



HAL
open science

Formulation and analysis of a diffusion-velocity particle model for transport-dispersion equations

Paul Mycek, Grégory Pinon, Grégory Germain, Elie Rivoalen

► **To cite this version:**

Paul Mycek, Grégory Pinon, Grégory Germain, Elie Rivoalen. Formulation and analysis of a diffusion-velocity particle model for transport-dispersion equations. *Computational & Applied Mathematics*, 2016, 35 (2), pp.447-473. 10.1007/s40314-014-0200-5 . hal-01087854

HAL Id: hal-01087854

<https://hal.science/hal-01087854>

Submitted on 3 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Formulation and analysis of a diffusion-velocity particle model for transport-dispersion equations

Paul Mycek · Grégory Pinon · Grégory Germain · Elie Rivoalen

Accepted: October 29, 2014 – Available online: November 25, 2014

Abstract The modelling of diffusive terms in particle methods is a delicate matter and several models were proposed in the literature to take such terms into account. The Diffusion Velocity Method (DVM), originally designed for the diffusion of passive scalars, turns diffusive terms into convective ones by expressing them as a divergence involving a so-called diffusion velocity. In this paper, DVM is extended to the diffusion of vectorial quantities in the three-dimensional Navier-Stokes equations, in their incompressible, velocity-vorticity formulation. The integration of a Large Eddy Simulation (LES) turbulence model is investigated and a DVM general formulation is proposed. Either with or without LES, a novel expression of the diffusion velocity is derived, which makes it easier to approximate and which highlights the analogy with the original formulation for scalar transport. From this statement, DVM is then analysed in one dimension, both analytically and numerically on test cases to point out its good behaviour.

Keywords Diffusion Velocity Method · Particle method · Fourier Analysis · Transport-dispersion equations · Navier-Stokes equations

Paul Mycek (Corresponding author)
Duke University, Department of Mechanical Engineering and Materials Science, 144 Hudson Hall, Box 90300, Durham, NC 27708, USA
E-mail: paul.mycek@duke.edu
Tel.: +1 919-660-5348

Grégory Pinon · Elie Rivoalen
Laboratoire Ondes et Milieux Complexes, UMR 6294, CNRS–Université du Havre, 53, rue de Prony, BP 540, F-76058 Le Havre Cedex, France
E-mail: gregory.pinon@univ-lehavre.fr

Grégory Germain
IFREMER, Hydrodynamic & Metocean Service, 150, quai Gambetta, BP 699, F-62321 Boulogne-Sur-Mer, France
E-mail: gregory.germain@ifremer.fr

Elie Rivoalen
Laboratoire d'Optimisation et Fiabilité en Mécanique des Structures, EA 3828, INSA de Rouen, Avenue de l'Université, BP 08, F-76801 Saint-Etienne-du-Rouvray, France
E-mail: elie.rivoalen@insa-rouen.fr

1 Introduction

Particle methods, such as vortex methods or Smoothed Particle Hydrodynamics, generally consist of a Lagrangian transport, thanks to fluid dynamics equations, of quantitative information carried by particles. The transport equations linked to each particle (position, vortical intensity, passive scalar, etc.) are in fact partial differential equations. Owing to the Lagrangian aspect of the method, it is particularly well suited to flows where convective effects are dominant. However, transport equations may involve terms that do not represent advection. This is the case of the Navier-Stokes equations, which involve, for instance, a diffusion term.

The Diffusion Velocity Method (DVM) consists in writing the diffusive term, with either uniform or variable diffusion coefficient, as a convective term. It was first introduced by Fronteau and Combis (1984) and popularized by Mustieles (1990); Degond and Mustieles (1990), Ogami and Akamatsu (1991) and Kempka and Strickland (1993) in the early 1990s. This method was then largely analysed (Strickland et al 1996; Mas-Gallic 1999; Lacombe and Mas-Gallic 1999; Lacombe 1999; Lions and Mas-Gallic 2001), adapted to dispersion equations (Lions and Mas-Gallic 2001; Chertock and Levy 2001, 2002), coupled with turbulence models (Milane 2004) and extended to the diffusion of a vector field and to axisymmetric flows (Rivoalen et al 1997; Grant and Marshall 2005; Rivoalen and Huberson 1999).

DVM is particularly well suited to the resolution of diffusion or dispersion in open media. Indeed, there is an adaptation of the convected and diffused support, represented by the particles, in an extending domain. Applications concern for instance, pollutant dispersion in porous media (Beaudoin et al 2003), heat-vortex interaction (Ogami 1999), airfoil wake modelling (Guvernyuk and Dynnikova 2007), as well as the modelling of the diffusive term in the Navier-Stokes (Rivoalen et al 1997), heat (Dynnikov and Dynnikova 2011) or Lotka-Volterra (Gambino et al 2009) equations.

Originally, the DVM scheme was designed for transport equations of passive scalars. However, work has been initiated by Rivoalen et al (1997) to adapt this scheme to a three-dimensional Vortex method modelling of the Navier-Stokes equations, which involve the diffusion of a vector field, namely the vorticity field. On the other hand, for physical applications with turbulence, these equations usually need to be complemented by a turbulence model. Depending on the application and on implementation issues, different turbulence models may be chosen. Large Eddy Simulation (LES) is a model based on scale invariance and consists in considering that the flow quantities are filtered at a certain scale related to the problem discretisation. As a consequence, new terms appear in the modelled equations, which represent the influence of the small (unresolved) scales on the resolved scales (see, e.g., Sagaut (2006); Meneveau and Katz (2000)).

This paper intends to investigate a DVM formulation of the 3D Navier-Stokes equations with a LES model. In section 2, the scalar DVM method is briefly recalled and the vectorial formulation is then derived for the Direct Numerical Simulation (DNS) and the Large Eddy Simulation (LES) of the 3D Navier-Stokes equations. The main result is a novel expression of the diffusion velocity that eventually reduces to the scalar case. Section 3 is dedicated to the particle approximation of the diffusion velocity in the scalar and vectorial case, which emphasises the fact that both expressions are treated similarly. From this statement, the following sections

are dedicated to unidimensional analyses of DVM, on the academic problem of pure diffusion of a Gaussian patch. Section 4 then consists of theoretical considerations resulting from a Fourier analysis. Analytic expressions of the diffusion velocity error issuing from the continuous particle approximation are presented on one-dimensional theoretical test cases. In particular, the well-known pure diffusion of a Gaussian patch together with the use of Gaussian kernels allows us to benefit from convenient Gaussian product properties. This is due to the fact that the fundamental solution of Gaussian diffusion is a Gaussian itself. Finally, section 5 focuses on the consequences of the discretisation and on the influence of the overlapping parameter. The analytic results from the second part are compared to numerical computations using DVM. A global overview of the numerical behaviour of the method is presented. It must be stressed that, in order to study the purely Lagrangian behaviour of the particles as well as the orders of approximation, remeshing techniques are not considered in this paper.

2 Formulation of a Diffusion-Velocity model for the 3D Navier-Stokes equations

The original scalar Diffusion Velocity Method (DVM) has been largely described in the literature (see, e.g., Fronteau and Combis (1984); Mustieles (1990); Degond and Mustieles (1990); Ogami and Akamatsu (1991); Kempka and Strickland (1993)), and is briefly recalled here to highlight the analogy with the vectorial model described in the sequel.

The transport equation associated to a passive scalar $c(\mathbf{x}, t)$, which depends on the time t and the position $\mathbf{x}(t)$ can be written as follows:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = \nabla \cdot (\nu \nabla c). \quad (1)$$

The last term $\nabla \cdot (\nu \nabla c)$ basically reduces to $\nu \Delta c$ when ν is assumed to be constant. A transport equation for the velocity $\mathbf{u}(\mathbf{x}, t)$ can be added to the previous equation as well as boundary conditions. Depending on the studied media, the dispersion or diffusion coefficient $\nu(\mathbf{x}, t)$ can be represented either as a tensor in the case of a soluté transport at a macroscopic level (Beaudoin et al 2003) or as a scalar coefficient for the diffusion of a concentration $c(\mathbf{x}, t)$. The previous expression can be written in a purely convective form:

$$\frac{\partial c}{\partial t} + \nabla \cdot [(\mathbf{u} + \mathbf{u}_d)c] = 0, \quad \text{with} \quad \mathbf{u}_d(\mathbf{x}, t) = -\nu \frac{\nabla c}{c}, \quad (2)$$

The diffusion operator is, after a few mathematical manipulations, transformed into an algebraic convection operator. \mathbf{u}_d is generally called the diffusion velocity (Ogami and Akamatsu 1991).

Particle methods formalism will be used to model this equation. The physical domain \mathcal{S} is discretised into N particles ξ_i of support \mathcal{P}_i whose volume is σ_i , for i going from 1 to N . For each particle ξ_i , a position vector $\mathbf{X}_i(t)$ and a weight $C_i(t)$ are defined as follows:

$$\mathbf{X}_i(t) = \frac{\int_{\mathcal{P}_i} \mathbf{x}(t) d\mathbf{x}}{\int_{\mathcal{P}_i} d\mathbf{x}} \quad \text{and} \quad C_i(t) = \int_{\mathcal{P}_i} c(\mathbf{x}, t) d\mathbf{x} \approx c(\mathbf{X}_i, t)\sigma_i. \quad (3)$$

Equations (2) and (3) then lead to the following particle transport equations:

$$\begin{cases} \frac{d\mathbf{X}_i}{dt} = \tilde{\mathbf{u}}(\mathbf{X}_i, t) + \tilde{\mathbf{u}}_d(\mathbf{X}_i, t) \\ \frac{dC_i}{dt} = 0. \end{cases} \quad (4)$$

where $\tilde{\mathbf{u}}_d$ denotes the discrete particle approximation of \mathbf{u}_d , which will be detailed in section 3.1. Generally speaking, \tilde{q} will denote the discrete particle approximation of the (scalar or vectorial) quantity q , while $\langle q \rangle$ will denote its continuous approximation (see section 3).

The convective aspect of the previous equations is then preserved. The diffusive aspects are taken into account thanks to a modification in the particles trajectory $\mathbf{X}_i(t)$, as in some stochastic models (Chorin 1973) although here, the diffusion velocity calculation is deterministic (Lions and Mas-Gallic 2001).

Let us now investigate the modelling of the three-dimensional Navier-Stokes equations. For an incompressible fluid, their velocity-vorticity formulation reads:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla) \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nu \Delta \boldsymbol{\omega} \quad (5)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (6)$$

where $\boldsymbol{\omega} = \nabla \wedge \mathbf{u}$ and where $(\boldsymbol{\omega} \cdot \nabla) \mathbf{u}$ represents the stretching term. Note that in two dimensions, $\boldsymbol{\omega} = \omega \mathbf{e}_z$ and the stretching term vanishes, leading to the scalar equation (1).

