Implicit iterative particle shifting for meshless numerical schemes using kernel basis functions

Implicit iterative particle shifting for meshless numerical schemes using kernel basis functions

P. Rastelli*[a], R. Vacondio[a], J.C. Marongiu[b], G. Fourtakas[c], Benedict D. Rogers[c]

[a]*Department of Engineering and Architecture, University of Parma, Parma, Italy*
[b]*R&D Department, ANDRITZ Hydro, Vevey, Switzerland.*
[c]*Department of Mechanical, Aerospace and Civil Engineering, Faculty of Science and Engineering, The University of Manchester, Manchester, M13 9PL, UK.*

Abstract

A novel particle shifting technique (PST) for meshless numerical methods is presented. The proposed methodology uses an implicit iterative particle shifting (IIPS) technique aiming to reduce the spatial particle' anisotropy, which is associated with the discretisation error in meshless numerical schemes based on kernel basis functions. The algorithm controls the particle spatial distribution based on an implicit minimization problem, related to the particle concentration gradient and therefore, to the particles anisotropy. This results in accurate particle distributions, near to the theoretical convergence rates. To demonstrate the effectiveness of the proposed method, the IIPS algorithm is tested within a smoothed particle hydrodynamics (SPH) framework, with static and kinematic cases, by examining the particle distributions and the corresponding spatial accuracy. Further, the computational cost of the proposed methodology is reported and it is shown that it introduces minimal overhead. Moreover, the simulations of the Taylor Green vortex (TGV), employing a weakly-compressible SPH Navier-Stokes solver, confirmed the superior accuracy of the IIPS in comparison to existing explicit shifting approaches, in simulating internal flows.

Keywords: meshless discretisation schemes; implicit iterative particle shifting; particle anisotropy; error minimisation; convergence rates;

1. Introduction

In meshless numerical methods, the continuum is discretized by means of computational nodes (or particles) without any topological interconnections. Numerous meshless numerical methods have been introduced during the second half of the last century to simulate problems involving the solution of partial differential equations (PDEs). The particle-in-cell (PIC) method [12] and its extension to the so-called material point method (MPM), introduced by Sulsky [44] in 1994, is a Lagrangian method in which the domain deformations follow a background Eulerian mesh to solve the governing equations (for example [11]). Similarly, in 1996, a scattered distribution of points was adopted in the finite pointset method (FPM) by Oñate [35], where the governing equations are solved in a fully Lagrangian framework. This approach also requires a remeshing technique to be applied in order to solve issues related to large

*Corresponding author. pietro.rastelli@unipr.it (P. Rastelli)

mesh distortion. In the same year, the moving particle semi-implicit (MPS) method was introduced to simulate viscous incompressible flows by Koshizuka [22], with multiple developments in the following 20 years (e.g. [48]). The MPS scheme utilises inter-particle repulsive-attractive forces to stabilize the Lagrangian discrete computational points. Later, a meshless model called finite volume particle method (FVPM) was presented in the early 2000s [15], which includes the advantages of the Lagrangian description of motion, and the exact partition of unity in the discretization domain. In most of those numerical schemes, kernel-based spatial interpolation is adopted to discretize the differential operators that appear in the PDEs. This has been developed further for free-surface flow by Quinlan [38]. In addition to these meshless interpolation schemes, the radial basis functions (RBFs) method was also introduced in 1990 by Kansa [18] where radial functions are used to determine basis function weights for the numerical solution for both steady-state and time-dependent PDEs.

Of all meshless schemes now available, the most widespread is the Lagrangian method smoothed particle hydrodynamics. In SPH the interpolation points are called particles, which move with the field velocity during the simulation [29] (see for example the review paper [24] for a general introduction of the method). No background mesh is required and the particle interactions are computed as summations using gradients of a weighting function (called the smoothing kernel) making it ideal for potentially scattered data points or particles. As a truly meshless method, the SPH scheme is well suited to hardware acceleration and has been successfully applied to a broad range of applications including e.g. astrophysical simulations [37], [57], compressible [45], [4] and incompressible fluids [7], structural analysis of elastic and plastic solids [19], geotechnical problem and geophysics [5], [59], [60], thermal conduction [30], [16] laser welding [41], [17] metal forming [8], wave simulations [1], [10], fluid-structure interactions [54], [40] and multiphase flows [61], [13], [36].

In kernel-based Lagrangian methods such as SPH and MPS, particles follow the streamlines and therefore, they cluster or create nonphysical voids in the computational domain. It is now well known that the particle spatial distribution affects the accuracy of the spatial interpolation, e.g., [50], [9]. Since SPH uses kernel-based operators, the spatial interpolation may exhibit numerical instability for disordered particles distributions, e.g., [39], [2]. To regularize the particle distribution numerous approaches have been proposed. In the framework of FVPM, the particle shifting technique (PST) was invented by Nestor *et al.* [32], (a detailed description of the algorithm is reported in Section 2.1). A similar approach, based on particles position, was initially applied in the incompressible SPH (ISPH) scheme by Xu *et al.* [58] for interior flows, with the aim of reducing spurious oscillations in the pressure field. Later this method was extended by Lind *et al.* [23] to free-surface problems, adopting the so-called Fickian formulation, based on Fick's diffusion law, which adjusts the particle distribution accordingly to the gradient of the particle concentration. This PST was modified and applied to body-water slam simulation by Skillen *et al.* [43] . In the weakly-compressible SPH (WCSPH) schemes, an artificial particle displacement method was initially introduced by Shadloo *et al.* [42] . Since then, Vacondio *et al.* [52], [51] combined a Fickian-based shifting technique with δ -SPH [27] and an adaptive particle refinement

(APR) method, later repeated in the δ^+ SPH formulation [47]. A more general method for free-surface or multiphase flows has been recently presented by Khayyer *et al.* [21] without tuning parameters, later implemented and optimized with an iterative procedure [20]. In addition, Oger *et al.* [34] introduced a shifting technique, with a limitation in the shifting displacement, remaining quasi-lagrangian, in the context of Arbitrarily Lagrangian-Eulerian SPH (ALE-SPH) [55], ensuring mass and momentum conservation, in the same framework a shifting procedure without limitation in the shifting distance has been introduced in [33]. Recently, Antuono *et al.* [3] introduced the δ^+ SPH formulation in an ALE-SPH scheme benefiting both from the consistency in the shifting formulation and from the numerical diffusion terms explicitly expressed.

The aforementioned shifting schemes assume that particle concentration is a good indicator for the discretization error of the SPH interpolation due to particle disorder and thus, they aim to adjust the particle distribution accordingly [9]. However, the explicit nature of these techniques implies that they cannot guarantee a pre-defined and spatially homogeneous SPH discretization error. Nor do they guarantee that the discretization error is minimised, either locally or globally. The aim of the present work is to introduce an algorithm capable of minimizing the discretization error by removing the effects of particle disorder globally, overcoming the limitation of the existing shifting techniques and avoiding any computationally expensive kernel correction algorithms. An implicit and iterative particle shifting algorithm is introduced, capable of keeping the particle concentration below a pre-defined threshold in the entire domain at each time step.

The paper is organized as follows, in Section 2 the fundamental of the SPH spatial interpolation with PST are briefly explained and an ALE-SPH solver is introduced; in Section 3 an overview of particle shifting techniques is presented, before deriving the proposed IIPS algorithm. Later in Section 4, the IIPS algorithm is analyzed considering a static and kinematic numerical test, finally, a Taylor-Green flow with Reynolds number 100 has been reproduced by means of an ALE-SPH numerical scheme in order to check the effectiveness of the proposed algorithm in improving the accuracy of the results in simulating a viscous flow. The conclusions and the future perspective of this work are finally reported in Section 5.

2. Numerical method

In meshless spatial interpolation techniques, in a generic point of the domain Ω, defined by the position vector $\mathbf{x}$, a generic scalar function $f(\mathbf{x})\colon \mathbb{R}^d \to \mathbb{R}$, with d the number of spatial components, is usually approximated by means of convolution over the domain,

$$f(\mathbf{x}) = \int_\Omega f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}',h)d\Omega, \tag{1}$$

where $W(\mathbf{x}-\mathbf{x}',h)$ is a weighting function [56], [25] with the support proportional to the characteristic length h. In numerical schemes, the integral of Equation (1) can be approximated by the following

discretization expression,

$$\langle f(\mathbf{x}_i)\rangle = \sum_{j=1}^{J} f(\mathbf{x}_j) W(\mathbf{x}_i - \mathbf{x}_j, h)\omega_j, \tag{2}$$

where i represents the calculation point and j denotes a neighbouring interpolation point with the associated discrete volume ω_j and J is the total number of particles within the support of particle i. If the weighting function has certain properties, such in [25], $\nabla f(\mathbf{x})$ can be also approximated at the continuous level as follows,

$$\nabla f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\nabla W(\mathbf{x} - \mathbf{x}', h)\, d\Omega. \tag{3}$$

The gradient approximation of Equation (3) is as accurate as the kernel at continuous level. A discrete approximation of Equation (3) can be also constructed leading to

$$\langle \nabla f(\mathbf{x}_i)\rangle = \sum_{j=1}^{J} f(\mathbf{x}_j) \nabla W(\mathbf{x}_i - \mathbf{x}_j, h)\omega_j. \tag{4}$$

Quinlan *et al.* [39] showed that Equation (4) can reach the theoretical convergence rate of 2 only if the smoothing error is much larger than the discretisation error, conversely the second-order convergence rate is no longer reached with significant particle disorder.
To demonstrate this property, a convergence analysis of $\partial_x f$ has been performed for the function $f(x, y) = sin(2\pi x/\lambda) + cos(2\pi y/\lambda)$ using Equation (4) and adopting the Wendland C6 Kernel function [56],

$$W(q) = \begin{cases} C\left(1 - \frac{q}{2}\right)^8 (4q^3 + 6.25q^2 + 4q + 1) & if \quad q \leq 2 \\ 0 & if \quad q > 2 \end{cases} \tag{5}$$

where $q = d/h$ is the ratio between the particles distance and the smoothing length and $C = 78/(28\pi h)$ is the renormalization constant in 2D.

The analysis has been conducted with different values of h/Δ, where Δ is the particle spacing, using an initial perturbed particle distribution obtained applying σ/Δ, where σ is the normalized standard deviation of this perturbation from the original Cartesian grid. Figure 1 shows the L_2 norm of the $\partial_x f(x, y)$ error against the particle spacing Δ. With the adopted particle distribution (which corresponds to $\sigma/\Delta = 0.1$), the error remains constant and does not decrease with Δ, whereas, by using a Cartesian distribution the theoretical convergence rate of 2 is obtained. Since in meshless schemes such as SPH [28] the particles move accordingly to the Lagrangian trajectories, the particle distribution cannot be controlled during the simulation which leads to poor accuracy for the spatial interpolation.

