, which consists in splitting our domain into cells C_i whose dimensions are Δx , Δy and Δz , in which we drop K particles having Langevin dynamics (see Fig. 1). Then, we compute some local characteristics of the fluid by averaging the corresponding particle attributes in each cell.

The Lagrangian stochastic model that we study is borrowed from S.B. Pope [6] and describes the behavior of our particles in the considered domain $\mathcal{D} \subset \mathbb{R}^3$. We consider a generic fluid particle, and denote by $(X_t, \mathcal{U}_t, \omega_t)$ its position, velocity and turbulence frequency. The model consists in writing a stochastic differential equation (SDE) that describes the dynamic of the pro-

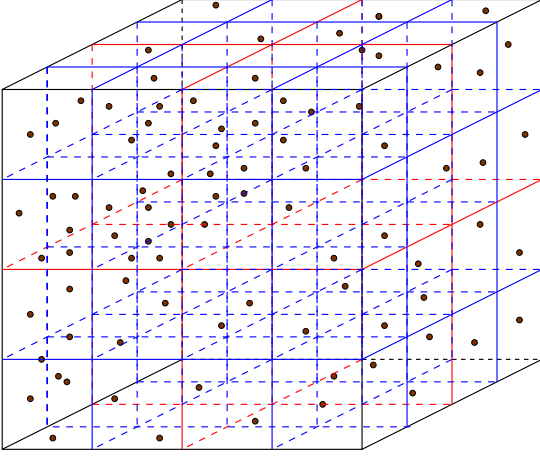


FIG. 1. The wind computational domain \mathcal{D} , divided in cells C_i , in which fluid particles are dropped.

cess $(X_t, \mathcal{U}_t, \omega_t)_{t \geq 0}$:

$$dX_t = \mathcal{U}_t dt, \quad (2.1a)$$

$$d\mathcal{U}_t = -\frac{1}{\rho} \nabla \langle \mathcal{P} \rangle(t, X_t) dt + D_{\mathcal{U}} dt + B_{\mathcal{U}} dW_t, \quad (2.1b)$$

$$d\omega_t = D_{\omega} dt + B_{\omega} dW'_t, \quad (2.1c)$$

where (W_t) and (W'_t) are respectively 3D and 1D independent Brownian motions.

The drift coefficients $D_{\mathcal{U}}$ and D_{ω} depend on some mean-reverting terms of the following form :

$$\begin{aligned} \mathcal{U}_t - \mathbb{E}(\mathcal{U}_t | X_t = x)|_{x=X_t} &\equiv \mathcal{U}_t - \langle \mathcal{U} \rangle(t, X_t) \\ \omega_t - \mathbb{E}(\omega_t | X_t = x)|_{x=X_t} &\equiv \omega_t - \langle \omega \rangle(t, X_t), \end{aligned}$$

where \mathbb{E} denotes the mathematical expectation under the underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and $\mathbb{E}(\mathcal{U}_t | X_t = x)$ is the conditional expectation of the velocity \mathcal{U}_t at the position $X_t = x$. In this model, the mean velocity field $\mathbb{E}(\mathcal{U}_t | X_t = x)$ is assimilated to the Reynolds averaged Eulerian velocity $\langle \mathcal{U} \rangle(t, x)$. Idem for the turbulent frequency ω_t . These conditional expectations are computed thanks to Monte-Carlo methods, and thus K must be chosen sufficiently large to insure the convergence of the MC approximations in each cell C_i :

$$\langle \mathcal{U} \rangle_i = \sum_{k=1}^K \mathbb{1}_{X_k \in C_i} \mathcal{U}_k \quad (2.2)$$

The diffusion coefficients $B_{\mathcal{U}}$ and B_{ω} depend on $\langle \omega \rangle(t, X_t)$ and the two first stochastic moments of the velocity. In particular, the coefficient B_{ω} is such that the stochastic process ω_t is nonnegative (see [2] for more details).