2.1 Direct Numerical Simulation (DNS) modelling

Under the incompressibility assumption given by equation (6) and since ν is uniform, the momentum equation (5) can be written as follows:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \boldsymbol{\omega}) = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nabla \cdot (\nu \nabla \otimes \boldsymbol{\omega}) \quad (7)$$

Now, let us find a diffusion velocity \mathbf{u}_d such that:

$$\nu \nabla \otimes \boldsymbol{\omega} = -\mathbf{u}_d \otimes \boldsymbol{\omega} + \mathbf{B} \quad (8)$$

where \mathbf{B} is a tensor term with no prescribed form, which represents the part of the diffusion term that cannot be modelled by the Diffusion Velocity Method. Let $\mathbf{A} = \nu \nabla \otimes \boldsymbol{\omega}$, then let us find the components u_{di} of the diffusion velocity that minimise $\mathbf{B} = \mathbf{A} + \mathbf{u}_d \otimes \boldsymbol{\omega}$ in a least squares fashion:

$$u_{di} = \arg \min_{x_i} \sum_{j=1}^3 (A_{ij} + x_i \omega_j)^2, \quad i = 1, \dots, 3. \quad (9)$$

By differentiating the objective function, one may find

$$u_{di} = -\frac{\sum_j A_{ij} \omega_j}{|\boldsymbol{\omega}|^2}, \quad i = 1, \dots, 3, \quad (10)$$

which finally leads to

$$\mathbf{u}_d = -\nu \frac{(\nabla \otimes \boldsymbol{\omega}) \cdot \boldsymbol{\omega}}{|\boldsymbol{\omega}|^2}, \quad (11)$$

corresponding to the expression given by Rivoalen et al (1997). Differentiating the objective function twice leads to $2|\boldsymbol{\omega}|^2$, which ensures that the critical point is a minimum. In addition, noticing that $(\nabla \otimes \boldsymbol{\omega}) \cdot \boldsymbol{\omega} = \nabla |\boldsymbol{\omega}|^2 / 2$, one may express \mathbf{u}_d in a more convenient form:

$$\mathbf{u}_d = -\frac{\nu}{2} \frac{\nabla \varphi^2}{\varphi^2} = -\nu \frac{\nabla \varphi}{\varphi}, \quad (12)$$

with $\varphi \equiv |\boldsymbol{\omega}|$. This novel expression is thus really close to the scalar one (see equation (2)).

Let us now define the particle transport equations. Let $\boldsymbol{\Omega}_i$ denote the vortical weight of particle ξ_i , defined by:

$$\boldsymbol{\Omega}_i(t) = \int_{\mathcal{P}_i} \boldsymbol{\omega}(\mathbf{x}, t) d\mathbf{x} \approx \boldsymbol{\omega}(\mathbf{X}_i, t) \sigma_i. \quad (13)$$

The corresponding Lagrangian description is then given by:

$$\begin{cases} \frac{d\mathbf{X}_i}{dt} = \tilde{\mathbf{u}}(\mathbf{X}_i, t) + \tilde{\mathbf{u}}_d(\mathbf{X}_i, t) \\ \frac{d\boldsymbol{\Omega}_i}{dt} = (\boldsymbol{\Omega}_i(t) \cdot \nabla) \tilde{\mathbf{u}}(\mathbf{X}_i, t) + \tilde{\mathbf{L}}_i(t), \end{cases} \quad (14)$$

where $\tilde{\mathbf{L}}_i$ is the discrete approximation of \mathbf{L}_i , which represents the remaining part of diffusion that cannot be treated with DVM:

$$\mathbf{L}_i = \int_{\mathcal{P}_i} \nabla \cdot (\mathbf{B}(\mathbf{x}, t)) d\mathbf{x}. \quad (15)$$

Now, denoting by \mathbf{e} the unitary vector field aligned with $\boldsymbol{\omega}$ defined by $\mathbf{e} = \boldsymbol{\omega}/|\boldsymbol{\omega}|$, \mathbf{B} can be written as follows (see Appendix A):

$$\mathbf{B} = \nu |\boldsymbol{\omega}| (\nabla \otimes \mathbf{e}), \quad (16)$$

and, locally, one has

$$\mathbf{e}(\mathbf{X}_i) \approx \frac{\boldsymbol{\Omega}_i}{|\boldsymbol{\Omega}_i|}. \quad (17)$$

The discrete approximation $\tilde{\mathbf{L}}_i$ of \mathbf{L}_i is then obtained thanks to Degond and Mas-Gallic's PSE approach with non-uniform coefficient (Degond and Mas-Gallic 1989):

$$\tilde{\mathbf{L}}_i = \frac{\nu}{2} \sum_{j=1}^N (|\boldsymbol{\Omega}_i| \sigma_j + |\boldsymbol{\Omega}_j| \sigma_i) \left(\frac{\boldsymbol{\Omega}_j}{|\boldsymbol{\Omega}_j|} - \frac{\boldsymbol{\Omega}_i}{|\boldsymbol{\Omega}_i|} \right) \eta_\varepsilon^{\text{lap}}(\mathbf{X}_i - \mathbf{X}_j), \quad (18)$$

where $\eta_\varepsilon^{\text{lap}}$ denotes the Laplacian approximation kernel (see, *e.g.*, Eldredge et al (2002)). The formulation and approximation of the remaining term \mathbf{B} is convenient and differs from those previously presented in the literature (Rivoalen et al 1997; Lacombe and Mas-Gallic 1999).

2.2 Large Eddy Simulation (LES) modelling

Large Eddy Simulation (see, *e.g.*, Sagaut (2006); Meneveau and Katz (2000)), is a turbulence model based on scale invariance, which consists in modelling the small scales that are not resolved. New terms thus appear in the momentum equation (7) in order to take those small scales into account (see Mansfield et al (1996, 1998)):

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \boldsymbol{\omega}) = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nabla \cdot (\nu \nabla \otimes \boldsymbol{\omega}) + \nabla \cdot \{ \nu_t [(\nabla \otimes \boldsymbol{\omega}) - (\nabla \otimes \boldsymbol{\omega})^\top] \} \quad (19)$$

$$= (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nabla \cdot [\nu^* (\nabla \otimes \boldsymbol{\omega}) - \nu_t (\nabla \otimes \boldsymbol{\omega})^\top] \quad (20)$$

where ν_t is the so-called turbulent viscosity and $\nu^* = \nu + \nu_t$. It should be noted that in this equation, $\boldsymbol{\omega}$ represents the *resolved* vorticity field.

The last two terms of the right-hand side of equation (19) represent respectively the molecular diffusion and the subgrid-scale (SGS) model. In particle-mesh methods such as Vortex-in-Cell (Christiansen 1997; Cottet and Poncet 2004), the vorticity field $\boldsymbol{\omega}$ is projected on a mesh to perform several computations (in particular the velocity field evaluation), which enables to use classical methods such as Finite Differences to evaluate the SGS term (see, *e.g.*, Chatelain et al (2013)). In meshless Vortex methods, the SGS term evaluation can be performed in different ways. Usually, this term is treated together with molecular diffusion using Degond and Mas-Gallic's (Degond and Mas-Gallic 1989) Particle Strength Exchange (PSE) method (see, *e.g.*, Mansfield et al (1996, 1998); Winckelmans et al (2005); Pinon et al (2012)). Milane and Nourazar (1995, 1997) used a diffusion velocity together with Leonard's (Leonard 1980) spreading core technique and Milane (2004) used a full scalar DVM method to treat LES terms in 2D computations. In the sequel, a vectorial formulation will be derived for treating 3D LES terms with DVM.

First, following the assumption that $\nabla \cdot [\nu_t (\nabla \otimes \boldsymbol{\omega})^\top]$ can be neglected (see Mansfield (1997); Mansfield et al (1998)), the momentum equation becomes:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \boldsymbol{\omega}) = (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} + \nabla \cdot [\nu^* (\nabla \otimes \boldsymbol{\omega})]. \quad (21)$$

As mentioned above, the last term, which combines both the molecular diffusion and the SGS model, is usually treated by means of PSE. However, noticing that equation (21) is close to the DNS equation (7), except $\nu^* = \nu + \nu_t$ is no longer uniform and also models the influence of unresolved scales, one can similarly derive a DVM formulation. Indeed, using the same DVM approach as for the DNS equations (see section 2.1), the LES Lagrangian description then reads:

$$\begin{cases} \frac{d\mathbf{X}_i}{dt} = \tilde{\mathbf{u}}(\mathbf{X}_i, t) + \tilde{\mathbf{u}}_d^*(\mathbf{X}_i, t) \\ \frac{d\boldsymbol{\Omega}_i}{dt} = (\boldsymbol{\Omega}_i(t) \cdot \nabla) \tilde{\mathbf{u}}(\mathbf{X}_i, t) + \tilde{\mathbf{L}}_i^*(t), \end{cases} \quad (22)$$

where $\mathbf{u}_d^* = -\nu^* \nabla \varphi / \varphi$ with $\varphi \equiv |\boldsymbol{\omega}|$, and where $\tilde{\mathbf{L}}_i^*$ is the discrete approximation of \mathbf{L}_i^* , defined by:

$$\mathbf{L}_i^* = \int_{\mathcal{P}_i} \nabla \cdot (\mathbf{B}^*(\mathbf{x}, t)) d\mathbf{x}, \quad \text{with } \mathbf{B}^* = \nu^* |\boldsymbol{\omega}| (\nabla \otimes \mathbf{e}). \quad (23)$$

Finally, $\tilde{\mathbf{L}}_i^*$ is given by:

$$\tilde{\mathbf{L}}_i^* = \frac{1}{2} \sum_{j=1}^N (\nu_i^* |\boldsymbol{\Omega}_i| \sigma_j + \nu_j^* |\boldsymbol{\Omega}_j| \sigma_i) \left(\frac{\boldsymbol{\Omega}_j}{|\boldsymbol{\Omega}_j|} - \frac{\boldsymbol{\Omega}_i}{|\boldsymbol{\Omega}_i|} \right) \eta_\varepsilon^{\text{lap}}(\mathbf{X}_i - \mathbf{X}_j), \quad (24)$$

since ν^* is no longer uniform.

It should be noted that, due to the expression of $\mathbf{u}_d = -\nu^* \nabla \varphi / \varphi$ in both the scalar and the vectorial case (i.e. with $\varphi \equiv c$ or $\varphi \equiv |\boldsymbol{\omega}|$), with or without LES (i.e. with $\nu^* = \nu$ or $\nu^* = \nu + \nu_t$), a singular behaviour is likely to occur when the quantity to diffuse (either c or components of $\boldsymbol{\omega}$) crosses zero (changes sign). This issue has been addressed in (Mycek et al 2013).

The new formulation combines two methods, namely DVM and PSE. While only 3D experiments could clearly quantify the contribution of either method, one can expect the DVM part to be preponderant, since \mathbf{u}_d is taken such that B, which accounts for the PSE part, has minimum components (see equation (9)).

3 Particle approximation of the diffusion velocity

The continuous particle approximation $\langle \varphi \rangle(\mathbf{x}, t)$ of any scalar field $\varphi(\mathbf{x}, t)$ is obtained from the following interpolation:

$$\langle \varphi \rangle(\mathbf{x}, t) = \int_{\mathcal{S}} \varphi(\mathbf{y}, t) \zeta_\varepsilon(\mathbf{x} - \mathbf{y}) d\mathbf{y}, \quad (25)$$

where ζ_ε is a smoothing function which tends to the Dirac measure when ε tends to 0 (Beale and Majda 1985). The discrete form $\tilde{\varphi}$ is then classically obtained thanks to a midpoint quadrature:

$$\tilde{\varphi}(\mathbf{x}, t) = \sum_{i=1}^N \varphi(\mathbf{X}_i, t) \sigma_i \zeta_\varepsilon(\mathbf{x} - \mathbf{X}_i). \quad (26)$$

The discretised form of the concentration gradient intervening in the diffusion velocity may be basically obtained by derivation of equation (25), which eventually only involves the kernel gradient, $\nabla \zeta_\varepsilon$:

$$\langle \nabla \varphi \rangle(\mathbf{x}, t) = \int_{\mathcal{S}} \varphi(\mathbf{y}, t) \nabla \zeta_\varepsilon(\mathbf{x} - \mathbf{y}) d\mathbf{y}, \quad \widetilde{\nabla \varphi}(\mathbf{x}, t) = \sum_{i=1}^N \varphi(\mathbf{X}_i, t) \sigma_i \nabla \zeta_\varepsilon(\mathbf{x} - \mathbf{X}_i). \quad (27)$$

The interpolation kernel ζ_ε is related to the radially symmetric function ζ as follows:

$$\zeta_\varepsilon(\mathbf{x}) = \frac{1}{\varepsilon^d} \zeta\left(\frac{\mathbf{x}}{\varepsilon}\right), \quad (28)$$

where d is the physical dimension ($\mathbf{x} \in \mathcal{S} \subset \mathbb{R}^d$). The expressions of ζ are built so as to meet some moment properties, directly connected to the order of approximation r (Beale and Majda 1985; Liu et al 1995, 1996; Eldredge et al 2002). As mentioned previously, the field $\nabla \zeta_\varepsilon$ is basically the gradient of ζ_ε . It follows from equation (28) that a similar relation between $\nabla \zeta_\varepsilon$ and $\nabla \zeta$ is given by:

$$\nabla \zeta_\varepsilon(x) = \frac{1}{\varepsilon^{d+1}} \nabla \zeta\left(\frac{x}{\varepsilon}\right). \quad (29)$$

Finally, in particle methods, the smoothing parameter ε is related to the space discretisation step h thanks to the so-called overlapping parameter κ as follows:

$$\varepsilon = \kappa h. \quad (30)$$

The particle volumes $\sigma_i(t)$ are not constant over time when using DVM, due to the fact that the diffusion velocity is not divergence-free. They are thus updated after each time step using the approximation $\sigma_i(t) \approx C_i(t)/c(\mathbf{X}_i, t)$ (see equation (3)). Once the volumes are updated, the ε parameter is updated accordingly using equation (30), where h stands for the largest inter-particle distance. Nevertheless, should one want to keep ε a constant, frequent remeshing can be applied to keep particles aligned on a fixed grid. As stated previously in the introduction, this option will not be further investigated in this paper.