2.1. Particle Shifting Technique

In order to reduce the disorder of the particle distribution, many authors have introduced a correction to the pure Lagrangian motion of the particles. In particular, different particle shifting techniques (PSTs) have been widely studied and introduced in different SPH numerical schemes ([58], [23], [52], [34], [49], [46]). Generally, PSTs are formulated as

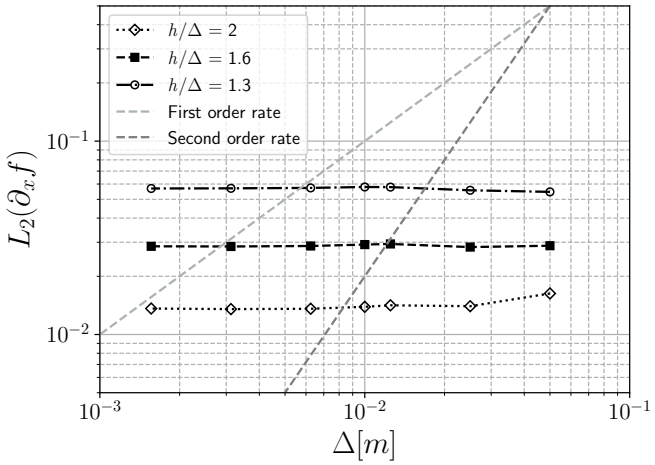


Figure 1: Numerical test. L_2 error norm in SPH estimates of the first partial derivative for a function $f(x, y) = sin(2\pi x/\lambda) + cos(2\pi y/\lambda)$, for a perturbed particle spacing $\sigma/\Delta = 0.1$

$$\bar{\mathbf{x}}_i = \mathbf{x}_i + \boldsymbol{\delta}_i, \tag{6}$$

where $\mathbf{x}_i$ is the particle position, $\bar{\mathbf{x}}_i$ is the adjusted particle position and $\boldsymbol{\delta}_i$ is the shifting vector for the ith particle. Different formulations have been proposed to compute $\boldsymbol{\delta}_i$, ([58], [23], [34]), currently, the state-of-the-art of particle shifting technique is the so-called the Fickian approach [23] where the $\boldsymbol{\delta}_i$ is proportional to the gradient of the particle concentration ∇C_i,

$$\boldsymbol{\delta}_i \propto \nabla C_i = \sum_{j=1}^{J} \nabla W_{ij} \omega_j. \tag{7}$$

∇C_i can be physically seen as a measure of the non-uniformity of particle distribution at particle i [9], it is the gradient operator of SPH and it shows the accuracy of the SPH interpolation. Large variations of the particle concentration, which can be due to particles not being homogeneously distributed and the presence of clusters or voids, are numerically translated by high values of ∇C_i. Theoretically, the value of ∇C_i should be ideally zero to minimise the error for a set of SPH particles with uniform resolution and uniform mass. A PST adjusts the particle position using Equation (6) aiming to reduce the ∇C_i, whihc corresponds to increase the accuracy of spatial interpolation. However, explicit shifting methods such as the ones adopted in [58], [23], [34] do not guarantee that the correction, applied to the particle pure Lagrangian motion, is able to keep the particle disorder below a pre-defined threshold and do not guarantee the minimisation of the error by enforcing a specific particle distribution. Moreover, when a classical Lagrangian SPH scheme is adopted, the magnitude of $\boldsymbol{\delta}$ for all particles in the domain has to be smaller than the particle displacement which is related to the Lagrangian velocity and prevents non-physical effects generated by this correction.

2.2. Arbitrary Lagrangian-Eulerian (ALE) SPH numerical solver

The Lagrangian form of the Navier-Stokes equations have the following expression:

$$\begin{aligned} \frac{d\rho}{dt} &= -\rho\nabla\boldsymbol{v} \\ \frac{d\boldsymbol{v}}{dt} &= +\boldsymbol{g} - \frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{v} \end{aligned} \tag{8}$$

where ρ is the fluid density, $\boldsymbol{v}$ is the fluid velocity, $\boldsymbol{g}$ is the gravity force, p is the pressure and ν is the kinematic viscosity. Equations (8) represent the compressible form and a barotropic equation of state needs to be introduced as follow,

$$p = \frac{c_0^2\rho_0}{\gamma}\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{9}$$

in which c_0 is the reference speed of sound and ρ_0 is the reference density, for the equation of state γ is equal to 7. The classical SPH approximation for Equations (8) have been derived by [14]. For the purpose of this work, an ALE-SPH numerical solver for Equations (8) is adopted [55], where Equations (8) are approximated as follows:

$$\begin{aligned} \frac{d\boldsymbol{x}_i}{dt} &= \boldsymbol{v}_{0i} \\ \frac{d\omega_i}{dt} &= \omega_i \sum_{j=1}^{J} \left(\boldsymbol{v}_{0j} - \boldsymbol{v}_{0i}\right)\nabla W_{ij}\omega_j \\ \frac{d(\omega_i\rho_i)}{dt} &= -\omega_i \sum_{j=1}^{J} \left(\rho_i\left(\boldsymbol{v}_i - \boldsymbol{v}_{0i}\right) + \rho_j\left(\boldsymbol{v}_j - \boldsymbol{v}_{0j}\right)\right)\nabla W_{ij}\omega_j \\ \frac{d(\omega_i\rho_i\boldsymbol{v}_i)}{dt} &= -\omega_i \sum_{j=1}^{J} \left(\rho_i\boldsymbol{v}_i \otimes \left(\boldsymbol{v}_i - \boldsymbol{v}_{0i}\right) + \rho_j\boldsymbol{v}_j \otimes \left(\boldsymbol{v}_j - \boldsymbol{v}_{0j}\right) + \boldsymbol{p}_i + \boldsymbol{p}_j\right)\nabla W_{ij}\omega_j + \omega_i\rho_i\boldsymbol{g} \end{aligned} \tag{10}$$

where $\boldsymbol{v}_0$ is the transport velocity, which is explicitly involved in the convective fluxes computation. Differently from the original weakly-compressible formulation in the ALE-SPH schemes the computational points move with the transport velocity $\boldsymbol{v}_0$ which can be arbitrarily defined and therefore particles do not have to follow the Lagrangian trajectories.
Specifically, in this work the TGV test case, presented in Section 4.3, is solved using the ASPHODEL numerical scheme [26] with MUSCL reconstruction at the particle interface and third-order Runge-Kutta time integration scheme.

3. Implicit iterative particle shifting technique

In this work a novel shifting method is presented aiming at solving the issues of the classical explicit Fickian shifting techniques. The corrected particle positions are computed adopting an implicit iterative scheme to obtain a constant and spatially homogeneous concentration in the particle distribution. In the present section the novel shifting method is derived in 1D and 2D.

3.1. 1D formulation

The problem can be described starting by defining a generic function in 1D $f(\mathbf{X})$: $\mathbb{R}^n \rightarrow \mathbb{R}$ where $\mathbf{X} = (x_1, ..., x_i, ..., x_n)$ represents the vector of particles position and n the number of particles in the domain. The objective is to identify the new particle's position array $\bar{\mathbf{X}} = (\bar{x}_1, ..., \bar{x}_i, ..., \bar{x}_n)$ in which the following conditions hold:

$$f_i(\bar{\mathbf{X}}) = 0, \ i = 1, ..., n, \tag{11}$$

where $f_i(\bar{\mathbf{X}})$ is the value of the scalar function f at particle i.
To find the roots of Equation (11) a Newton-Raphson procedure is adopted, therefore the Taylor expansion truncated at the first order of function f at position $\bar{x}_i$ has been considered,

$$f_i(\bar{\mathbf{X}}) = f_i(\mathbf{X}) + \sum_{j=1}^{J} \frac{\partial f_i(\mathbf{X})}{\partial x_j}(\bar{x}_j - x_j) + \mathcal{O}(\bar{x}_i - x_i)^2. \tag{12}$$

Here it has been assumed that the function f corresponds to the derivative of the particle concentration,

$$f(\mathbf{X}) = \frac{\partial C(\mathbf{X})}{\partial x}, \tag{13}$$

therefore,

$$\frac{\partial f(\mathbf{X})}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\partial C(\mathbf{X})}{\partial x}\right), \tag{14}$$

substituting Equations (13, 14) in Equation (12) leads to

$$\frac{\partial C_i(\bar{\mathbf{X}})}{\partial x} = \frac{\partial C_i(\mathbf{X})}{\partial x} + \sum_{j=1}^{J} \frac{\partial}{\partial x_j}\left(\frac{\partial C_i(\mathbf{X})}{\partial x}\right)(\bar{x}_j - x_j) + \mathcal{O}(\bar{x}_i - x_i)^2. \tag{15}$$

If the SPH approximation of the derivative of particle concentration C is

$$\frac{\partial C_i(\mathbf{X})}{\partial x} = \sum_{k=1}^{K} \frac{\partial W(x_i - x_k)}{\partial x_k}\omega_k, \tag{16}$$

where K is the number of neighbouring particles inside the kernel support of particle i, is adopted then,

$$\frac{\partial}{\partial x_j}\left(\frac{\partial C(\mathbf{X})}{\partial x}\right) = \sum_{k=1}^{K} \frac{\partial}{\partial x_j}\left(\frac{\partial W(x_i - x_k)}{\partial x_k}\right)\omega_k. \tag{17}$$

The only term in Equation (17) which is non-null is the one in which $j = k$, therefore, it can be re-written as,

$$\frac{\partial}{\partial x_j}\left(\frac{\partial C_i(\mathbf{X})}{\partial x}\right) = \frac{\partial^2 W\left(x_i - x_j\right)}{\partial x_j^2}\omega_j. \tag{18}$$

Substituting Equation (18) in Equation (15) leads to

$$\frac{\partial C_i(\bar{\mathbf{X}})}{\partial x} = \frac{\partial C_i(\mathbf{X})}{\partial x} + \sum_{j=1}^{J} \frac{\partial^2 W(x_i - x_j)}{\partial x_j^2} \omega_j (\bar{x}_j - x_j) + \mathcal{O}(\bar{x}_i - x_i)^2. \tag{19}$$

As explained before, in order to improve the accuracy of the SPH operator at the discrete level, the particle concentration C has to be uniform and thus its derivative should be equal to zero. Imposing this constraint in Equation (19), and neglecting non-linear terms, leads to the following equation for the generic particle i,

$$\sum_{j=1}^{J} \frac{\partial^2 W(x_i - x_j)}{\partial x_j^2} \omega_j (\bar{x}_j - x_j) = \frac{\partial C_i(\mathbf{X})}{\partial x}. \tag{20}$$

In Equation (20) the terms on the left-hand side $(\bar{x}_j - x_j)$ are the unknowns which correspond to the particle shifting δ_j for $j = 1, ..., n$, which have to be applied to all particles respectively to obtain a uniform particle concentration. This leads to a linear system of equations which can be expressed in matrix form (where $W(x_i - x_j)$ is replaced with W_{ij} for brevity) as follows:

$$\underbrace{\begin{bmatrix} \frac{\partial^2 W_{11}}{\partial x_1^2}\omega_1 & \dots & \frac{\partial^2 W_{1i}}{\partial x_i^2}\omega_i & \dots & \frac{\partial^2 W_{1n}}{\partial x_n^2}\omega_n \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^2 W_{i1}}{\partial x_1^2}\omega_1 & \dots & \frac{\partial^2 W_{ii}}{\partial x_i^2}\omega_i & \dots & \frac{\partial^2 W_{in}}{\partial x_n^2}\omega_n \\ \vdots & & \vdots & & \vdots \\ \frac{\partial^2 W_{n1}}{\partial x_1^2}\omega_1 & \dots & \frac{\partial^2 W_{ni}}{\partial x_i^2}\omega_i & \dots & \frac{\partial^2 W_{nn}}{\partial x_n^2}\omega_n \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} (\bar{x}_1 - x_1) \\ \vdots \\ (\bar{x}_i - x_i) \\ \vdots \\ (\bar{x}_n - x_n) \end{bmatrix}}_{\mathbf{X}} \approx \underbrace{\begin{bmatrix} \frac{\partial C(x_1)}{\partial x} \\ \vdots \\ \frac{\partial C(x_i)}{\partial x} \\ \vdots \\ \frac{\partial C(x_n)}{\partial x} \end{bmatrix}}_{\mathbf{B}} \tag{21}$$

By solving the linear system of Equation (21), the new particle positions $\bar{\mathbf{X}} = (\bar{x}_1, ..., \bar{x}_i, ..., \bar{x}_n)$ are found.