We supplement the SDEs (2.1) with the following continuity equation :

$$\frac{\partial \rho}{\partial t} + \text{div} \langle \rho \mathcal{U} \rangle = 0, \quad (2.3)$$

which is insured thanks to the two following constraints :

$$\rho(t) = \text{cst} = \rho_0, \quad (2.4a)$$

$$\text{div} \langle \mathcal{U} \rangle = 0. \quad (2.4b)$$

Finally, we recall that the particle dynamics in (2.1) must be constrained at the boundary of the computational domain : the mean velocities at the boundary are imposed by the MM5 values (MM5 guideline) :

$$\forall t \geq 0, \forall x \in \partial \mathcal{D}, \quad \langle \mathcal{U}_t \rangle(t, x) = V_g(t, x). \quad (2.5)$$

The system (2.1) describes the evolution of $(X_t, \mathcal{U}_t, \omega_t)$, which corresponds to the particle position, its velocity, and its turbulence frequency. These quantities are of Lagrangian type since the information is attached to the particle. On the contrary, the Eulerian quantities, that appear in the equations above, mainly $\langle \mathcal{U} \rangle(t, x)$ and $\langle \omega_t \rangle(t, x)$, correspond to averaged quantities. They are computed on each cell C_i by averaging the corresponding values of the particles that belong to C_i .

$$\langle \mathcal{U} \rangle(t, x) = \langle \mathcal{U} \rangle_i, \quad \forall x \in C_i.$$

The main steps of our numerical algorithm are the following (from time $t_n = n\Delta t$ to time $t_{n+1} = t_n + \Delta t$). Given the values of the processes (X^n, U^n, ω^n) at time $t = t_n$, we :

1. compute new values of velocity and turbulence frequency. To this aim, we compute $(\tilde{U}^{n+1}, \omega^{n+1})$ thanks to equation (2.1c) and

$$d\mathcal{U}_t = D_{\mathcal{U}} dt + B_{\mathcal{U}} dW_t. \quad (2.6)$$

Compared to original equation (2.1b), the latter equation does not take the pressure gradient into account, but this is done in the correction step for the velocities (see item 4 below).

The simulation of such stochastic differential equations inside \mathcal{D} is rather classical and is done thanks to a Euler scheme (e.g. [7]). A particular attention must be taken when the particles reach the boundary, in order to take into account the constraint (2.5).

2. move particles, that is compute \tilde{X}^{n+1} in order to verify equations (2.1a). Some particles can go out the domain according to outward MM5 velocity at the boundary. In that case, introduction of new particles at inward boundary must be performed.
3. modify the positions from \tilde{X}^{n+1} to X^{n+1} , so that (2.4a) is satisfied. This step is not classical at all and concerns the particle positions. The aim is to correct the particle positions in the domain such that :
 - their density is uniform (i.e. the number of particles is the same in each cell).
 - the “transport cost” is minimum.

This is a problem of discrete optimal transportation, which is known to be nonlinear and numerically very difficult in dimension 3 (see [1], [8], [5]), whereas the one-dimensional corresponding problem is linear, and easy to handle from the numerical viewpoint.

We thus introduce a method based on the simple case of dimension 1, for which the optimal transportation simply consists in sorting the particles (see e.g. Brenier [1]). We split the domain \mathcal{D} in what we call particle-tubes (see Fig. 2 below), defined by :

$$\begin{aligned} \mathcal{T}_{j,k}^x &= \{(x, y, z) \in \mathcal{D}, \\ &\quad (j-1)\Delta y \leq y \leq j\Delta y, \\ &\quad (k-1)\Delta z \leq z \leq k\Delta z\}, \\ \mathcal{T}_{i,k}^y &= \{(x, y, z) \in \mathcal{D}, \\ &\quad (i-1)\Delta x \leq x \leq i\Delta x, \\ &\quad (k-1)\Delta z \leq z \leq k\Delta z\}, \\ \mathcal{T}_{i,j}^z &= \{(x, y, z) \in \mathcal{D}, \\ &\quad (i-1)\Delta x \leq x \leq i\Delta x, \\ &\quad (j-1)\Delta y \leq y \leq j\Delta y\}, \end{aligned}$$

for all $1 \leq i \leq N_c^x$, $1 \leq j \leq N_c^y$, $1 \leq k \leq N_c^z$, where N_c^x , N_c^y and N_c^z respectively denote the

number of cells in the directions x , y and z .