3.1 Scalar diffusion velocity

Using these approximations to discretise equation (2), the most largely spread formulation of the scalar diffusion velocity reads (Degond and Mustieles 1990; Ogami and Akamatsu 1991):

$$\langle \mathbf{u}_d \rangle(\mathbf{X}_i, t) = -\nu \frac{\langle \nabla c \rangle(\mathbf{X}_i, t)}{\langle c \rangle(\mathbf{X}_i, t)} = -\nu \frac{\int_{\mathcal{S}} c(\mathbf{y}, t) \nabla \zeta_\varepsilon(\mathbf{X}_i - \mathbf{y}) d\mathbf{y}}{\int_{\mathcal{S}} c(\mathbf{y}, t) \zeta_\varepsilon(\mathbf{X}_i - \mathbf{y}) d\mathbf{y}}, \quad (31)$$

whose corresponding discrete form is:

$$\tilde{\mathbf{u}}_d(\mathbf{X}_i, t) = -\nu \frac{\sum_{j=1}^N C_j(t) \nabla \zeta_\varepsilon(\mathbf{X}_i - \mathbf{X}_j)}{\sum_{j=1}^N C_j(t) \zeta_\varepsilon(\mathbf{X}_i - \mathbf{X}_j)}. \quad (32)$$

3.2 Vectorial diffusion velocity

From equations (14) (or equivalently (22)) and the approach used in the scalar case, the discrete approximation $\tilde{\mathbf{u}}_d$ of the vectorial diffusion velocity basically reads:

$$\tilde{\mathbf{u}}_d(\mathbf{X}_i, t) = -\nu \frac{\sum_{j=1}^N |\boldsymbol{\Omega}_j|(t) \nabla \zeta_\varepsilon(\mathbf{X}_i - \mathbf{X}_j)}{\sum_{j=1}^N |\boldsymbol{\Omega}_j|(t) \zeta_\varepsilon(\mathbf{X}_i - \mathbf{X}_j)}, \quad (33)$$

where ν can basically be replaced by $\nu^* = \nu + \nu_t$ if one wants to include a LES turbulence model (see section 2.2). One can thus see that this formulation simply reduces to a scalar approximation. On the other hand, the behaviour of interpolation kernels should be similar whatever the physical dimension d . In addition, as

mentioned previously, the diffusion velocity part is expected to be preponderant as compared to the remaining, PSE-treated part. The desirable consequence of this is that most of the diffusion is expected to be treated in a convective way. In either one or higher dimensions, the behaviour of the numerical scheme is expected to be good, since Lagrangian particle schemes such as vortex methods are extremely well-suited to the modelling of convection. Nevertheless, the behaviour of such methods also strongly depends on the spatial arrangement of the particles, which is why additional techniques such as remeshing may be required to maintain a decent particle arrangement.

In light of the previous comments, the diffusion velocity method is now going to be analysed in one dimension ($d = 1$).

4 Analytic results

In the sequel, Gaussian kernels are considered, built from $\zeta(x) = P(x)e^{-x^2}$ where P is a polynomial. However, it is clear that any other family of kernels (*e.g.*, splines) may be suitable. With an approach inspired from Liu et al (1996), presented later on by Eldredge et al (2002) with a particle method formalism, and thanks to convenient properties of Gaussian functions, an explicit expression can be given for 1D Gaussian kernels ζ of any order without solving any system of linear equations. One may refer to Appendix B for details. For notation purposes, the kernel gradient will be denoted by $\eta_\varepsilon^{(1)} = \nabla\zeta_\varepsilon = d\zeta_\varepsilon/dx$ in one dimension. The concentration gradient will be denoted by $\nabla c = \partial c/\partial x$.

4.1 Fourier analysis

One interesting feature of formulation (31) is that it can be written so as to involve convolution products of c with the appropriate kernels. In one dimension, equation (31) can thus be written as follows:

$$\langle u_d \rangle(x, t) = -\nu \frac{c * \eta_\varepsilon^{(1)}}{c * \zeta_\varepsilon}(x, t), \quad (34)$$

where $*$ denotes the spatial convolution product (*i.e.* with regard to x). In terms of Fourier transform, this means that

$$\langle u_d \rangle(x, t) = -\nu \frac{\mathcal{F}^{-1} \left[\hat{c}(k, t) \hat{\eta}_\varepsilon^{(1)}(k) \right]}{\mathcal{F}^{-1} \left[\hat{c}(k, t) \hat{\zeta}_\varepsilon(k) \right]}(x, t) = -\nu \frac{\mathcal{F}^{-1} \left[ik \hat{c}(k, t) \hat{\zeta}_\varepsilon(k) \right]}{\mathcal{F}^{-1} \left[\hat{c}(k, t) \hat{\zeta}_\varepsilon(k) \right]}(x, t), \quad (35)$$

where $\hat{\cdot}$ and $\mathcal{F}(\cdot)$ denote the Fourier transform. The following convention is used for the Fourier transform pair:

$$\hat{f}(k) = \int f(x) e^{-ikx} dx \quad (36)$$

$$f(x) = \frac{1}{2\pi} \int \hat{f}(k) e^{ikx} dk \quad (37)$$

The right-hand side of equation (35) is obtained by recalling that $\eta_\varepsilon^{(1)}$ is the gradient of ζ_ε and by using the following property of the Fourier transform:

$$\mathcal{F} \left[\frac{d^n}{dx^n} f(x) \right] (k) = (ik)^n \hat{f}(k). \quad (38)$$

Equations (34) and (35) thus encourages to consider the interpolation kernel ζ as a filter applied to c .

Analytic expressions of $\hat{\zeta}$, for Gaussian ζ functions can be computed whatever the kernel order $r = 2(m + 1)$:

$$\hat{\zeta}(k) = e^{-(k/2)^2} \sum_{n=0}^m \frac{1}{n!} \left(\frac{k}{2} \right)^{2n}, \quad (39)$$

One may refer to Appendix B for details. The Fourier transform of ζ_ε is basically derived from $\hat{\zeta}$ as follows:

$$\hat{\zeta}_\varepsilon(k) = \hat{\zeta}(\varepsilon k), \quad (40)$$

where $\hat{\zeta}(\varepsilon k)$ stands for the Fourier transform of ζ evaluated at εk . It is striking to see from expression (39) that, unlike ζ , its Fourier transform $\hat{\zeta}$ can be directly inducted from the previous order expression without modifying it. This only requires the addition of one single new term. This relation is much simpler than the more general relations presented in the literature (see, *e.g.*, Cortez (1997)) to the authors' knowledge, though one should keep in mind that it is restricted to the particular case of Gaussian kernels.

Figure 1 depicts the representation of $\hat{\zeta}_\varepsilon(k)$ as a function of the non-dimensional wave number kh/π . It is well known that interpolation kernels act as a low-pass filters (see, *e.g.*, Liu et al (1996)), that is to say that short waves (high wave numbers) are filtered in the signal $c(x, t)$. This can clearly be observed on Figure 1. Moreover, it shows that higher order kernels get closer to rectangular shaped filters in the Fourier space. This implies that signals (consisting of the combination of several wave lengths) are globally better reconstructed with higher order kernels (cf. equation (35)). In terms of convolution (cf. equations (31) and (34)), this means that the interpolation process is more accurate.

Examples of pure diffusion are now presented. The one-dimensional pure diffusion equation reads:

$$\frac{\partial c}{\partial t} = \nu \frac{\partial^2 c}{\partial x^2}, \quad (41)$$

which corresponds to equation (1) in one dimension without any convection, *i.e.* with $u = 0$. The diffusion of a propagative wave (single wave number), inspired from (Eldredge et al 2002), and of a Gaussian patch (endless number of wave numbers) is now considered.

4.2 Propagative wave

A propagative wave signal with wave number k_0 is defined by:

$$c(x, t, k_0, \omega_0) = \exp i(k_0 x - \omega_0 t). \quad (42)$$

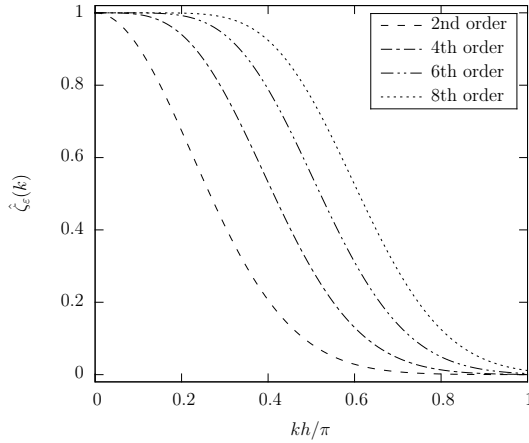


Fig. 1: Representation of $\hat{\zeta}_\varepsilon(k)$ function of the non-dimensional wave number kh/π for different orders of approximation, with $\kappa = 2.0$.

Injected into the pure diffusion equation (41), it must verify the following dispersion equation:

$$\omega_0 = -i\nu k_0^2. \quad (43)$$

In that case, the exact solution of the velocity diffusion is $u_d = -i\nu k_0$. From equations (25) and (27), and using equation (38), the following identities hold:

$$\frac{\langle c \rangle}{c}(k_0) = \hat{\zeta}_\varepsilon(k_0) \quad \text{and} \quad \frac{\langle \nabla c \rangle}{\nabla c}(k_0) = -\frac{i}{k_0} \hat{\eta}_\varepsilon^{(1)}(k_0) = \hat{\zeta}_\varepsilon(k_0). \quad (44)$$

A purely theoretical diffusion velocity for this propagative wave would then satisfy:

$$\frac{\langle u_d \rangle}{u_d}(k_0) = 1 \quad \forall k_0, \quad (45)$$

which means that the continuous approximation of the diffusion velocity would be infinitely accurate, whatever the kernel order and whatever the wave number k_0 , as long as ω_0 satisfies the dispersion equation.