Note that in Equation (12) the problem has been linearised by neglecting high-order terms in the Taylor series, this effectively corresponds to a Newton-Raphson algorithm to find the solution of Equation (19) and thus, an iterative approach is necessary in order to obtain the particle distribution which fulfils the condition $\partial C_i/\partial x$=0 for $i = 1, ..., n$.

3.2. 2D formulation

Similarly, a generic function is defined in 2D $f(\mathbf{X}, \mathbf{Y})$: $\mathbb{R}^{2n} \rightarrow \mathbb{R}$ with $\mathbf{X} = (x_1, ..., x_i, ..., x_n)$ and $\mathbf{Y} = (y_1, ..., y_i, ..., y_n)$. As previously explained for the 1D case, the objective is to obtain the updated particle distribution represented by $\bar{\mathbf{X}} = (\bar{x}_1, ..., \bar{x}_i, ..., \bar{x}_n)$ and $\bar{\mathbf{Y}} = (\bar{y}_1, ..., \bar{y}_i, ..., \bar{y}_n)$ in which

$$f_i(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) = 0, i = 1, ..., n. \tag{22}$$

Recalling Equation (12), in two dimensions, the Taylor series expansion truncated at the first-order of function f at position $\bar{x}_i$, $\bar{y}_i$ is

$$f_i(\bar{\mathbf{X}}, \bar{\mathbf{Y}}) = f_i(\mathbf{X}, \mathbf{Y}) + \sum_{j=1}^{J} \frac{\partial f_i(\mathbf{X}, \mathbf{Y})}{\partial x_j}(\bar{x}_j - x_j) + \sum_{j=1}^{J} \frac{\partial f_i(\mathbf{X}, \mathbf{Y})}{\partial y_j}(\bar{y}_j - y_j) + \mathcal{O}((\bar{x}_i - x_i)(\bar{y}_i - y_i))^2. \quad (23)$$

For the two dimensional case, two different functions can express the gradient of particle concentration components along the two axes ,

$$f^{(1)}(\mathbf{X}, \mathbf{Y}) = \frac{\partial C(\mathbf{X}, \mathbf{Y})}{\partial x}, \quad (24)$$

$$f^{(2)}(\mathbf{X}, \mathbf{Y}) = \frac{\partial C(\mathbf{X}, \mathbf{Y})}{\partial y}. \quad (25)$$

Therefore, Equation (24, 25) can be substituted separately in Equation (23), leading to two separate equations for each spatial component,

$$\begin{aligned} \frac{\partial C_i(\bar{\mathbf{X}}, \bar{\mathbf{Y}})}{\partial x} = \frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial x} &+ \underbrace{\sum_{j=1}^{J} \frac{\partial}{\partial x_j}\left[\frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial x}\right]}_{second \quad derivative}(\bar{x}_i - x_i) \\ &+ \underbrace{\sum_{j=1}^{J} \frac{\partial}{\partial y_j}\left[\frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial x}\right]}_{cross \quad derivative}(\bar{y}_i - y_i) + \mathcal{O}((\bar{x}_i - x_i)(\bar{y}_i - y_i))^2, \end{aligned} \quad (26)$$

along the x axis, and

$$\begin{aligned} \frac{\partial C_i(\bar{\mathbf{X}}, \bar{\mathbf{Y}})}{\partial y} = \frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial y} &+ \underbrace{\sum_{j=1}^{J} \frac{\partial}{\partial x_j}\left[\frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial y}\right]}_{cross \quad derivative}(\bar{x}_i - x_i) \\ &+ \underbrace{\sum_{j=1}^{J} \frac{\partial}{\partial y_j}\left[\frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial y}\right]}_{second \quad derivative}(\bar{y}_i - y_i) + \mathcal{O}((\bar{x}_i - x_i)(\bar{y}_i - y_i))^2, \end{aligned} \quad (27)$$

along the y axis.

The SPH approximation for ∇C components in 2D are

$$\frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial x} = \sum_{k=1}^{K} \frac{\partial W\left((x_i - x_k), (y_i - y_k)\right)}{\partial x_k}\omega_k, \quad (28)$$

$$\frac{\partial C_i(\mathbf{X}, \mathbf{Y})}{\partial y} = \sum_{k=1}^{K} \frac{\partial W\left((x_i - x_k), (y_i - y_k)\right)}{\partial y_k}\omega_k. \quad (29)$$

The second and the cross derivatives of Equations (26, 27) can be manipulated as previously illustrated in Equation (17). For the x axis, introducing equation (28) in the first order term of Equation (26),

$$\frac{\partial}{\partial x_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial x}\right) = \sum_{k=1}^{K} \frac{\partial}{\partial x_j}\left(\frac{\partial W\left((x_i - x_k),(y_i - y_k)\right)}{\partial x_k}\right)\omega_k, \tag{30}$$

$$\frac{\partial}{\partial y_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial x}\right) = \sum_{k=1}^{K} \frac{\partial}{\partial y_j}\left(\frac{\partial W\left((x_i - x_k),(y_i - y_k)\right)}{\partial x_k}\right)\omega_k, \tag{31}$$

and for the y axis, introducing Equation (29) in the first order term of Equation (27),

$$\frac{\partial}{\partial x_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial y}\right) = \sum_{k=1}^{K} \frac{\partial}{\partial x_j}\left(\frac{\partial W\left((x_i - x_k),(y_i - y_k)\right)}{\partial y_k}\right)\omega_k, \tag{32}$$

$$\frac{\partial}{\partial y_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial y}\right) = \sum_{k=1}^{K} \frac{\partial}{\partial y_j}\left(\frac{\partial W\left((x_i - x_k),(y_i - y_k)\right)}{\partial y_k}\right)\omega_k. \tag{33}$$

On the RHS of Equations (30-33), the only non-null terms of the sums are the ones where $j = k$. Therefore Equations (30-33) can be written as

$$\frac{\partial}{\partial x_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial x}\right) = \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial x_j^2}\omega_j, \tag{34}$$

$$\frac{\partial}{\partial y_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial x}\right) = \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial y_j \partial x_j}\omega_j, \tag{35}$$

along the x axis,

$$\frac{\partial}{\partial x_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial y}\right) = \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial x_j \partial y_j}\omega_j, \tag{36}$$

$$\frac{\partial}{\partial y_j}\left(\frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial y}\right) = \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial y_j^2}\omega_j, \tag{37}$$

along the y axis.

Substituting Equations (34-37) in Equations (26, 27) leads to

$$\begin{aligned}\frac{\partial C_i(\bar{\mathbf{X}},\bar{\mathbf{Y}})}{\partial x} = \frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial x} &+ \sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial x_j^2}\omega_j(\bar{x}_j - x_j) \\ &+ \sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial x_j \partial y_j}\omega_j(\bar{y}_j - y_j) + \mathcal{O}((\bar{x}_i - x_i)(\bar{y}_i - y_i))^2,\end{aligned} \tag{38}$$

for the x axis,

$$\frac{\partial C_i(\bar{\mathbf{X}},\bar{\mathbf{Y}})}{\partial y} = \frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial y} + \sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial y_j \partial x_j}\omega_j(\bar{x}_j - x_j) + \sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_k),(y_i - y_k)\right)}{\partial y_j^2}\omega_j(\bar{y}_j - y_j) + \mathcal{O}((\bar{x}_i - x_i)(\bar{y}_i - y_i))^2, \tag{39}$$

for the y axis.

As explained previously, in order to improve the accuracy of the SPH operator at the discrete level, the particle concentration C has to be uniform and therefore each component of its derivatives should be equal to zero. In 2D in order to satisfy Equation (22) and by neglecting non-linear terms, two linear equations have to be solved to for the generic particle i,

$$\sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_j),(y_i - y_k)\right)}{\partial x_j^2}\omega_j\left(\bar{x}_j - x_j\right) + \sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_j),(y_i - y_k)\right)}{\partial x_j \partial y_j}\omega_j\left(\bar{y}_j - y_j\right) = \frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial x}, \tag{40}$$

and

$$\sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_j),(y_i - y_k)\right)}{\partial y_j \partial x_j}\omega_j\left(\bar{x}_j - x_j\right) + \sum_{j=1}^{J} \frac{\partial^2 W\left((x_i - x_j),(y_i - y_k)\right)}{\partial y_j^2}\omega_j\left(\bar{y}_j - y_j\right) = \frac{\partial C_i(\mathbf{X},\mathbf{Y})}{\partial y}. \tag{41}$$

The notation $W((x_i - x_j),(y_i - y_j))$ is shortened to W_{ij}, the 2D system of linear equations in a matrix form is:

$$\underbrace{\begin{bmatrix}
\frac{\partial^2 W_{11}}{\partial x_1^2}\omega_1 & \frac{\partial^2 W_{11}}{\partial x_1 \partial y_1}\omega_1 & \dots & \frac{\partial^2 W_{1i}}{\partial x_i^2}\omega_i & \frac{\partial^2 W_{1i}}{\partial x_i \partial y_i}\omega_i & \dots & \frac{\partial^2 W_{1n}}{\partial x_n^2}\omega_n & \frac{\partial^2 W_{1n}}{\partial x_n \partial y_n}\omega_n \\
\frac{\partial^2 W_{11}}{\partial y_1 \partial x_1}\omega_1 & \frac{\partial^2 W_{11}}{\partial y_1^2}\omega_1 & \dots & \frac{\partial^2 W_{1i}}{\partial y_i \partial x_i}\omega_i & \frac{\partial^2 W_{1i}}{\partial y_i^2}\omega_i & \dots & \frac{\partial^2 W_{1n}}{\partial y_n \partial x_n}\omega_n & \frac{\partial^2 W_{1n}}{\partial y_n^2}\omega_n \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial^2 W_{i1}}{\partial x_1^2}\omega_1 & \frac{\partial^2 W_{i1}}{\partial x_1 \partial y_1}\omega_1 & \dots & \frac{\partial^2 W_{ii}}{\partial x_i^2}\omega_i & \frac{\partial^2 W_{ii}}{\partial x_i \partial y_i}\omega_i & \dots & \frac{\partial^2 W_{in}}{\partial x_i^2}\omega_n & \frac{\partial^2 W_{in}}{\partial x_n \partial y_n}\omega_n \\
\frac{\partial^2 W_{i1}}{\partial y_1 \partial x_1}\omega_1 & \frac{\partial^2 W_{i1}}{\partial y_1^2}\omega_1 & \dots & \frac{\partial^2 W_{ii}}{\partial y_i \partial x_i}\omega_i & \frac{\partial^2 W_{ii}}{\partial y_i^2}\omega_i & \dots & \frac{\partial^2 W_{in}}{\partial y_n \partial x_n}\omega_n & \frac{\partial^2 W_{in}}{\partial y_n^2}\omega_n \\
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial^2 W_{n1}}{\partial x_1^2}\omega_1 & \frac{\partial^2 W_{n1}}{\partial x_1 \partial y_1}\omega_1 & \dots & \frac{\partial^2 W_{ni}}{\partial x_i^2}\omega_i & \frac{\partial^2 W_{ni}}{\partial x_i \partial y_i}\omega_i & \dots & \frac{\partial^2 W_{nn}}{\partial x_i^2}\omega_n & \frac{\partial^2 W_{nn}}{\partial x_n \partial y_n}\omega_n \\
\frac{\partial^2 W_{n1}}{\partial y_1 \partial x_1}\omega_1 & \frac{\partial^2 W_{n1}}{\partial y_1^2}\omega_1 & \dots & \frac{\partial^2 W_{ni}}{\partial y_i \partial x_i}\omega_i & \frac{\partial^2 W_{ni}}{\partial y_i^2}\omega_i & \dots & \frac{\partial^2 W_{nn}}{\partial y_n \partial x_n}\omega_n & \frac{\partial^2 W_{nn}}{\partial y_n^2}\omega_n
\end{bmatrix}}_{\mathbf{A}}
\underbrace{\begin{bmatrix}
(\bar{x}_1 - x_1) \\
(\bar{y}_1 - y_1) \\
\vdots \\
(\bar{x}_i - x_i) \\
(\bar{y}_i - y_i) \\
\vdots \\
(\bar{x}_n - x_n) \\
(\bar{y}_n - y_n)
\end{bmatrix}}_{\mathbf{X}}
\approx
\underbrace{\begin{bmatrix}
\frac{\partial C(x_1,y_1)}{\partial x} \\
\frac{\partial C(x_1,y_1)}{\partial y} \\
\vdots \\
\frac{\partial C(x_i,y_i)}{\partial x} \\
\frac{\partial C(x_i,y_i)}{\partial y} \\
\vdots \\
\frac{\partial C(x_n,y_n)}{\partial x} \\
\frac{\partial C(x_n,y_n)}{\partial y}
\end{bmatrix}}_{\mathbf{B}} \tag{42}$$