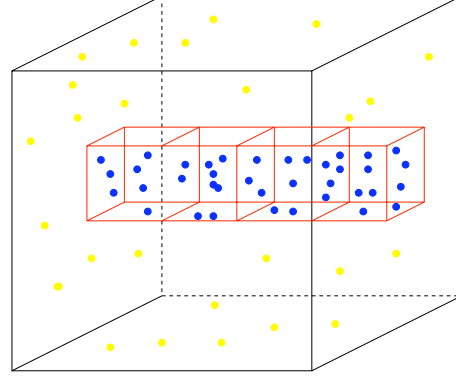


FIG. 2. Example of a particle-tube

In every tube, a 1D optimal transportation is performed, and one can prove that, after dealing with all the particle-tubes in each direction, one obtains a uniform number of particles in every cell. It has to be noticed that the final transportation is clearly not optimal, and improvements of the above method, called *Triangular Transportation Method*, are to be done in subsequent studies.

4. modify the velocities from $\tilde{U}^{(n+1)}$ to $U^{(n+1)}$ in order to verify equation (2.4b), hence (2.3). This correction is a standard issue in CFD : the aim is to obtain a divergence free velocity field. This is done thanks to the resolution of the following Poisson problem :

$$\Delta P = \frac{1}{\Delta t} \nabla \cdot \langle \tilde{U}^{(n+1)} \rangle. \quad (2.8)$$

We supplement this equation with Neumann boundary conditions, and solve it thanks to solvers based on fast Fourier transforms (FFT). The equation above only concerns Eulerian unknowns, and hence does not contribute significantly to the numerical cost of the algorithm.

We then compute $U^{(n+1)}$ thanks to

$$U^{(n+1)} = \tilde{U}^{(n+1)} - \Delta t \nabla P. \quad (2.9)$$

The velocity field is then such that $\text{div} \langle U^{(n+1)} \rangle = 0$.

3 Future works

In the coming months, we are planning to run *MM5* with two different resolutions. The first mesh will be coarse, and will be devoted to feed our stochastic model. The second mesh will be finer, and we hope that our simulations will provide comparable results, with a better (smaller) computational cost (CPU time).

Then, we will use observations from measurements campaigns, namely the campaign *FETCH* that took place in 1998 in Southern France, for the validation of our simulations. We are going to use surface wind measurements as well as wind measurements from radio soundings from the ground up to several kilometers.

4 Acknowledgments

This work was partially supported by the French Agency for the Environment and Energy Management (ADEME). The authors also want to thank Eric Peirano for very helpful discussions related to this article.

Références

- [1] 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.
- [2] T. Dreeben and S. Pope. Wall-function treatment in pdf methods for turbulent flows. *Phys. Fluids*, 9(9) :2692–2703, 1997.
- [3] G. Grell and J. Dudhia. A description of the fifth-generation penn state/ncar mesoscale model (mm5). *NCAR Technical Note*, NCAR/TN-398+STR :122, 1995.
- [4] Y. Guo and S. Chen. Terrain and land use for the fifth-generation penn state/ncar mesoscale modeling system (mm5). *NCAR Technical Note*, NCAR/TN-397+IA :114, 1994.
- [5] R. J. McCann. Existence and uniqueness of monotone measure-preserving maps. *Duke Math. J.*, 80(2) :309–323, 1995.
- [6] S. Pope. Lagrangian pdf methods for turbulent flows. *Annu. Rev. Fluid Mech.*, 26 :23–63, 1994.
- [7] D. Talay. Probabilistic numerical methods for partial differential equations : elements of analysis.

- In D. Talay and L. Tubaro, editors, *Probabilistic Models for Nonlinear Partial Differential Equations*, volume 1627 of *Lecture Notes in Mathematics*, pages 148–196. Springer-Verlag, 1996.
- [8] C. Villani. *Topics in optimal transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2003.