In the sequel, the relative error $\mathcal{E}_{q'}$ committed on the approximation q' , may it be continuous or discrete, with regard to the exact value of quantity q will be defined by:

$$\mathcal{E}_{q'} = \left| \frac{q' - q}{q} \right|. \quad (46)$$

In terms of relative error, equation (45) means:

$$\mathcal{E}_{\langle u_d \rangle} = 0 \quad \forall k_0. \quad (47)$$

4.3 Gaussian patch

The elementary solution of the diffusion equation, *i.e.* for initial condition $c(x, 0) = \delta(x)$, is given by:

$$c(x, t) = \frac{1}{\sqrt{4\pi\nu t}} \exp\left(-\frac{x^2}{4\nu t}\right), \quad \forall t > 0. \quad (48)$$

The analytic Fourier transform of c is then given by:

$$\hat{c}(k, t) = e^{-\nu k^2 t}, \quad \forall t > 0. \quad (49)$$

From this expression and the expressions of the kernel Fourier transform (39) and (40), one can compute the analytic error induced by the continuous approximations $\langle c \rangle$, $\langle \nabla c \rangle$ and $\langle u_d \rangle$, defined either in the function domain, or in the Fourier domain. One can refer to Appendix B for details. In particular, when using a 2nd order approximation, the initial diffusion velocity continuous error does not depend upon x and reads:

$$\mathcal{E}_{\langle u_d \rangle} = \frac{\varepsilon^2}{\varepsilon^2 + \sigma^2}, \quad \text{for order 2.} \quad (50)$$

with $\sigma = \sqrt{4\nu t}$ for time $t > 0$. The generic expressions of $\mathcal{E}_{\langle c \rangle}$, $\mathcal{E}_{\langle \nabla c \rangle}$ and $\mathcal{E}_{\langle u_d \rangle}$ for any order of approximation can be found in Appendix B.

Figure 2 shows the initial distribution of the squared relative error committed on the continuous approximation of u_d as computed in Appendix B. Except at the domain boundaries, the accuracy is globally better with higher order kernels. The higher error observed at the domain limits can be interpreted thanks to the Fourier analysis carried out on the kernels previously. Indeed, the Gaussian patch $c(x, t_0)$ at the domain boundaries correspond to a part of the signal that is mostly composed of short waves. As a consequence, these short waves are cut off by the filter (*i.e.* the interpolation kernel ζ_ε) and this part of the signal is thus badly reconstructed (*i.e.* the interpolation is less accurate). However, increasing the kernel order improves the overall accuracy. The sharp peaks observed in the error correspond to a change of sign in the difference $(\langle u_d \rangle - u_d)$, which is concealed by the raising to the power of 2. Finally, one can observe that the 2nd order error on u_d remains constant over the domain, which is consistent with equation (50).

As for the Fourier transform of the kernel, the influence of κ and the consequence of the discretisation will be addressed in the following section 5.

5 Numerical behaviour

In order to test the numerical implementation of the different approximations, the resolution of the one-dimensional pure diffusion equation (41) is now considered.

5.1 Diffusion velocity

The transport equations associated to the particles are then presented in a simplified and one-dimensional version of equations system (4), where the convection

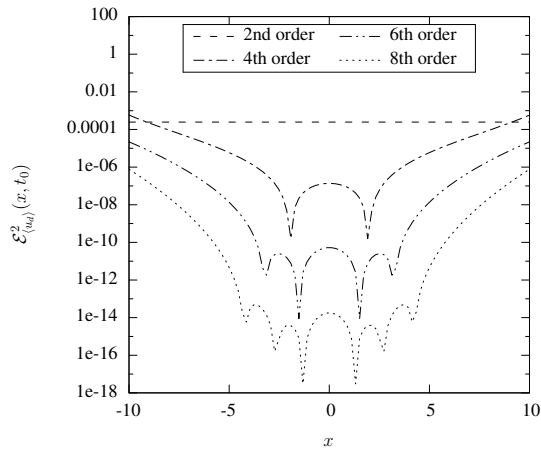


Fig. 2: Initial spatial distribution of the analytic error on u_d at $t_0 = 2.5$ for different orders of approximation, with $\varepsilon = 0.2$.

velocity $u(\mathbf{x}, t)$ is assumed to be null:

$$\begin{cases} \frac{dX_i}{dt} = \tilde{u}_d(X_i, t), & \text{with } \tilde{u}_d(X_i, t) = -\nu \left. \frac{\widetilde{\nabla} c}{c} \right|_i \\ \frac{dC_i}{dt} = 0 \end{cases} \quad (51)$$

The choice of setting the convective velocity to zero was made in order to give a better insight into the effects of diffusion, more importantly because it is treated here in a convective way by means of the diffusion velocity. However, DVM can easily be used with a convective velocity u , simply by adding its discrete form \tilde{u} in the convective part of the transport equations.

The elementary solution of the pure diffusion equation is considered. An initial time t_0 is chosen for the computation and the initial concentration $c(x, t_0)$ is initialised thanks to equation (48) on a $2L$ -long segment centred around the origin. The segment is discretised into N particles, whose volume (length in 1D) is basically given by $\sigma = 2L/N$. Each particle weight is initialised as $C_i = c(X_i, t_0)\sigma$ with $X_i = -L + (i - 0.5)\sigma$. From then, $t^* = t - t_0$ will denote the time elapsed since the initial time t_0 . All the results shown in the sequel were obtained using the parameter values presented in Table 1.

Parameter	Value
t_0	2.5
ν	0.25
δt	0.001
L	10
N	from 50 to 334
κ	from 1.1 to 3.0

Table 1: Summary of the parameter values.

Owing to the elementary solution (48) of the diffusion equation, the analytic diffusion velocity is given by:

$$u_d(x, t) = \frac{x}{2t} \quad (52)$$

5.2 Influence of the overlapping parameter

Figure 3 shows the initial distribution at $t^* = 0$ of the squared relative error committed on the continuous approximation of $\langle u_d \rangle$, for different values of κ . Both the analytic error $\mathcal{E}_{\langle \cdot \rangle}$ (cf. B) and the discrete particle error \mathcal{E}_{\cdot} (cf. equation (32)) are shown. The x axis was slightly enlarged so as to distinguish the two kinds of curves.

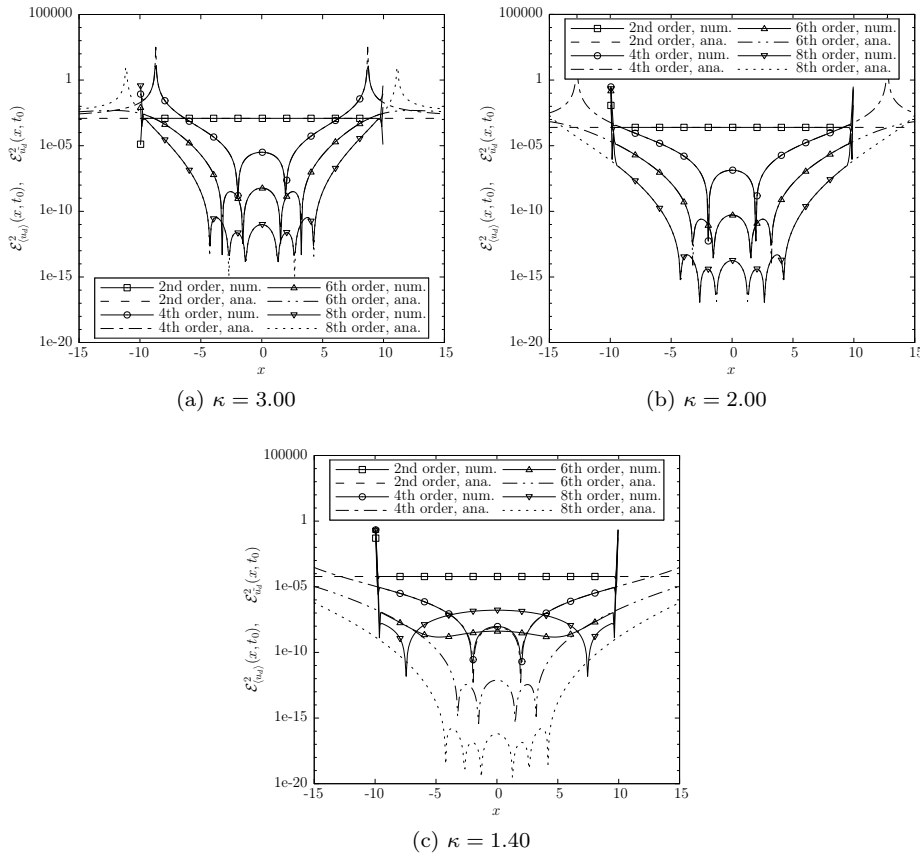


Fig. 3: Initial spatial distribution of both the continuous analytic and the discrete error on u_d at $t_0 = 2.5$ for different orders of approximation and three values of κ , with $N = 200$ and the parameters given in Table 1.

First of all, one can observe an almost exact agreement between the analytic and the numerical error for $\kappa = 2$ or 3 (Figures 3a and 3b), whatever the kernel order. However, higher error on u_d is observed at the very limit of the numerical domain (*i.e.* $x \approx \pm 10$) on the numerical curves. This problem is caused by the fact that the approximation suffer from a lack of particles at the boundaries. The infinite domain condition, which ensures the correct interpolation and approximation of the differential operators (cf. moment conditions), is then critically not respected. Actually, the approximation behaves as though the concentration were null in zones that are not discretised (*i.e.* beyond the domain limits). The use of “one-sided” kernels, which satisfy moment properties on half the domain only, was proposed to deal with this issue (Eldredge et al 2002). However, this solution implies a special explicit treatment at the boundaries, which may strongly depend upon κ , and does not give fully satisfying results. Similarly, Schrader et al (2010) recently suggested to build the kernel so as to correct the discrete error by introducing the concept of discrete moments. Unfortunately, for non-uniformly distributed particles, this requires to re-compute the kernel for each particle at each time step, which is extremely time consuming and thus prohibitive. Another technique to correct this flaw, which is commonly used in SPH, consists in renormalising the kernel (Bonet and Lok 1999) and its gradient (Oger et al 2007). Although it may correct the bad approximation of c and of its gradient, the u_d approximation, which involves the ratio between those two approximations, should thus already include some kind of renormalisation.

The other observation that can be drawn from these first two figures (Figures 3a and 3b) is that, whatever the order of approximation, the u_d error is globally higher with $\kappa = 3$ than it is with $\kappa = 2$. This seems to indicate that lower values of κ make the approximation more accurate.

However, the last figure (Figure 3c) indicates that for $\kappa = 1.4$, the numerical error is dramatically higher at the domain centre for 6th and 8th orders computations. In addition, it appears that they really differ from the analytic curves. Indeed, analytic curves show that, whatever the order of approximation, the approximation keeps improving as κ decreases.

Both behaviours, as well as the difference between them, are a direct consequence of the discretisation and can be explained in the Fourier domain by studying the kernel Fourier transform. As a matter of fact, one should recall that ζ_ε can be seen as a filter. It is well known that a convolution in the function domain is equivalent to a product of the Fourier transforms in the Fourier domain (cf. equations (34) and (35)). At this point, the continuous and the discrete case must be considered differently. While in the continuous Fourier domain, the “ideal” filter would be a function identically equal to one (it can be seen as the identity element), the discrete ideal filter, on the other hand, would be a rectangular function. As a matter of fact, the discretisation process, which is referred to as the sampling process in signal processing, introduces high frequency replicas in the discrete Fourier spectrum. Thus, the famous Nyquist-Shannon sampling theorem states that the cut-off frequency of a filter, *i.e.* the frequency beyond which its Fourier spectrum is *about* null, should correspond to the sampling frequency – or equivalently the wave number.