The extension to 3D can be done in a similar way, simply adding a third equation for each particle relative to the third dimension, and also considering the cross derivatives in the matrix $\mathbf{A}$. Generally the size of $\mathbf{A}$ is $d \cdot n \times d \cdot n$, therefore, in 3D, it becomes $3n \times 3n$.

3.3. Other properties

Due to the nature of the SPH interpolation, the matrix **A** defined in Equations (21, 42), is sparse and it is convenient to adopt an iterative solver for the linear system, similar to the one used for the Poisson equation of pressure in incompressible SPH schemes [58],[23] [6]. In the present work the Jacobi preconditioner and the BiCGStab have been utilized and the Wendland C6 kernel has been used in all SPH interpolations as it has been found empirically that with this kernel the adopted iterative solver converges faster. In the present formulation two different levels of iterations are present, the external one is related to the Newton-Raphson algorithm adopted to solve the non-linear Equations (12, 23), whereas internal iterations are referred to the linear system solver, Equation (21, 42) in 1D and 2D.

4. Numerical tests

In order to validate the proposed method two different numerical experiments are initially presented in this section. A static case is used to analyze the robustness of the IIPS algorithm for different SPH properties (such as the smoothing length h, the interparticle distance Δ and initial particle disorder). Later, the algorithm behaviour has been evaluated in a second test that involves a continuous distortion of the particle distribution (disorder is continuously injected) by imposing the kinematics that reproduce counter-rotating vortexes in the domain, through the entire simulation. This flow (that resembles the Taylor-Green vortices motion) has been chosen because it is known to be very demanding in maintaining uniformity in the distribution: if particles follow the exact Lagrangian trajectories their distribution becomes distorted [34], [58]. In both tests no physical quantities are attached to the particles and only the accuracy of the SPH approximation is evaluated by means of a test function.
In this way, since the main area of interest is the accuracy of the discrete operators, the effectiveness of the proposed iterative shifting formulation can be verified regardless of the properties of the specific meshless solver adopted. The non-dimensional L_2 and L_∞ norms of the ∇C error have been employed as a measure of particle disorder,

$$L_2\left(\nabla C\right) = h\sqrt{\sum_{i=1}^{n}\frac{\left\|\nabla C_i\right\|^2}{n}}, \tag{43}$$

and

$$L_\infty\left(\nabla C\right) = h(\max_i\left|\nabla C_i\right|). \tag{44}$$

In both test cases a comparison (in term of efficiency and accuracy) with the explicit shifting formulation [49] has also been conducted. The accuracy of the spatial SPH interpolation has been evaluated using the test function

$$f(x,y) = sin\Big(\frac{\pi x}{\lambda}\Big) + cos\Big(\frac{\pi y}{\lambda}\Big), \tag{45}$$

where λ can be changed to modify the steepness. For the test function, the error has been evaluated with the reference norm

$$L_2\left(\partial_x f\right) = \sqrt{\sum_{i=1}^{n} \frac{\left\|\partial_x f_i^{SPH} - \partial_x f_i^{an}\right\|^2}{n}}, \tag{46}$$

where $\partial_x f_i^{an}$ and $\partial_x f_i^{SPH}$ are respectively the analytical and the SPH-approximated value of function f gradient component evaluated at particle i. Later, a simulation of a 2D Taylor-Green vortices test case with Re=100 has been performed by means of the ALE-SPH scheme described in Section 2.2, where the capability of the IIPS algorithm in controlling the particle distribution has been compared against the explicit non-iterative Fickian shifting, [23].

4.1. Static test case

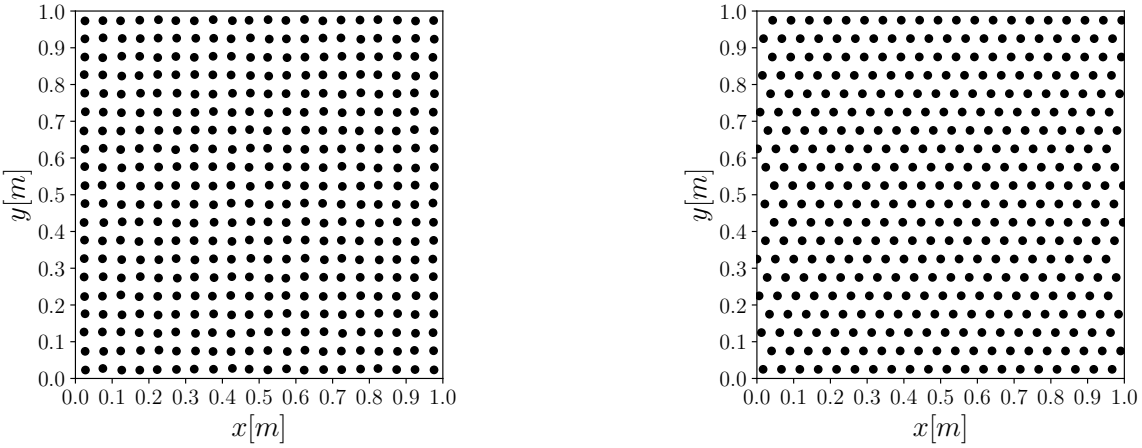
The aim of this numerical experiment is to assess how effective and efficient the IIPS formulation is in restoring a uniform particle distribution starting form a pseudo-random one. For this purpose a bi-periodic squared domain has been initialized with particles placed on a Cartesian grid, then a pseudo-random normalized perturbation with standard deviation σ has been assigned to the particles position and the IIPS algorithm has been activated. With the aim of investigating the performance of the proposed formulation, the IIPS procedure has been run for 100 Newton-Raphson (NR) iterations it_{Max}, as described in the pseudo-code (Algorithm 1). At the end of each iteration, the particle positions are updated using the results obtained solving the linear system.

Algorithm 1 Static case

```
procedure IMPLICIT ITERATIVE SHIFTING                    ▷
    CartesianGridDistribution()
    RandomPerturbation σ
    Compute ∇C_i, L_2(∇C), L_2(∂_x f)
    while it ≤ it_Max do
        AssembleMatrix(∇C_i)
        LinearSystemSolver(AssembleMatrix)
        UpdateParticlePosition(Implicit PST)
        Compute L_2(∇C), L_2(∂_x f)
    end while
end procedure
```

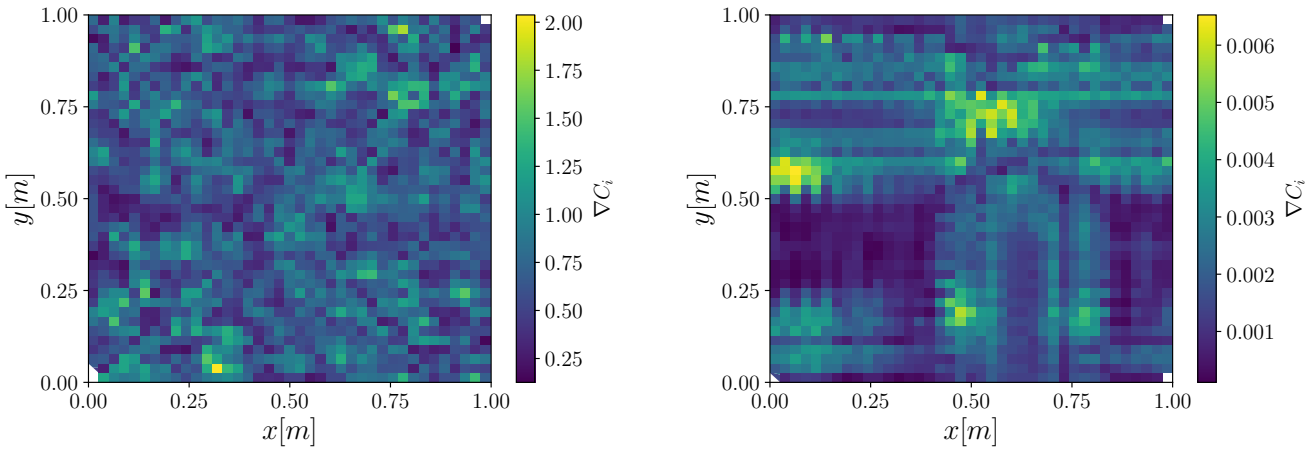
Figure 2 shows the particle distribution before (Fig. 2(a)) and after (Fig. 2(b)) the IIPS procedure for a resolution $\Delta/L = 0.05$ where L is 1 m, it can be immediately noted how the particle stencil changes, particles arrange themselves into a triangular-staggered configuration. This distribution is reached for the different resolutions ($\Delta/L = 0.025, 0.0125$) presented. It is notable that the final particle configuration is very similar to a hexagonal-centred distribution which represents a minimum extreme for ∇C in the particle system configuration as demonstrated in [9].

Figure 3 shows the maps of ∇C magnitude at the beginning (Fig. 3(a)) and at the end (Fig. 3(b)) of the simulation, it can be seen that the magnitude is reduced in the whole domain by more than two orders of magnitude.



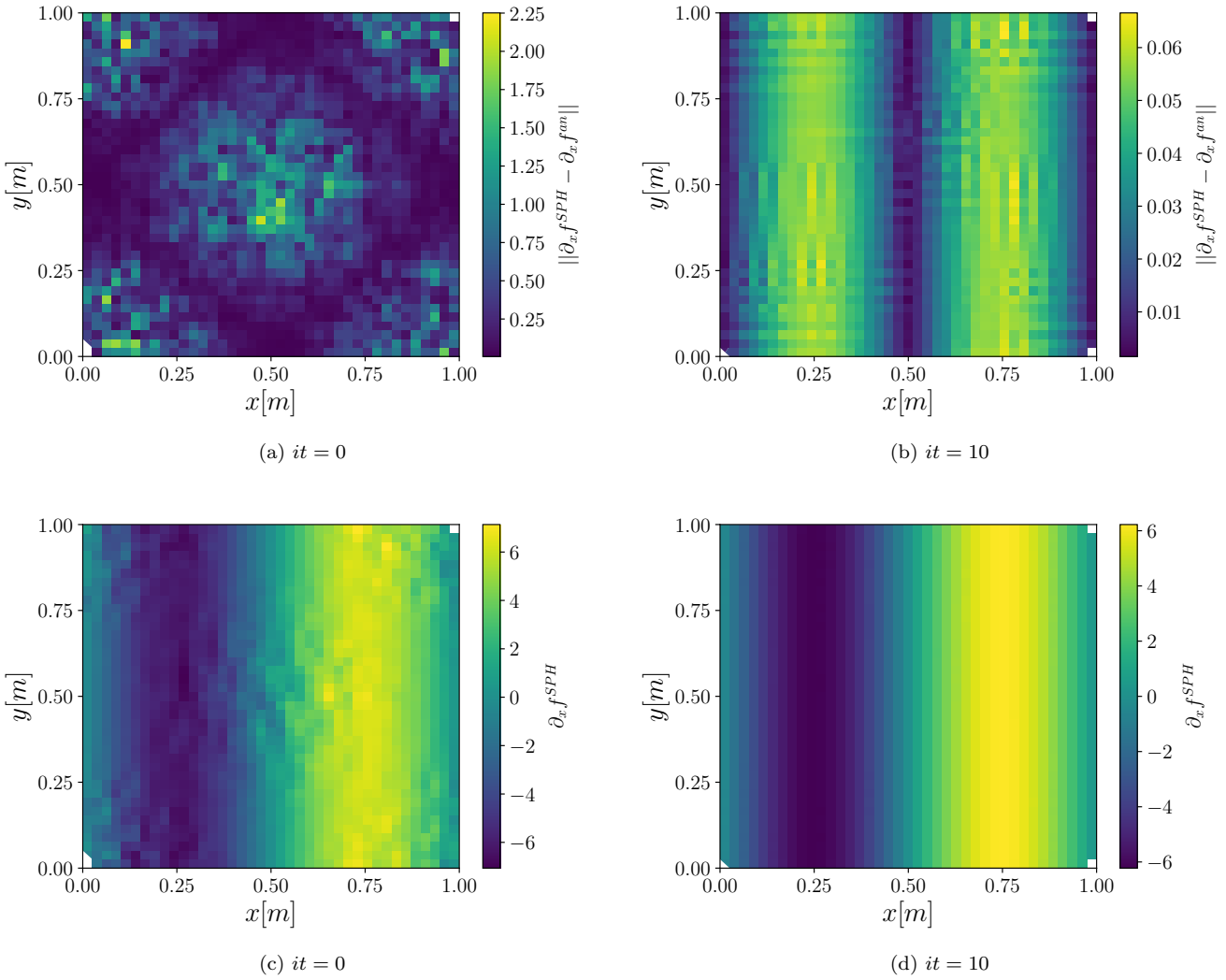
(a) (b)

Figure 2: Static test case. Particle distribution $\Delta/L = 0.05$ and initial perturbation $\sigma/\Delta = 0.10$. (a) $it = 0$ and (b) $it = 10$.



(a) (b)

Figure 3: Static test case. $\Delta/L = 0.025$ and initial perturbation $\sigma/\Delta = 0.10$. ∇C magnitude (a) $it = 0$ and (b) $it = 10$. Note that images have different colorbars.



(a) $it = 0$ (b) $it = 10$

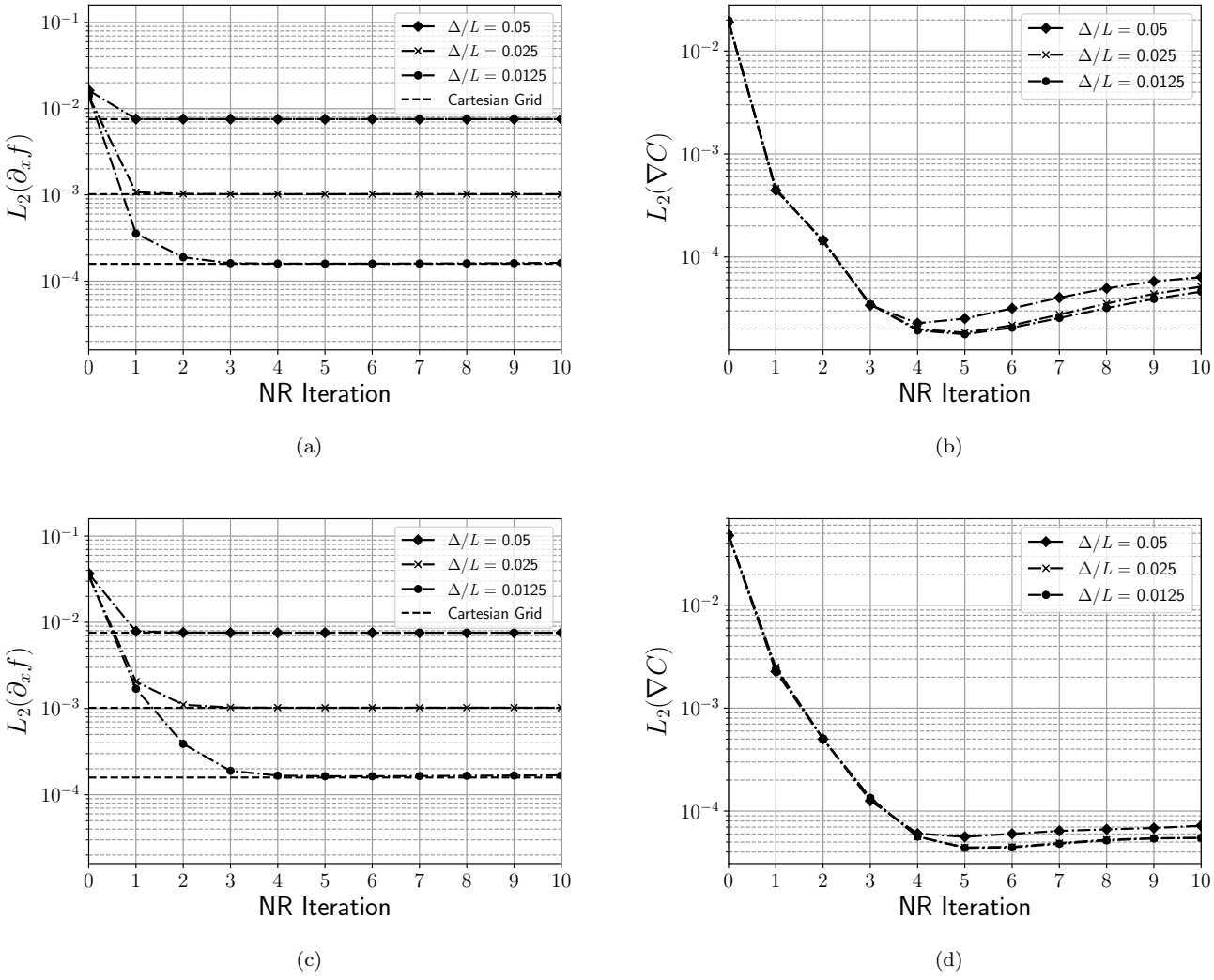
(c) $it = 0$ (d) $it = 10$

Figure 4: Static test case. $\Delta/L = 0.025$ and initial perturbation $\sigma/\Delta = 0.10$. $\left\|\partial_x f^{SPH} - \partial_x f^{an}\right\|$ (top) and $\partial_x f^{SPH}$ (bottom). Note that images have different colorbars.

The error on the test function gradient and the test function itself are shown in Figure 4. It can be noted that the error on the test function gradient is reduced (Fig. 4(a), Fig. 4(b)) and the quality of the SPH approximation for the test function itself increases and distortions, due to the anisotropy of particle distribution, are corrected (Fig. 4(c), Fig. 4(d)).

As previously mentioned, different resolutions have been analysed with an increasingly initial disorder, and it has been verified that the IIPS algorithm is able to generate particle distributions that globally minimizes the $L_2\left(\nabla C\right)$. Herein, two different values of σ have been considered, as initial maximum values of perturbation, and results are reported in Figure 5.

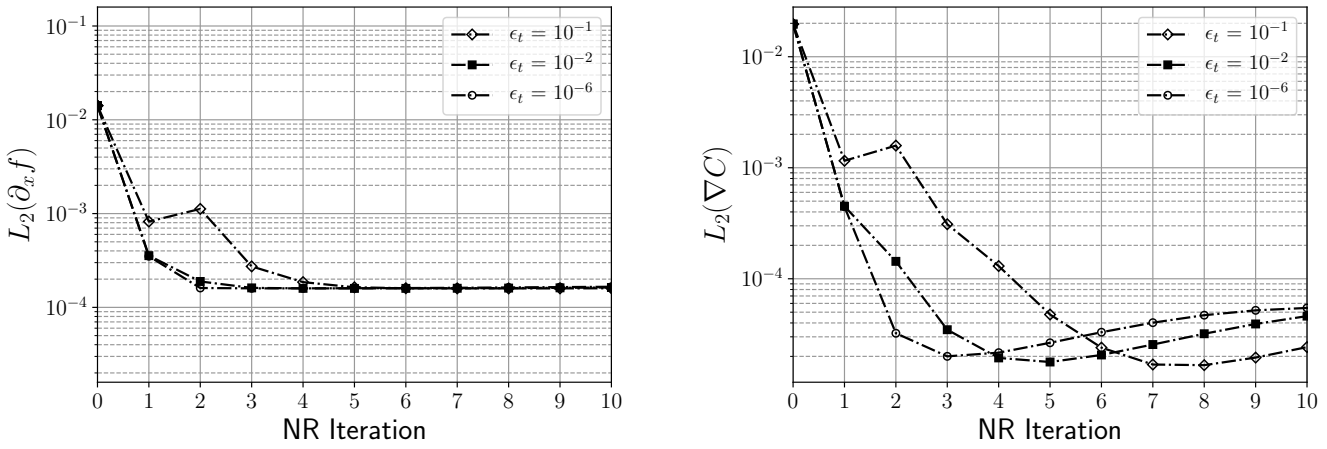
In Figure 5 (Fig. 5(b), Fig. 5(d)) the $L_2\left(\nabla C\right)$ for three different levels of particle size ($\Delta/L = 0.05, 0.025, 0.0125$) are shown against the Newton-Raphson (NR) iterations of the IIPS. It can be observed that the $L_2\left(\nabla C\right)$ is reduced by almost three orders of magnitude and that the minimum value is reached after approximately 5 Newton-Raphson iterations, regardless of the resolution adopted and the initial level of particle disorder. This demonstrates that the IIPS algorithm is robust and can generate a uniform



(a) (b)

(c) (d)

Figure 5: Static test case. Initial perturbation $\sigma/\Delta = 0.10$ (top) and $\sigma/\Delta = 0.25$ (bottom).