The discretisation step being h , the sampling wave number is then π/h , and the discrete spectrum of $\hat{\zeta}_\varepsilon$ should therefore tend to zero for wave numbers beyond this value. Figure 4 shows the representation of $\hat{\zeta}_\varepsilon(k)$ function of the non-dimensional

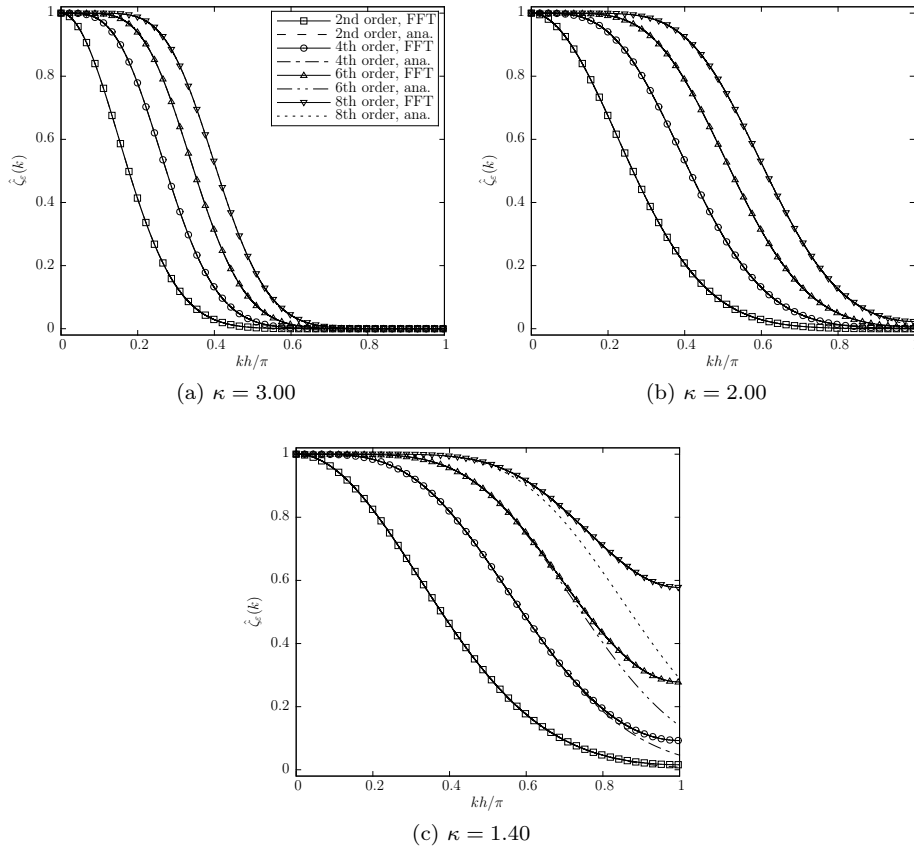


Fig. 4: Representation of $\hat{\zeta}_\varepsilon(k)$ function of the non-dimensional wave number kh/π for different orders of approximation, and different values of κ . The continuous analytic curves are compared to the corresponding FFT approximations computed with 2^{20} points in order to highlight the consequences of discretisation.

wave number kh/π for the values of κ previously studied. The continuous analytic curves are compared to the corresponding discrete Fast Fourier Transform (FFT) approximations computed with 2^{20} points. First, one can observe that for $\kappa \geq 2$, there is a fairly good agreement between the analytic and the FFT Fourier transforms. On the contrary, with $\kappa = 1.4$, the curves differ in the higher frequency zone, especially for higher order approximations. This is due to the fact that the discretisation step of ζ_ε actually only depends on κ . Low values of κ correspond to a coarse discretisation and thus to undersampling, which induces aliasing: there is a mix in high frequencies which leads to an erroneous calculation of the spectrum in this area.

In addition, Figure 4 also shows that for $\kappa = 1.4$ and high order kernels, the value of the Fourier transform at $k = \pi/h$ is far from being null. This problem is even emphasized in the discrete case by the undersampling of ζ_ε mentioned in

the previous paragraph. As a consequence, the Nyquist-Shannon conditions are not respected and the *reconstruction* is very erroneous in the numerical computations. On the contrary, the analytical error keeps decreasing since the analytic Fourier transform get closer to the continuous ideal filter, identically equal to one (whatever k).

Finally, the improvement observed on Figures 3a and 3b with $\kappa = 2$ as compared to $\kappa = 3$ is explained by the fact that in the first case, fewer high frequencies are cut-off by the filter (cf. Figure 4b).

In order to evaluate the overall error at a given time t on the whole domain, the L_2 error on quantity q is now introduced. It is denoted $\|\mathcal{E}\|_{\tilde{q}}$ and defined by:

$$\|\mathcal{E}\|_{\tilde{q}} = \sqrt{\frac{\sum_{j=1}^N [\tilde{q}(X_j) - q(X_j)]^2}{\sum_{j=1}^N q^2(X_j)}}. \quad (53)$$

Note that $\|\mathcal{E}\|_{\tilde{q}}$ is *not* the norm of $\mathcal{E}_{\tilde{q}}$ previously introduced in equation (46).

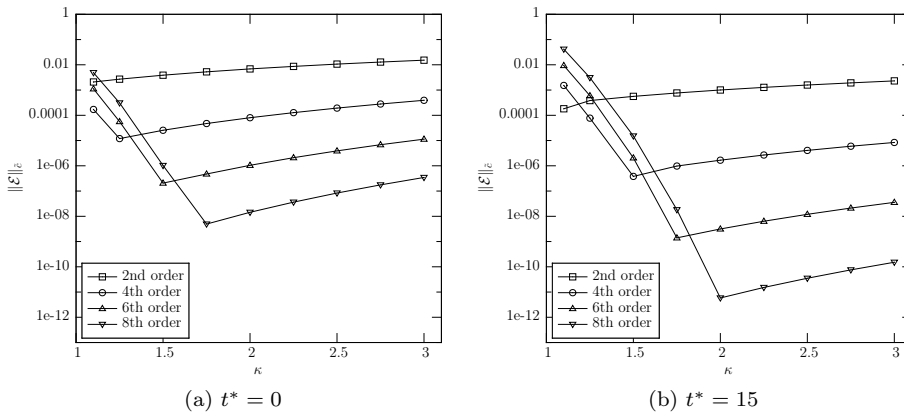


Fig. 5: Concentration L_2 error $\|\mathcal{E}\|_{\tilde{q}}$ at $t^* = 0$ and $t^* = 15$ as a function of the overlapping parameter κ for different orders of approximation, with $N = 200$ and the parameters given in Table 1.

The influence of the overlapping parameter κ on the L_2 concentration error is illustrated on Figure 5. These results show that the optimal choice of κ depends on the order of approximation. More precisely, except for a 2nd order approximation, the optimum value of κ increases together with the order of approximation. This behaviour is explained in the Fourier domain as described previously in this section (Figures 3 and 4). Besides, the optimal value of κ with regard to the final concentration error appears to increase as the computations get longer. This might be explained by the stabilising effect of higher values of κ , in particular at the domain limits (cf. Figure 3b as compared to Figure 3a).

As a matter of comparison, in the SPH community, small values of κ such as $\kappa = 1.33$ are commonly used (Colagrossi and Landrini 2003). This is justified by

the fact that 2nd order kernels are used. However, values of κ should be carefully chosen for higher order kernels.

5.3 Trajectories

By a simple integration of equation (52), the analytic trajectory $X_i(t)$ of particle ξ_i is obtained as a function of its initial position X_i^0 at initial time t_0 :

$$X_i(t) = X_i^0 \sqrt{\frac{t}{t_0}}. \quad (54)$$

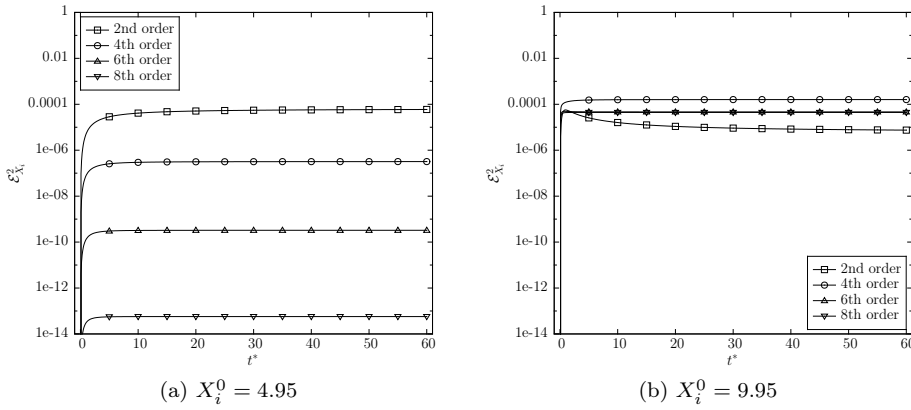


Fig. 6: Evolution of the trajectory error on particles $X_i^0 = 4.95$ (left) and $X_i^0 = 9.95$ (right) as a function of time for different orders of approximation, with $\kappa = 2$, $N = 200$ and the parameters given in Table 1.

The results of Figure 6 illustrate one of the main advantages of DVM, that is to say its Lagrangian aspect. Comparisons between the analytic trajectory and their approximations for different orders are shown in terms of the position squared relative error $\mathcal{E}_{X_i}^2$ as compared to the analytic trajectory x . They indicate that the particle really follows its own trajectory with an increasing precision as the kernel order raises, as shows Figure 6a.

However, the trajectory is not as accurately reproduced for particles located near the boundaries, as shows the trajectory of the “last” particle presented on Figure 6b. This was predictable given the u_d high numerical error observed for the particles at the very limit of the discretised domain (cf. Figure 3b). In addition, the accuracy on these particles trajectory is no longer related to the approximation order. On the contrary, the 2nd order approximation seems to be more robust and stable.

5.4 Evolution of the concentration

Figure 7 depicts the well known diffusion of a Gaussian patch of a passive scalar $c(x, t)$. Theoretical and numerical results are superimposed. It shows the particles spreading owing to the diffusion velocity $\tilde{u}_d(X_i, t)$ as time evolves thanks to equation (51). The use of the logarithmic scale shows that the concentration evolution is well resolved even for very small concentration (less than 10^{-10}). This result is an additional proof the the precision of the present method.

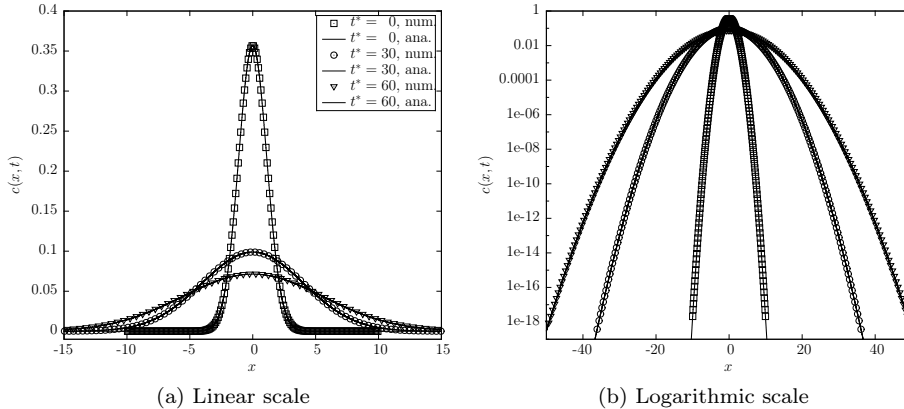


Fig. 7: Concentration and particles position evolution over different instants t^* . Continuous lines stand for theoretical results and points represent 6th order approximation with $\kappa = 2$, $N = 200$ and the parameters given in Table 1.

It has to be mentioned that the trajectory error pointed out on Figure 6b has no visible effect on the concentration approximation, which remains fairly accurate even after a long simulation time $t^* = 60$. It only means that the analytic trajectory due to diffusion is not correctly respected. In addition, one can see that the error previously observed at the domain boundaries on Figure 3 is actually not that high, in the sense that it is hardly noticeable, even in a logarithmic scale. This is due to the fact that the errors depicted on these figures are actually relative errors. Since the concentration at the domain boundaries is very small, the concentration approximation remains decent in the absolute error sense.

5.5 Convergence analysis

A convergence analysis was performed for different space discretisations h with the present approximation (32) according to the order of approximation. Figure 8 shows the result for four orders of approximation.