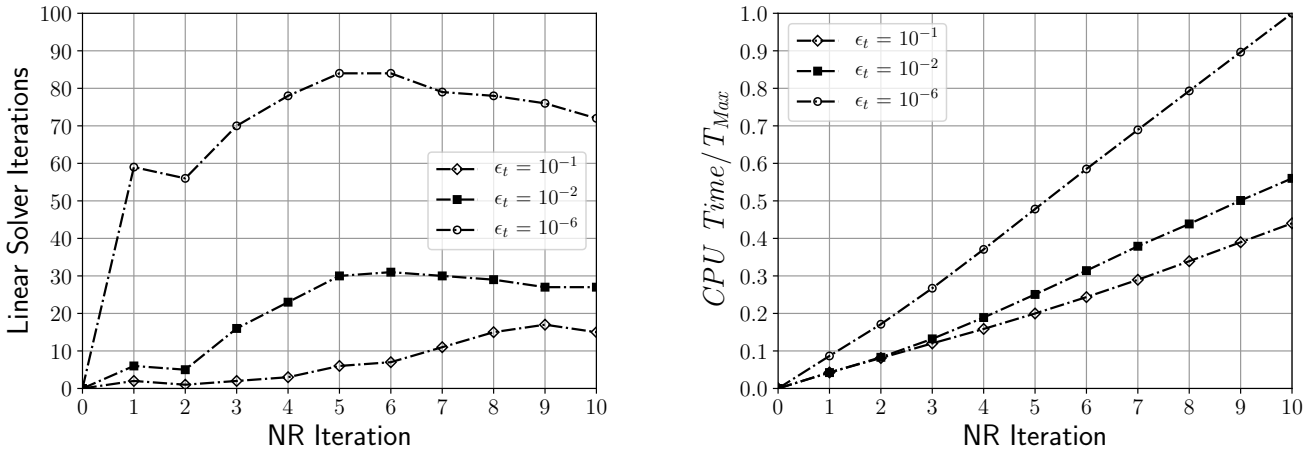
(a) (b)

Figure 6: Static test case. $\Delta/L = 0.0125$ and initial perturbation $\sigma/\Delta = 0.10$.

particle distribution also starting from a significant initial disorder with a small number of Newton-Raphson iterations. In order to assess the quality of the particle distribution in terms of accuracy of the SPH spatial operators, the $L_2\left(\partial_x f\right)$ norm is shown in Figure 5 (Fig. 5(a), Fig. 5(c)). As a reference, in the same figure, the same norms obtained adopting no initial disorder (which corresponds to an initial Cartesian particle distribution) are plotted with dashed lines. It can be noticed that, with reference to the two different initial values of σ/Δ, an accuracy similar to the Cartesian one is achieved with less than 4 iterations, demonstrating that the IIPS algorithm is able to remove the effects of particle anisotropy from the SPH operators in a few iterations. The initial tests described above have been conducted using a tolerance ϵ_t value equal to 10^{-6} for the linear system iterative solver. However, since the value of ϵ_t clearly affects the efficiency, additional investigations have been carried out to analyze its effect on the overall accuracy of the IIPS algorithm. In particular, three different value of $\epsilon_t = 0.1, 0.01, 10^{-6}$ have been adopted with a particle spacing equal to $\Delta/L = 0.0125$ and an initial particle disorder corresponding to $\sigma/\Delta = 0.10$.

Figure 6 shows the norms $L_2\left(\nabla C\right)$ and $L_2\left(\partial_x f\right)$ (as defined in Equations 43 and 46) against the number of Newton-Raphson iterations for the three different values of ϵ_t. While, with $\epsilon_t = 0.01$, and 10^{-6} results are similar, for $\epsilon_t = 0.1$ both norms are not monotonically decreasing, suggesting that this value is not adequate to obtain a robust convergence of the IIPS algorithm. The number of the linear solver iterations and the cumulative computational time for each Newton-Raphson iteration are also shown in Figure 7. It is clear that the number of linear solver iterations required decreases using $\epsilon_t = 0.01$, and thus, as a good compromise between computational cost and accuracy, this value will be used for the rest of the simulations.

A convergence analysis has been performed by running the IIPS with different particle sizes and considering an initial particle distribution that corresponds to $\sigma/\Delta = 0.10$. In Figure 8 the minimum value of $L_2(\partial_x f)$ is shown for each simulation with $h/\Delta = 2.0, 1.6, 1.3$. In Table 1 the numerical values obtained are reported together with the order of convergence θ. The SPH spatial interpolation has two



(a) (b)

Figure 7: Static test case. (a) iteration needed to reach the ϵ value by the linear system solver, (b) normalized CPU time, scaled on $\epsilon = 10^{-6}$, using tolerance (normalized residual) value in the linear system solver.

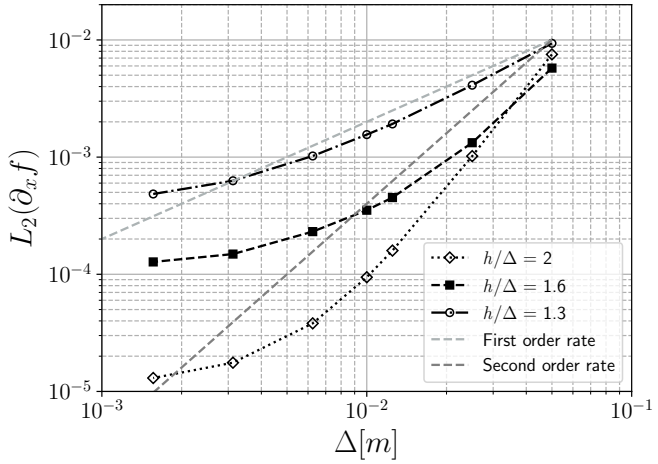


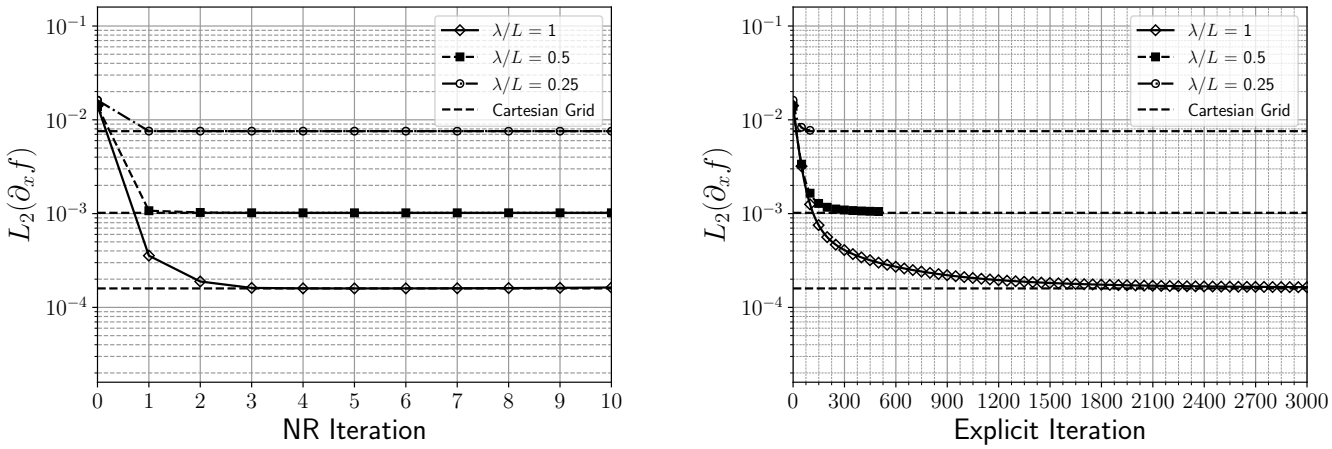
Figure 8: Static test case. Convergence analysis results, $L_2(\partial_x f)$ for different h/Δ values.

	$\frac{h}{\Delta} = 2.0$		$\frac{h}{\Delta} = 1.6$		$\frac{h}{\Delta} = 1.3$	
Δ/L	$L_2(\partial_x f)$	θ	$L_2(\partial_x f)$	θ	$L_2(\partial_x f)$	θ
0.05	0.007532		0.005754		0.008354	
0.025	0.001020	2.89	0.001327	2.12	0.004109	1.19
0.0125	0.000159	2.68	0.000452	1.55	0.001919	1.10
0.01	0.000094	2.35	0.000352	1.11	0.001557	0.92
0.00625	0.000038	1.93	0.000231	0.90	0.001022	0.89
0.003125	0.000018	1.11	0.000149	0.64	0.000628	0.70
0.0015625	0.000013	0.43	0.000127	0.22	0.000484	0.38

Table 1: Static test case. Convergence analysis results, $L_2(\partial_x f)$ and convergence ratio θ.

main different sources of error, called the smoothing and the discretization error, which are generated by the approximation of Equation (3) and by the quadrature Equation (4). For each h/Δ adopted in the convergence analysis, the $L_2(\partial_x f)$ reduces when the smoothing error is much larger than the discretization error [39]. Smaller h/Δ corresponds to larger discretization error, which therefore becomes dominant for a larger Δ causing saturation for the overall convergence. Looking at the convergence rate θ in Table 1, for the largest value of $h/\Delta = 2.0$, it can be noticed that values greater than the theoretical order of convergence (equal to 2 for the adopted kernel) are obtained for $\Delta/L > 0.0125$ m, this is due to the fact that the IIPS scheme reduces also the discretization error when the resolution is increased. By comparing the results of Figure 8 with the one shown in Figure 1 (where the SPH error did not reduce while the resolution was increased), the capability of the IIPS procedure to increase the accuracy and to restore the convergence of the SPH spatial interpolation becomes apparent and this implies that the discretization error is removed or made much smaller by the IIPS operation without any kernel or kernel gradient correction.

For the static test case, the iterative implicit method has been compared with the iterative Fickian shifting explicit method [23], [49], with the purpose to assess the number of iterations and the computational time needed to reach a pre-defined level of the $L_2(\partial_x f)$. The analysis has been conducted using $h/\Delta = 2.0$ and $\Delta/L = 0.0125$, considering different $\lambda/L = 1, 0.5, 0.25$ which correspond respectively to a 40, 20, 10 particles for each periodicity in the test function of Equation (45). Figure 9 shows the $L_2(\nabla f)$ for $\lambda/L = 1, 0.5, 0.25$ against the number of required iterations with the IIPS (Fig. 9(a)) and explicit iterative shifting (Fig. 9(b)). As a reference, the $L_2(\nabla f)$ values obtained by adopting a Cartesian particle distribution are plotted with dashed lines. In Table 2 the number of iterations needed to reach the Cartesian grid accuracy, (with a 5% tolerance) are shown together with the computational time required for each simulation. While the IIPS reaches the same level of spatial interpolation error obtained with the Cartesian particle distribution in a maximum of 3 Newton-Raphson iterations, the number of iterations necessary to achieve a similar accuracy with the explicit iterative Fickian shifting is much larger on the order of 68, 380 and 2408. Moreover, for the explicit algorithm, the total number of iterations strongly depends on λ. Therefore, despite the fact that each iteration of the IIPS has a larger computational cost (due to the fact that a linear system with $d \times n$ unknowns has to be solved), it is



(a) (b)

Figure 9: Static test case. $\Delta/L = 0.0125$ and initial perturbation $\sigma/\Delta = 0.10$. (a) implicit iterative particle shifting method, (b) explicit iterative shifting method.