The results are very clean at $t^* = 5$ and the related linear regression slopes are 2.2 for 2nd order, 4.0 for 4th order, 5.9 for 6th order, and finally 7.8 for 8th order. For later times t^* , the 8th order error stabilizes around 10^{-11} even if the discretisation is refined, which is most likely caused by the boundary errors.

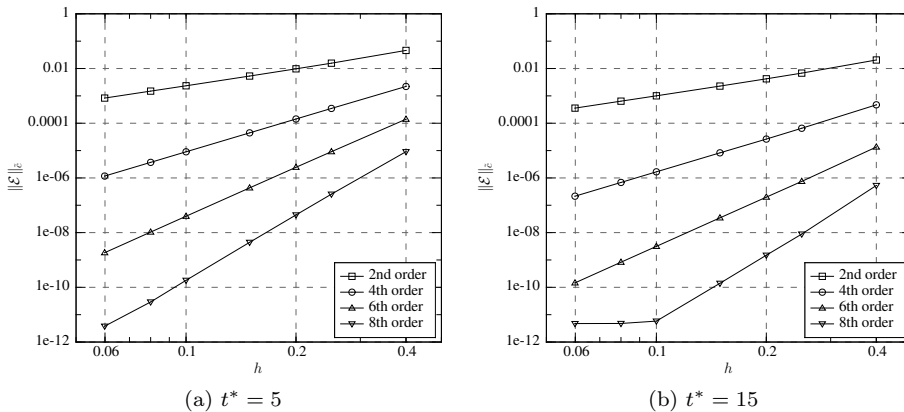


Fig. 8: Concentration L_2 error $\|\mathcal{E}\|_c$ at $t^* = 5$ and $t^* = 15$ as a function of space discretisations h for different orders of approximation, with $\kappa = 2$ and the parameters given in Table 1.

These results are very interesting especially concerning the choice of the approximation order: for a given space discretisation, $h = 0.1$ for instance, one really earns two decades of precision for the result between the 2nd and the 4th order approximation for only a small increase of CPU time consumption. Similarly, if one wants to reach a given precision without increasing the space discretisation, it might be useful to chose higher order kernels.

6 Conclusion

A novel formulation of a diffusion-velocity particle model was derived for the three-dimensional Navier-Stokes equations. This formulation can be easily extended to account for small turbulence scales with a LES model. However, it should be noted that the (LES or DNS) three-dimensional Navier-Stokes equations cannot generally be entirely re-written with advective terms. As a matter of fact, there is a remaining term, expected to be small, that still needs to be modelled by means of another technique, e.g. here PSE. This new vectorial diffusion model can be integrated into various numerical methods, such as the well-known vortex method, but also emerging methods such as the dipole domains method currently developed by Dynnikova (2011). In addition, one crucial advantage of these novel diffusion-velocity expressions is that they reduce to scalar approximations, which enables analogy with the original method for scalar transport.

As a consequence, the scalar Diffusion Velocity Method was then analysed on uni-dimensional cases. Kernel construction was briefly investigated, more particularly in the Fourier domain, and thus related to the notion of filter in signal processing. The effect of the discrete parameters on the shape of the kernel Fourier transform was presented. By analogy with the filtering process in signal processing, general recommendations on the choice of the overlapping parameter were given in order to obtain a kernel (or filter) with satisfying reconstruction properties.

To complement analytic results, numerical simulations were run using DVM for the pure diffusion of a Gaussian patch. A very clean and accurate diffusion was observed over long simulation times and the convective behaviour of the method was verified. Error analyses were performed and showed that the diffusion velocity u_d was accurately approximated. As a result, the particles trajectories were also correctly reproduced, thus ensuring a correct diffusion through advection. In addition, it was shown that a poor approximation of the diffusion velocity at the domain boundaries hardly affect the diffusion process, even in the long run.

The presented analysis holds for the uni-dimensional transport of a passive scalar such as a concentration. Such quantities are generally non-negative by definition, which ensures the good behaviour of the method. The method straightforwardly extends to the transport of a passive scalar in two or three dimensions, as demonstrates the general description of the original DVM presented in the introduction of section 2. However, the extension to the 3D Navier-Stokes equations arises one important issue. Indeed, the spatial distribution of the vorticity field ω may cross zero with steep gradients. For instance, this is the case of the diffusion of counter-rotative Gaussian vortices. In such cases, DVM suffers from a severe singularity problem and shows dramatically unstable behaviours – computations are likely to collapse. This issue has been discussed for the original DVM scheme in one dimension, and a weighted self-regularising DVM-PSE hybrid model was proposed to overcome such problems by Mycek et al (2013). This hybrid model should be straightforwardly extendable to the 3D diffusion of a vector field. Ongoing work focuses on using this vectorial DVM formulation in 3D simulations, such as marine current turbine wake and performance modelling (Pinon et al 2012). Such simulations will be used to quantify the relative contributions of DVM and PSE.

A Vectorial DVM developments

The aim of this appendix is to show that B can be expressed as in equation (16). First, from the definition (8), it follows that:

$$B_{ij} = \nu \frac{\partial \omega_j}{\partial x_i} - \nu \frac{\partial \omega_p}{\partial x_i} \frac{\omega_p \omega_j}{|\omega|^2} \quad (55)$$

were repeated indices (here p) indicate a sum over those indices. Using the unit vector $\mathbf{e} = \omega/|\omega|$, the previous equation becomes

$$B_{ij} = \nu \frac{\partial \omega_j}{\partial x_i} - \nu \frac{\partial \omega_p}{\partial x_i} e_p e_j \quad (56)$$

One can then make the following transformations:

$$\frac{\partial \omega_p}{\partial x_i} e_p e_j = \frac{\partial \omega_p e_p e_j}{\partial x_i} - \omega_p \frac{\partial e_p e_j}{\partial x_i} = \frac{\partial \omega_j e_p e_p}{\partial x_i} - \omega_p e_p \frac{\partial e_j}{\partial x_i} - \omega_p e_j \frac{\partial e_p}{\partial x_i} \quad (57)$$

$$= \frac{\partial \omega_j}{\partial x_i} - |\omega| \frac{\partial e_j}{\partial x_i} - \omega_j e_p \frac{\partial e_p}{\partial x_i}, \quad (58)$$

whose last term vanishes since $e_p \partial e_p / \partial x_i = \frac{1}{2} \partial (e_p e_p) / \partial x_i = 0$. Finally, B_{ij} becomes:

$$B_{ij} = \nu |\omega| \frac{\partial e_j}{\partial x_i}. \quad (59)$$

B Analytic expressions of the Gaussian kernel and its Fourier transform

The purpose of this appendix is to determine the explicit expressions of the one-dimensional Gaussian kernel and its Fourier transform, for any order of accuracy. The approach is mostly inspired by Reproducing Kernel Particle Method (RKPM) techniques (see for example Liu et al (1995, 1996)). One should remember that the following derivation is restricted to Gaussian kernels, and aims at providing expressions that are explicit and simpler than the more general recurrence relations proposed in the existing literature.

Notations

Let $\zeta_{m,\varepsilon}$ be the interpolation kernel of order $r = 2(m + 1)$, where ε is the mollifying parameter, issuing from the function ζ_m (cf. equation (28)). Since the kernel $\zeta_{m,\varepsilon}$ (resp. its Fourier transform $\hat{\zeta}_{m,\varepsilon}$) can easily be deduced from ζ_m (resp. $\hat{\zeta}_m$), whose expression is much simpler, the proof will be performed on ζ_m and $\hat{\zeta}_m$. The 2nd order function ζ_0 is defined by:

$$\zeta_0(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}. \quad (60)$$

It follows that

$$\hat{\zeta}_0(k) = e^{-k^2/4}. \quad (61)$$

Hermite polynomials

Let H_n be the n -th Hermite polynomial, whose degree is n , $n \geq 0$. H_n is classically defined by:

$$\frac{d^n}{dx^n} e^{-x^2} = (-1)^n e^{-x^2} H_n(x), \quad \forall n \geq 0. \quad (62)$$

These polynomials have an explicit expression, which is here split up according to the parity of n :

$$H_{2n}(x) = (2n)! \sum_{p=0}^n \frac{(-1)^{n-p}}{(2p)!(n-p)!} (2x)^{2p}, \quad \forall n \geq 0. \quad (63a)$$

$$H_{2n+1}(x) = (2n+1)! \sum_{p=0}^n \frac{(-1)^{n-p}}{(2p+1)!(n-p)!} (2x)^{2p+1}, \quad \forall n \geq 0. \quad (63b)$$

Hermite polynomials with parameter a

Hermite polynomials can easily be generalized to Gaussian functions with scaling parameter $a > 0$. Let $H_n^*(x, a)$ be the n -th generalized Hermite function of parameter a , defined by:

$$\frac{d^n}{dx^n} e^{-ax^2} = (-1)^n e^{-ax^2} H_n^*(x, a), \quad \forall n \geq 0, \forall a > 0. \quad (64)$$

It is related to H_n by the following relation:

$$H_n^*(x, a) = a^{n/2} H_n(\sqrt{ax}), \quad \forall n \geq 0, \forall a > 0. \quad (65)$$

Hermitian linear combination

The r^{th} order function ζ_m is built from a Gaussian template as follows (Eldredge et al 2002):

$$\zeta_m(x) = \frac{e^{-x^2}}{\sqrt{\pi}} \sum_{n=0}^m \gamma_n^{[m]} x^{2n}. \quad (66)$$

where the $\gamma_n^{[m]}$ are m related coefficients determined by the moment conditions. This means that ζ_m is the product of $\frac{e^{-x^2}}{\sqrt{\pi}}$ with an even polynomial (*i.e.* of even powers only) of degree $2m$. The proof relies on the fact that ζ_m can be written as a linear combination of the $m+1$ first even derivatives of ζ_0 (including the zero-th derivative):

$$\zeta_m(x) = \sum_{n=0}^m c_n \zeta_0^{(2n)}(x). \quad (67)$$

As a matter of fact, $\zeta_0^{(2n)}$ are even polynomials of degree $2n$ multiplied by the Gaussian function $\frac{e^{-x^2}}{\sqrt{\pi}}$. It can be easily shown that any family of $m+1$ even polynomials of degrees 0 to $2m$ forms a basis for the $2m$ -th degree even polynomials.

By taking the Fourier transform of (67), one obtains:

$$\hat{\zeta}_m(k) = \sum_{n=0}^m c_n \mathcal{F} \left(\frac{d^{2n}}{dx^{2n}} \zeta_0(x) \right) = \hat{\zeta}_0(k) \sum_{n=0}^m (-1)^n c_n k^{2n}. \quad (68)$$

Moment conditions

The moment conditions (cf., amongst others, Beale and Majda (1985); Liu et al (1995); Eldredge et al (2002)) are:

$$\int \zeta_m(y) dy = 1, \quad (69a)$$

$$\int y^p \zeta_m(y) dy = 0, \quad \forall 1 \leq p \leq r-1. \quad (69b)$$

However, thanks to the choice of even functions (cf. equation (66)), odd moment conditions are naturally respected. Moment conditions (69b) thus reduce to:

$$\int y^{2p} \zeta_m(y) dy = 0, \quad \forall 1 \leq p \leq m. \quad (70)$$

In the Fourier domain, those conditions are respectively equivalent to:

$$\hat{\zeta}_m(0) = 1, \quad (71a)$$

$$\hat{\zeta}_m^{(2p)}(0) = 0, \quad \forall 1 \leq p \leq m. \quad (71b)$$

Expression of the c_n

Using (68) evaluated at $k=0$, it follows directly from (71a) that $c_0 = 1$.