	Implicit		Explicit	
	Iteration	Time $[ms]$	Iteration	Time $[ms]$
$\lambda/L = 0.25$	1	66	68	2619
$\lambda/L = 0.5$	2	126	380	13964
$\lambda/L = 1.0$	3	207	2408	98302

Table 2: Static test case. Particle Shifting Techniques comparison

always more efficient, with speedups varying from 40 for $\lambda/L = 0.25$ to 475 for $\lambda/L = 1$. The estimated extra memory required for the implicit procedure is proportional to $d \times n \times n_b$, where the size of the kernel support h/Δ determines the number of neighbour particles n_b, that correspond to the number of non-null elements on the matrix, and to $d \times n$ that is needed to allocate the unknowns vector.

4.2. Kinematic test case

The second validation test is a purely kinematic case, the aim is to test the capability of IIPS in obtaining low discretization error for a generic test function f, while an external motion is maintained to introduce disorder in the particle distribution. Particles move following the Lagrangian trajectories defined by the integration in time of the analytical TGV velocity field:

$$\begin{aligned} \Delta x_i &= -Ucos(2\pi x_i/L)sin(2\pi y_i/L)\Delta t \\ \Delta y_i &= Usin(2\pi x_i/L)cos(2\pi y_i/L)\Delta t \end{aligned} \tag{47}$$

where Δx_i and Δy_i are the particles' displacements for the i-th particle on the x and y axis, respectively. In Equation (47), U, the reference velocity, is equal to 1 m/s and L, the size of the squared domain, is equal to 1 m. The reference speed of sound $c_0 = 10U$ and the CFL number equal to 0.2 have been set to computed an equivalent time step size, $\Delta t = 0.2(\Delta/c_0)$, to obtain the particle displacement. The IIPS procedure, illustrated in the previous section, can be applied during a physical simulation in a different

way. In order to assess which one guarantees the lower error in the SPH interpolation and optimizes the computational costs, two methodologies, *Procedure A*, and *Procedure B* have been investigated and presented respectively in Algorithm 2 and Algorithm 3.

The initial particle distribution has been obtained introducing an iterative pre-procedure (line 2 of Algorithm 2 and 3), which corresponds to the test case described in Section (4.1), considering 100 Newton-Raphson iterations. Using the pre-procedure the minimal value of $L_\infty(\nabla C)_{Init}$ is obtained and it is utilized to set the threshold used to trigger the IIPS, $L_\infty(\nabla C)_{Thr}$, (line 3 of Algorithm 2 and 3), which is set equal to:

$$L_\infty(\nabla C)_{Thr} = \beta L_\infty(\nabla C)_{Init}, \tag{48}$$

where β is an arbitrary coefficient. At each physical time step the particle distribution has to fulfil the condition, (line 5 of Algorithm 2 and 3):

$$L_\infty(\nabla C) \leq L_\infty(\nabla C)_{Thr}. \tag{49}$$

Therefore, the maximum value of $h|\nabla C_i|$ has to be always below the fixed threshold otherwise the IIPS procedure is activated. This condition is valid in both procedures, the only difference is the extra condition added in Algorithm 2, it requires to run at least one Newton-Raphson iteration at each physical time step even if the condition, defined in Equation (49), is satisfied. The IIPS procedure runs for an unlimited number of Newton-Raphson iterations until each particle has $h|\nabla C_i|$ that meet the condition.

Algorithm 2 Kinematic case

```
1: procedure A Implicit iterative                                   ▷
2:     Static test case (Algorithm 1)
3:     Set L_∞(∇C)_Thr =β L_∞(∇C)_Init
4:     Start Simulation
5:     while (L_∞(∇C) ≤ L_∞(∇C)_Thr & NR_it ≥ 1) do
6:         AssembleMatrix(∇C_i)
7:         LinearSystemSolver(AssembleMatrix)
8:         UpdateParticlePosition(Implicit PST)
9:         Compute L_∞(∇C), L_2(∂_x f)
10:    end while
11:    ParticleDisplacement (Eq. (47))
12: end procedure
```

In *Procedure A* and in *Procedure B* two different factors have been tested, β=5 and β=10, but because at least one Newton-Raphson iteration is required in *Procedure A*, $L_\infty(\nabla C)$ never reaches the imposed threshold and actually the results, as presented later, do not depend on β. This test case has been studied with $\Delta/L = 0.025$ (which corresponds to 20 particles across the vortex diameter) and $h/\Delta = 2.0$, the test function defined in Equation (45) has $\lambda/L = 0.5$. Results in Figure 10 and Figure 11 show the comparison between the different implicit procedures described previously and the explicit method similar to the one described in [23], defined as *Procedure C* in Algorithm 4. The explicit shifting technique is called in line 8 of Algorithm 4, at each time step the particle position is updated using

Algorithm 3 Kinematic case

procedure B IMPLICIT ITERATIVE ▷
    Static test case (Algorithm 1)
    Set $L_\infty(\nabla C)_{Thr} = \beta\ L_\infty(\nabla C)_{Init}$
    Start Simulation
    while $(L_\infty(\nabla C) \leq L_\infty(\nabla C)_{Thr})$ **do**
        AssembleMatrix(∇C_i)
        LinearSystemSolver(AssembleMatrix)
        UpdateParticlePosition(Implicit PST)
        Compute $L_\infty(\nabla C)$, $L_2(\partial_x f)$
    end while
    ParticleDisplacement (Eq. (47))
end procedure

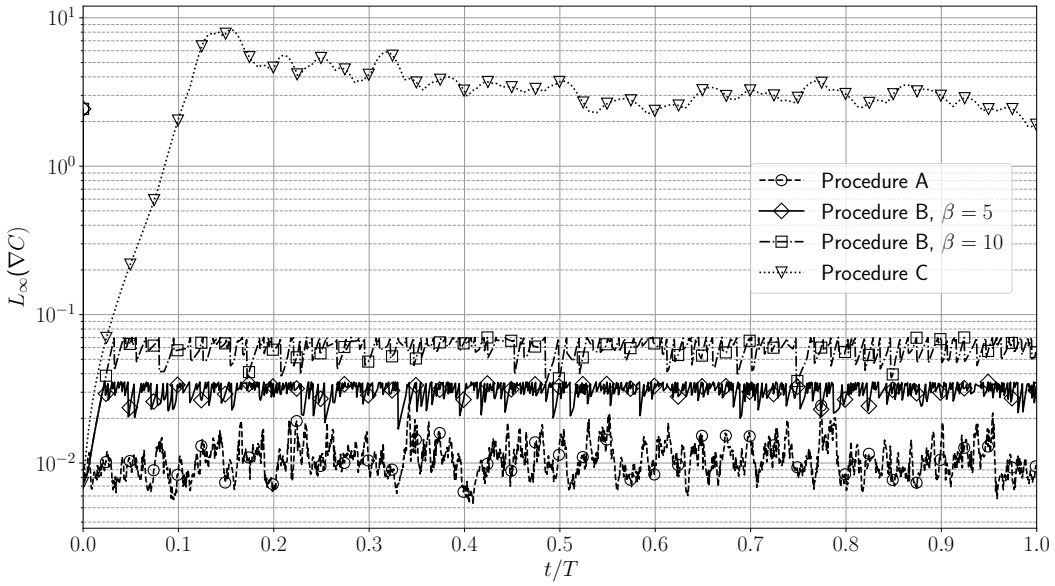
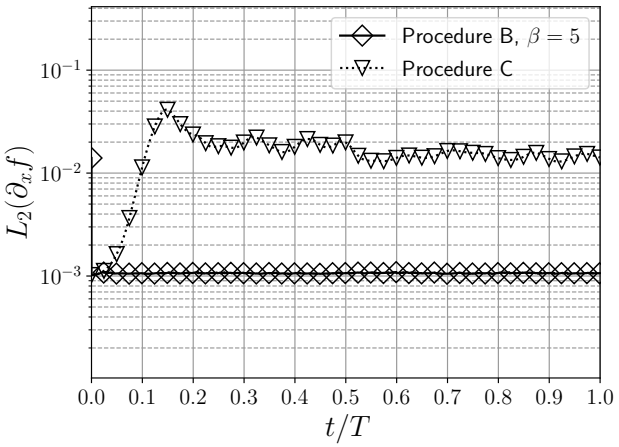
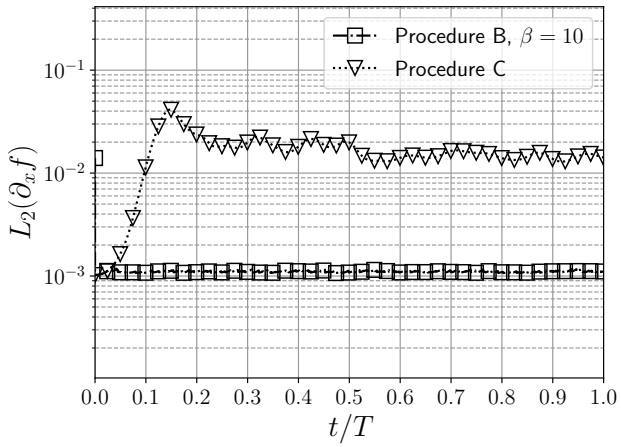


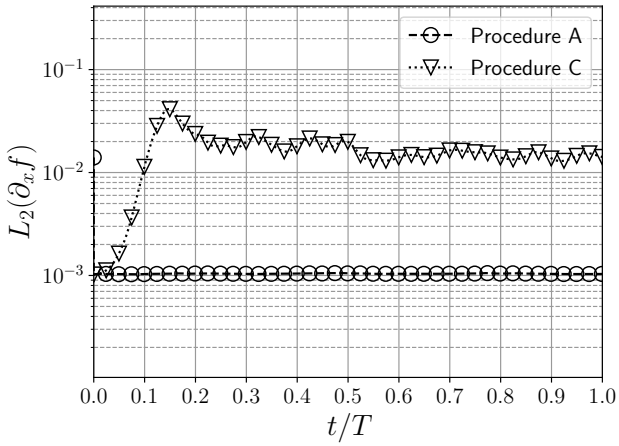
Figure 10: Kinematic test case. Procedure comparison, $\Delta/L = 0.025$ and $h/\Delta = 2.0$.



(a)



(b)



(c)

Figure 11: Kinematic test case. Procedure comparison, $\Delta/L = 0.025$ and $h/\Delta = 2.0$.

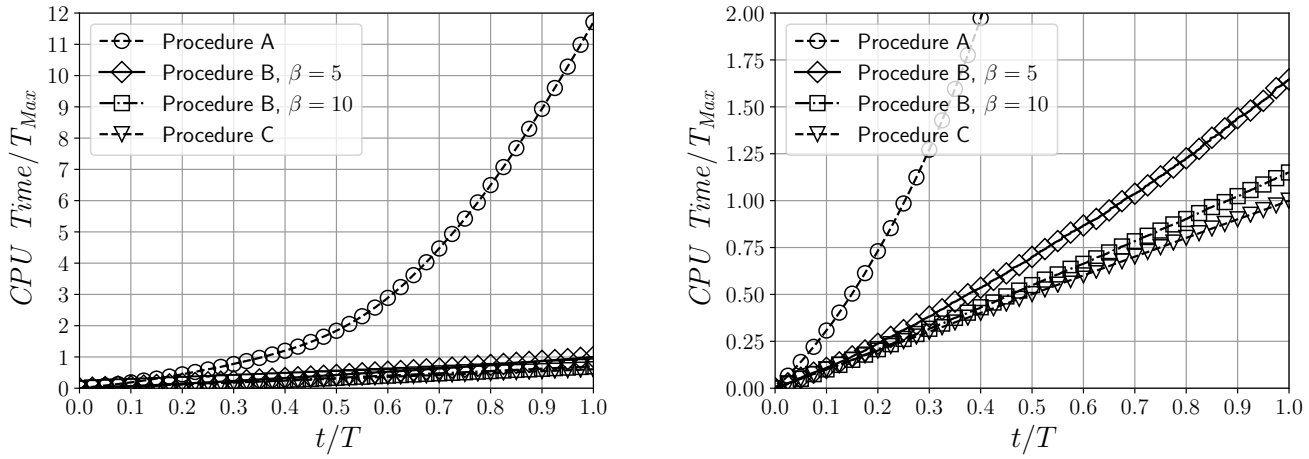


Figure 12: Kinematic test case. Method comparison, CPU time. Note different scales.