In addition, for $1 \leq p \leq m$, $\hat{\zeta}_m^{(2p)}(k)$ is given by:

$$\hat{\zeta}_m^{(2p)}(k) = \sum_{n=0}^m (-1)^n c_n \hat{\phi}_{p,n}(k), \quad \text{with } \hat{\phi}_{p,n}(k) = \frac{d^{2p}}{dk^{2p}} \left(k^{2n} \hat{\zeta}_0(k) \right). \quad (72)$$

– If $n > p$, $\hat{\phi}_{p,n}(0) = 0$ and thus

$$\hat{\zeta}_m^{(2p)}(0) = \sum_{n=0}^p (-1)^n c_n \hat{\phi}_{p,n}(0). \quad (73)$$

– If $n \leq p$, using Leibniz rule,

$$\hat{\phi}_{p,n}(k) = \sum_{j=0}^{2p} \binom{2p}{j} \frac{d^j}{dk^j} (k^{2n}) \frac{d^{2p-j}}{dk^{2p-j}} (\hat{\zeta}_0(k)), \quad (74)$$

which, evaluated at $k = 0$, reduces to

$$\hat{\phi}_{p,n}(0) = \binom{2p}{2n} (2n)! \hat{\zeta}_0^{(2(p-n))}(0) = \frac{(2p)!}{[2(p-n)]!} \hat{\zeta}_0^{(2(p-n))}(0). \quad (75)$$

Thanks to Hermite polynomials, it follows that, for any $q > 0$,

$$\hat{\zeta}_0^{2q}(0) = \left(\frac{d^{2q}}{dk^{2q}} e^{-k^2/4} \right)_{k=0} = \left(\frac{1}{2^{2q}} e^{-k^2/4} H_{2n}(k) \right)_{k=0} = \frac{(-1)^q}{2^{2q}} \frac{(2q)!}{q!} \quad (76)$$

and, combined with equations (75) and (76), equation (73) becomes

$$\hat{\zeta}_m^{(2p)}(0) = \frac{(-1)^p (2p)!}{2^{2p}} \sum_{n=0}^p \frac{2^{2n}}{(p-n)!} c_n. \quad (77)$$

Condition (71b) implies

$$\sum_{n=0}^p \frac{2^{2n}}{(p-n)!} c_n = 0, \quad \forall 1 \leq p \leq m, \quad (78)$$

which eventually leads to the following recursive relation:

$$c_p = -\frac{1}{2^{2p}} \sum_{n=0}^{p-1} \frac{2^{2n}}{(p-n)!} c_n, \quad \forall 1 \leq p \leq m. \quad (79)$$

By a simple recursive induction, it can be shown that the sequence $(c_n)_{n \geq 0}$ has the following explicit definition:

$$c_n = \frac{(-1)^n}{2^{2n} n!}, \quad \forall 0 \leq n \leq m. \quad (80)$$

Consequences on the kernel and its Fourier transform

Injecting (80) into (67) leads to

$$\zeta_m(x) = \sum_{n=0}^m \frac{(-1)^n}{2^{2n} n!} \zeta_0^{(2n)}(x), \quad (81)$$

and, using Hermite polynomials to explicitly express $\zeta_0^{(2n)}(x)$, a general analytic expression for a Gaussian kernel of any order r is obtained:

$$\zeta_m(x) = \frac{e^{-x^2}}{\sqrt{\pi}} \sum_{p=0}^m \left[(-1)^p \sum_{n=p}^m \beta_p^{[n]} \right] x^{2p}, \quad (82)$$

where

$$\beta_p^{[n]} = \frac{(2n)!}{(2p)!} \frac{1}{2^{2(n-p)} n!(n-p)!} \in \mathbb{Q}, \quad \forall p = 0, \dots, m. \quad (83)$$

This explicit expression is not very helpful as such. However, it follows from (81) that ζ_m is defined from ζ_{m-1} and the $2m$ -th derivative of ζ_0 :

$$\zeta_m(x) = \zeta_{m-1}(x) + \frac{(-1)^m}{2^{2m}m!} \zeta_0^{(2m)}(x). \quad (84)$$

Using again Hermite polynomials to explicitly express $\zeta_0^{(2m)}(x)$, the values of the $\gamma_n^{[m]} \in \mathbb{Q}$, which entirely define ζ_m , can thus be incrementally induced from the previous order kernel:

$$\begin{cases} \gamma_n^{[m]} = \gamma_n^{[m-1]} + (-1)^n \beta_n^{[m]} & \forall n = 0, \dots, m-1 \\ \gamma_m^{[m]} = (-1)^m \beta_m^{[m]} = \frac{(-1)^m}{m!}. \end{cases} \quad (85)$$

In fact, the explicit expression (82) is more interesting and useful in the Fourier domain. By injecting (80) into (68), one obtains:

$$\hat{\zeta}_m(k) = e^{-k^2/4} \sum_{n=0}^m \frac{k^{2n}}{2^{2n}n!}. \quad (86)$$

and thus

$$\hat{\zeta}_{m,\varepsilon}(k) = e^{-\varepsilon^2 k^2/4} \sum_{n=0}^m \frac{1}{n!} \left(\frac{\varepsilon}{2}\right)^{2n} k^{2n}. \quad (87)$$

Those results are very interesting in the sense that they provide explicit expressions of Gaussian kernels of any order of accuracy. In addition, their Fourier transform have an incremental explicit definition and satisfy the condition $\hat{\zeta}_m(k) \leq \hat{\zeta}_m(0)$, which is the condition required by Cortez (1997) for stability.

Particle approximation

In the sequel, the particle approximation of the Gaussian patch $c(x, t) = \exp(-x^2/(4\nu t)) / \sqrt{4\pi\nu t}$, presented in section 4, is analysed.

In an effort to simplify notations, the time dependence of some functions may be omitted in the sequel, although it should be present both in the function and the Fourier domains. Let $\sigma = \sqrt{4\nu t}$ and $\alpha = \sqrt{\varepsilon^2 + \sigma^2}$. Then

$$\hat{\zeta}_{m,\varepsilon}\hat{c}(k) = e^{-\alpha^2 k^2/4} \sum_{n=0}^m \frac{1}{n!} \left(\frac{\varepsilon}{2}\right)^{2n} k^{2n}, \quad (88)$$

and thus

$$\langle c \rangle(x) = \mathcal{F}^{-1} [\hat{\zeta}_{m,\varepsilon}\hat{c}] = \sum_{n=0}^m \frac{1}{n!} \left(\frac{\varepsilon}{2}\right)^{2n} \mathcal{F}^{-1} [k^{2n} e^{-\alpha^2 k^2/4}] \quad (89)$$

$$= \frac{1}{\alpha\sqrt{\pi}} \sum_{n=0}^m \frac{(-1)^n}{n!} \left(\frac{\varepsilon}{2}\right)^{2n} \frac{d^{2n}}{dx^{2n}} e^{-x^2/\alpha^2}. \quad (90)$$

Using Hermite polynomials,

$$\langle c \rangle(x) = \frac{1}{\alpha\sqrt{\pi}} e^{-x^2/\alpha^2} P_m(x), \quad (91)$$

with

$$P_m(x) = \sum_{p=0}^m \frac{(-1)^p}{(2p)!} \left(\sum_{n=p}^m \frac{(2n)! \varepsilon^{2n}}{2^{2(n-p)} n! (n-p)! \alpha^{2(n+p)}} \right) x^{2p}. \quad (92)$$

Likewise, since $\langle \nabla c \rangle = \mathcal{F}^{-1} [ik\hat{c}_{m,\varepsilon}\hat{c}]$ it follows that:

$$\langle \nabla c \rangle(x) = \frac{-2x}{\alpha^3 \sqrt{\pi}} e^{-x^2/\alpha^2} Q_m(x), \quad (93)$$

with

$$Q_m(x) = \sum_{p=0}^m \frac{(-1)^p}{(2p+1)!} \left(\sum_{n=p}^m \frac{(2n+1)! \varepsilon^{2n}}{2^{2(n-p)} n! (n-p)! \alpha^{2(n+p)}} \right) x^{2p}. \quad (94)$$

Consequences on the error

The analytic exact expressions of c , ∇c and u_d are given by:

$$c(x) = \frac{1}{\sigma \sqrt{\pi}} e^{-x^2/\sigma^2}, \quad (95)$$

$$\nabla c(x) = \frac{-2x}{\sigma^3 \sqrt{\pi}} e^{-x^2/\sigma^2}, \quad (96)$$

and

$$u_d(x) = \frac{x}{2t}. \quad (97)$$

It follows that

$$\frac{\langle c \rangle}{c}(x) = \frac{\sigma}{\alpha} e^{\varepsilon^2 x^2/\alpha^2} P_m(x), \quad (98)$$

$$\frac{\langle \nabla c \rangle}{\nabla c}(x) = \frac{\sigma^3}{\alpha^3} e^{\varepsilon^2 x^2/\alpha^2} Q_m(x), \quad (99)$$

and

$$\frac{\langle u_d \rangle}{u_d}(x) = \frac{\sigma^2}{\alpha^2} \frac{Q_m}{P_m}(x). \quad (100)$$

Expressing $\mathcal{E}_{\langle c \rangle}$, $\mathcal{E}_{\langle \nabla c \rangle}$ and $\mathcal{E}_{\langle u_d \rangle}$ from these last three expressions is straightforward.

It should be noted that P_m and Q_m being polynomials with degree $2m$, the error $\mathcal{E}_{\langle u_d \rangle}$ is spatially constant for the 2nd order ($r = 2$ and thus $m = 0$) approximation, at any given time $t > 0$.

Acknowledgements The authors would like to thank the Haute-Normandie Regional Council and the Institut Français de Recherche pour l'Exploitation de la Mer (IFREMER) for their financial support of co-financed PhD theses, as well as the Réseau d'Hydrodynamique Normand (RHYN0). The authors would also like to thank the Centre des Ressources Informatiques de HAute-Normandie (CRIHAN) for their available numerical computation resources.

References

- Beale JT, Majda A (1985) High order accurate vortex methods with explicit velocity kernels. *Journal of Computational Physics* 58(2):188–208, DOI 10.1016/0021-9991(85)90176-7, URL <http://www.sciencedirect.com/science/article/B6WHY-4DDR2KW-47/2/a1fa01ee4805464c4e45e0a3dededa59>
- Beaudoin A, Huberson S, Rivoalen E (2003) Simulation of anisotropic diffusion by means of a diffusion velocity method. *Journal of Computational Physics* 186(1):122 – 135, DOI 10.1016/S0021-9991(03)00024-X, URL <http://www.sciencedirect.com/science/article/B6WHY-47YPK0S-2/2/d4dea06c15c422a20f525b6f67683cd4>
- Bonet J, Lok TS (1999) Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering* 180(1–2):97 – 115, DOI 10.1016/S0045-7825(99)00051-1, URL <http://www.sciencedirect.com/science/article/pii/S0045782599000511>

- Chatelain P, Backaert S, Winckelmans G, Kern S (2013) Large eddy simulation of wind turbine wakes. *Flow, Turbulence and Combustion* 91(3):587–605, DOI 10.1007/s10494-013-9474-8, URL <http://dx.doi.org/10.1007/s10494-013-9474-8>
- Chertock A, Levy D (2001) Particle methods for dispersive equations. *Journal of Computational Physics* 171(2):708 – 730, DOI 10.1006/jcph.2001.6803, URL <http://www.sciencedirect.com/science/article/pii/S0021999101968032>
- Chertock A, Levy D (2002) A particle method for the KdV equation. *J Sci Comput* 17(1-4):491–499, DOI 10.1023/A:1015106210404, URL <http://dx.doi.org/10.1023/A:1015106210404>
- Chorin AJ (1973) Numerical study of slightly viscous flow. *Journal of Fluid Mechanics* 57(04):785–796, DOI 10.1017/S0022112073002016, URL <http://dx.doi.org/10.1017/S0022112073002016>
- Christiansen J (1997) Numerical simulation of hydrodynamics by the method of point vortices. *Journal of Computational Physics* 135(2):189 – 197, DOI <http://dx.doi.org/10.1006/jcph.1997.5701>, URL <http://www.sciencedirect.com/science/article/pii/S0021999197957016>
- Colagrossi A, Landrini M (2003) Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of Computational Physics* 191(2):448–475, DOI 10.1016/S0021-9991(03)00324-3, URL <http://www.sciencedirect.com/science/article/pii/S0021999103003243>
- Cortez R (1997) Convergence of high-order deterministic particle methods for the convection-diffusion equation. *Comm Pure Appl Math* 50(L):1235–1260
- Cottet GH, Poncet P (2004) Advances in direct numerical simulations of 3d wall-bounded flows by vortex-in-cell methods. *Journal of Computational Physics* 193(1):136–158, DOI <http://dx.doi.org/10.1016/j.jcp.2003.08.025>, URL <http://www.sciencedirect.com/science/article/pii/S0021999103004248>
- Degond P, Mas-Gallic S (1989) The weighted particle method for convection-diffusion equations. Part I: The case of an isotropic viscosity. *Math Comp* 53(188):485–507, DOI 10.2307/2008716, URL <http://dx.doi.org/10.2307/2008716>
- Degond P, Mustieles FJ (1990) A deterministic approximation of diffusion equations using particles. *SIAM Journal on Scientific and Statistical Computing* 11(2):293–310, DOI 10.1137/0911018, URL <http://link.aip.org/link/?SCE/11/293/1>
- Dynnikov Y, Dynnikova G (2011) Numerical stability and numerical viscosity in certain meshless vortex methods as applied to the navier-stokes and heat equations. *Computational Mathematics and Mathematical Physics* 51:1792–1804, DOI 10.1134/S096554251110006X, URL <http://dx.doi.org/10.1134/S096554251110006X>
- Dynnikova G (2011) Calculation of three-dimensional flows of an incompressible fluid based on a dipole representation of vorticity. *Doklady Physics* 56(3):163–166, DOI 10.1134/S1028335811030025, URL <http://dx.doi.org/10.1134/S1028335811030025>
- Eldredge JD, Leonard A, Colonius T (2002) A general deterministic treatment of derivatives in particle methods. *Journal of Computational Physics* 180(2):686–709, DOI 10.1006/jcph.2002.7112, URL <http://www.sciencedirect.com/science/article/B6WHY-46G47HY-F/2/a01a8d80f16f9d8f1275850e03158789>
- Fronteau J, Combis P (1984) A lie admissible method of integration of folkler-plank equations with non linear coefficients (exact and numerical solutions). *Hadronic J* 7:911–930
- Gambino G, Lombardo M, Sammartino M (2009) A velocity-diffusion method for a lotka-volterra system with nonlinear cross and self-diffusion. *Applied Numerical Mathematics* 59(5):1059 – 1074, DOI 10.1016/j.apnum.2008.05.002, URL <http://www.sciencedirect.com/science/article/pii/S0168927408000925>
- Grant J, Marshall J (2005) Diffusion velocity for a three-dimensional vorticity field. *Theoretical and Computational Fluid Dynamics* 19:377–390, DOI 10.1007/s00162-005-0004-8, URL <http://dx.doi.org/10.1007/s00162-005-0004-8>
- Guvernyuk S, Dynnikova G (2007) Modeling the flow past an oscillating airfoil by the method of viscous vortex domains. *Fluid Dynamics* 42:1–11, DOI 10.1134/S0015462807010012, URL <http://dx.doi.org/10.1134/S0015462807010012>
- Kempka S, Strickland J (1993) A method to simulate viscous diffusion of vorticity by convective transport of vortices at a non-solenoidal velocity. Technical Report SAND-93-1763, Sandia Laboratory, URL <http://www.osti.gov/energycitations/servlets/purl/10190654-dLAMwy/native/>
- Lacombe G (1999) Analyse d’une équation de vitesse de diffusion. *Comptes Rendus de l’Académie des Sciences - Series I - Mathe-*

- matics 329(5):383 – 386, DOI 10.1016/S0764-4442(00)88610-3, URL <http://www.sciencedirect.com/science/article/pii/S0764444200886103>
- Lacombe G, Mas-Gallic S (1999) Presentation and analysis of a diffusion-velocity method. *ESAIM: Proc* 7:225–233, DOI 10.1051/proc:1999021, URL <http://dx.doi.org/10.1051/proc:1999021>
- Leonard A (1980) Vortex methods for flow simulation. *Journal of Computational Physics* 37(3):289–335, DOI 10.1016/0021-9991(80)90040-6, URL <http://www.sciencedirect.com/science/article/B6WHY-4DD1RFV-1T/2/68d5c3a0ce2f03f137353c200b9df3e8>
- Lions PL, Mas-Gallic S (2001) Une méthode particulière déterministe pour des équations diffusives non linéaires. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics* 332(4):369–376, DOI 10.1016/S0764-4442(00)01795-X, URL <http://www.sciencedirect.com/science/article/B6VJ2-42JYXCG-J/2/7ba004247cb104c46218511bd49d8956>
- Liu WK, Jun S, Zhang YF (1995) Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 20(8-9):1081–1106, DOI 10.1002/flid.1650200824, URL <http://dx.doi.org/10.1002/flid.1650200824>
- Liu WK, Chen Y, Uras R, Chang CT (1996) Generalized multiple scale reproducing kernel particle methods. *Computer Methods in Applied Mechanics and Engineering* 139(1-4):91–157, DOI 10.1016/S0045-7825(96)01081-X, URL <http://www.sciencedirect.com/science/article/pii/S004578259601081X>
- Mansfield JR (1997) A dynamic Lagrangian Large Eddy Simulation scheme for the vorticity transport equation. PhD thesis, John Hopkins University, Baltimore, Maryland, USA
- Mansfield JR, Knio OM, Meneveau C (1996) Towards lagrangian large vortex simulation. *ESAIM: Proc* 1:49–64, DOI 10.1051/proc:1996019, URL <http://dx.doi.org/10.1051/proc:1996019>
- Mansfield JR, Knio OM, Meneveau C (1998) A dynamic LES scheme for the vorticity transport equation: Formulation and a priori tests. *Journal of Computational Physics* 145(2):693–730, DOI 10.1006/jcph.1998.6051, URL <http://www.sciencedirect.com/science/article/B6WHY-45J58W3-2N/2/ea859704c5e2194b3645e0ad3bcbff90>
- Mas-Gallic S (1999) A presentation of the diffusion velocity method. In: Leach P, Bouquet S, Rouet JL, Fijalkow E (eds) *Dynamical Systems, Plasmas and Gravitation*, Lecture Notes in Physics, vol 518, Springer Berlin / Heidelberg, pp 74–81, DOI 10.1007/BFb0105914, URL <http://dx.doi.org/10.1007/BFb0105914>
- Meneveau C, Katz J (2000) Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics* 32(1):1–32, DOI 10.1146/annurev.fluid.32.1.1, URL <http://www.annualreviews.org/doi/abs/10.1146/annurev.fluid.32.1.1>, <http://www.annualreviews.org/doi/pdf/10.1146/annurev.fluid.32.1.1>
- Milane R, Nourazar S (1995) On the turbulent diffusion velocity in mixing layer simulated using the vortex method and the subgrid scale vorticity model. *Mechanics Research Communications* 22(4):327 – 333, DOI 10.1016/0093-6413(95)00032-M, URL <http://www.sciencedirect.com/science/article/pii/009364139500032M>
- Milane R, Nourazar S (1997) Large-eddy simulation of mixing layer using vortex method: Effect of subgrid-scale models on early development. *Mechanics Research Communications* 24(2):215 – 221, DOI [http://dx.doi.org/10.1016/S0093-6413\(97\)00015-3](http://dx.doi.org/10.1016/S0093-6413(97)00015-3), URL <http://www.sciencedirect.com/science/article/pii/S0093641397000153>
- Milane RE (2004) Large eddy simulation (2d) using diffusion-velocity method and vortex-in-cell. *International Journal for Numerical Methods in Fluids* 44(8):837–860, DOI 10.1002/flid.673, URL <http://dx.doi.org/10.1002/flid.673>
- Mustieles FJ (1990) L'équation de boltzmann des semiconducteurs. Étude mathématique et simulation numérique. PhD thesis, École Polytechnique
- Mycek P, Pinon G, Germain G, Rivoalen E (2013) A self-regularising DVM-PSE method for the modelling of diffusion in particle methods. *Comptes Rendus Mécanique* 341(9-10):709–714, DOI <http://dx.doi.org/10.1016/j.crme.2013.08.002>, URL <http://www.sciencedirect.com/science/article/pii/S1631072113001137>
- Ogami Y (1999) Simulation of heat-vortex interaction by the diffusion velocity method. *ESAIM: Proc* 7:313–324, DOI 10.1051/proc:1999029, URL <http://dx.doi.org/10.1051/proc:1999029>

- Ogami Y, Akamatsu T (1991) Viscous flow simulation using the discrete vortex model—the diffusion velocity method. *Computers & Fluids* 19(3-4):433–441, DOI 10.1016/0045-7930(91)90068-S, URL <http://www.sciencedirect.com/science/article/pii/004579309190068S>
- Oger G, Doring M, Alessandrini B, Ferrant P (2007) An improved sph method: Towards higher order convergence. *Journal of Computational Physics* 225(2):1472–1492, DOI 10.1016/j.jcp.2007.01.039, URL <http://www.sciencedirect.com/science/article/pii/S0021999107000630>
- Pinon G, Mycek P, Germain G, Rivoalen E (2012) Numerical simulation of the wake of marine current turbines with a particle method. *Renewable Energy* 46(0):111 – 126, DOI 10.1016/j.renene.2012.03.037, URL <http://www.sciencedirect.com/science/article/pii/S0960148112002418>
- Rivoalen E, Huberson S (1999) Numerical simulation of axisymmetric viscous flows by means of a particle method. *Journal of Computational Physics* 152(1):1 – 31, DOI 10.1006/jcph.1999.6210, URL <http://www.sciencedirect.com/science/article/pii/S0021999199962101>
- Rivoalen E, Huberson S, Hauville F (1997) Simulation numérique des équations de Navier-Stokes 3D par une méthode particulière. *Comptes Rendus de l'Académie des Sciences - Series IIB - Mechanics-Physics-Chemistry-Astronomy* 324(9):543–549, DOI 10.1016/S1251-8069(97)83187-9, URL <http://www.sciencedirect.com/science/article/B6VJ4-3WNN24T-J/2/059096dfb12885bea81e404865dd6a63>
- Sagaut P (2006) *Large Eddy Simulation for Incompressible Flows: an Introduction*. Scientific Computation, Springer, URL <http://books.google.fr/books?id=ODYiH6RNyoQC>
- Schrader B, Reboux S, Sbalzarini IF (2010) Discretization correction of general integral pse operators for particle methods. *Journal of Computational Physics* 229(11):4159–4182, DOI 10.1016/j.jcp.2010.02.004, URL <http://www.sciencedirect.com/science/article/B6WHY-4YCWNPg-2/2/674fb3a5ba4f77a55861454096dafc5c>
- Strickland JH, Kempka SN, Wolfe WP (1996) Viscous diffusion of vorticity using the diffusion velocity concept. *ESAIM: Proc* 1:135–151, DOI 10.1051/proc:1996033, URL <http://dx.doi.org/10.1051/proc:1996033>
- Winckelmans G, Coale R, Dufresne L, Capart R (2005) Vortex methods and their application to trailing wake vortex simulations. *Comptes Rendus Physique* 6(4–5):467 – 486, DOI <http://dx.doi.org/10.1016/j.crhy.2005.05.001>, URL <http://www.sciencedirect.com/science/article/pii/S1631070505000575>