Equation (6) and similarly to Equation (7) the shifting vector $\delta_{\mathbf{i}}$ for particle ith is,

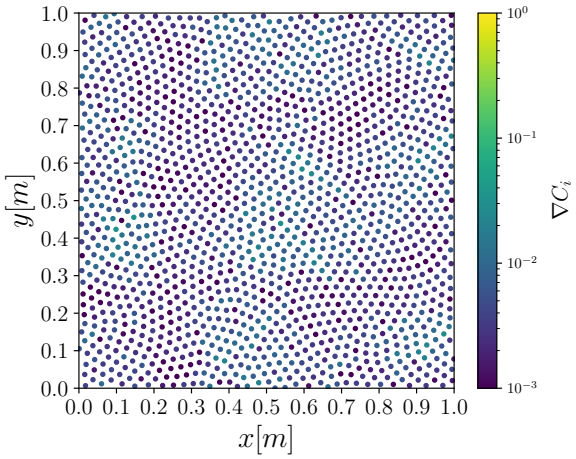
$$\boldsymbol{\delta}_i = \Delta t^2 c_0^2 \sum_{j=1}^{J} \nabla W_{ij} \omega_j. \tag{50}$$

Algorithm 4 Kinematic case.

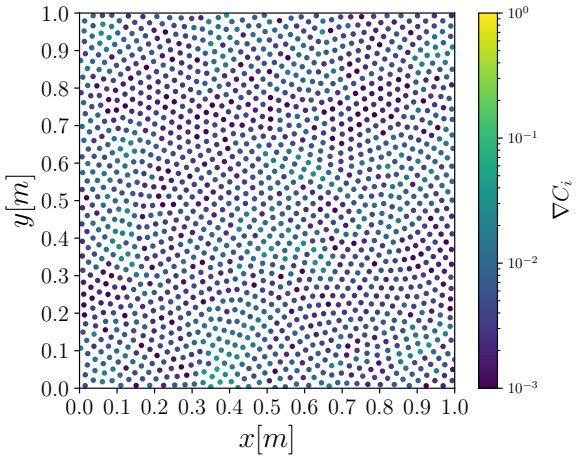
1: **procedure** C EXPLICIT ▷
2: Static test case (Algorithm 1)
3: Start Simulation
4: UpdateParticlePosition(Explicit PST)
5: Compute $L_\infty(\nabla C)$, $L_2(\partial_x f)$
6: ParticleDisplacement (Eq. (47))
7: **end procedure**

The implicit procedures show different trends analysing $L_\infty(\nabla C)$ (Figure 10) and similarities for $L_2(\partial_x f)$ (Figure 11). In all the different implicit procedures, a gain of more than an order of magnitude in accuracy in the test function and almost two orders of magnitude in the particle concentration is obtained.

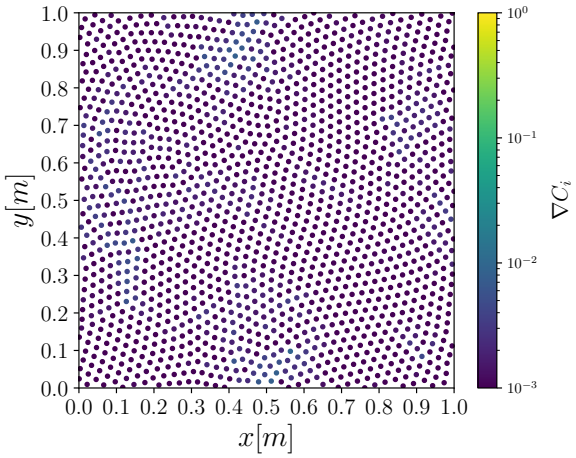
The analysis of computational cost (the total CPU time for Algorithm 2, Algorithm 3 and Algorithm 4) can be seen in Figure 12, results are re-scaled with explicit method computational time. These implicit iterative procedures have been compared with a non-iterative explicit procedure because as shown in the static test case (Figure 9) an iterative explicit methodology is not always sustainable in terms of computational effort (see Table 2). The *Procedure A*, which calls the IIPS at each time step, significantly increases the computational cost, the trend is not linear (Figure 12(a)) due to the significant increase in the number of iterations required to the linear solver to converge. This can be explained by the fact that the IIPS procedure is activated when the $|\nabla C_i|$ is already close to the minimal value, therefore the linear solver needs a large number of iterations to reduce the normalized residuals. The



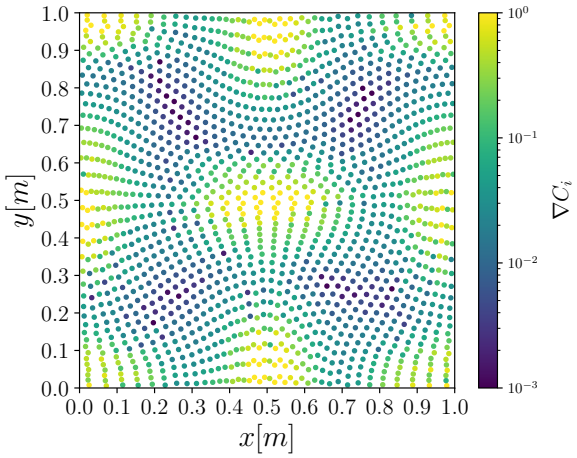
(a) Procedure B Implicit iterative, $\beta = 5$



(b) Procedure B Implicit iterative, $\beta = 10$



(c) Procedure A Implicit iterative



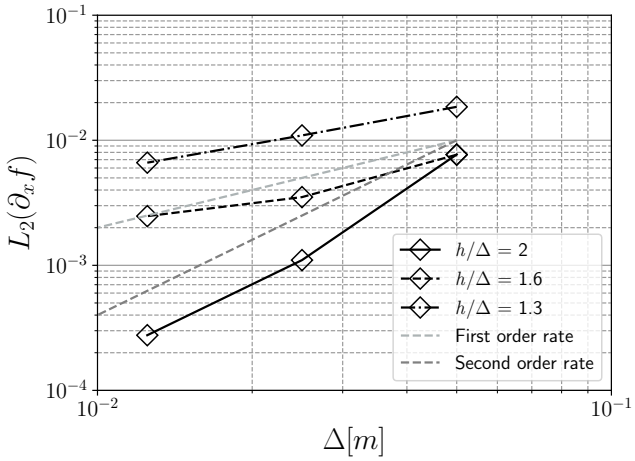
(d) Procedure C Explicit not iterative

Figure 13: Kinematic test case. Particle distribution at $t = 0.2\ s$.

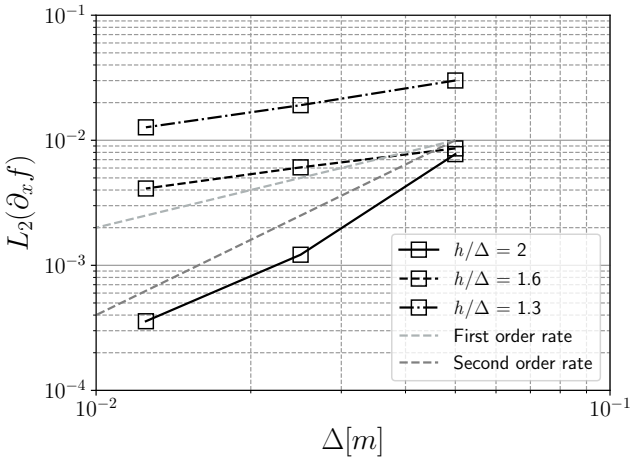
Procedure B calls the IIPS whenever the $L_\infty(\nabla C)$ value rises above the set threshold and it generates an extra computational cost between 10 to 70 percent based on β, (Figure 12(b)). Figure 13 shows the particle distributions, it can be immediately noted that the implicit procedures maintain the particles uniformly distributed avoiding the creation of distinct streamlines in the domain.

Results presented in Figure 14 shows the convergence analysis for all the different procedures, using $\Delta/L = 0.05, 0.025, 0.0125$ and $h/\Delta = 1.3, 1.6, 2.0.$, The *Procedure A* shows the same results regardless of the value of β, as it was already seen for just a single resolution and kernel size in Figure 11.

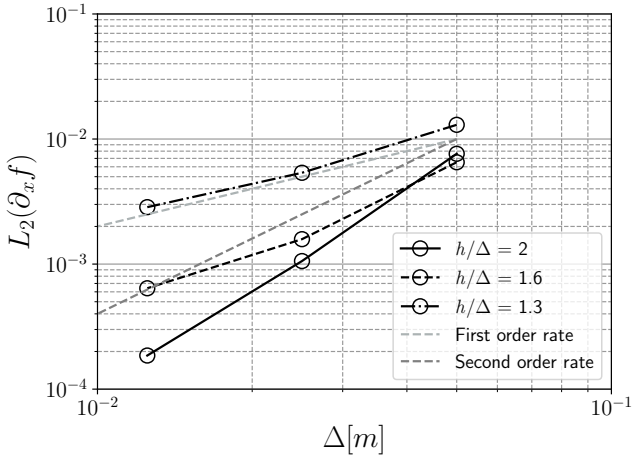
The convergence rates for results presented in Figure 14 are reported in Table 3, where it can be noted that these results are similar to the convergence rates obtained in the static test case Table 1, meaning that even with a continuous source of perturbation the IIPS is able to prevent anisotropy in particle distribution, differently to the explicit procedure which convergence rate is lower.



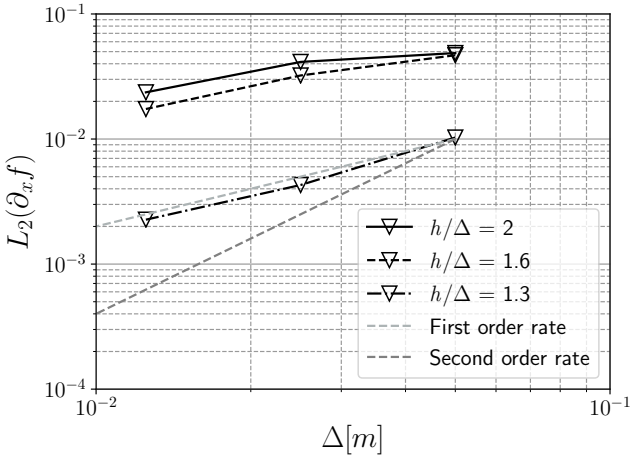
(a) Procedure B Implicit iterative, $\beta = 5$



(b) Procedure B Implicit iterative, $\beta = 10$



(c) Procedure A Implicit iterative



(d) Procedure C Explicit not iterative $x[m]$

Figure 14: Kinematic test case. Convergence analysis.

	Implicit iterative all		Implicit iterative β=5		Implicit iterative β=10		Explicit not iterative	
	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2	θ_1	θ_2
$\frac{h}{\Delta} = 2.0$	2.85	2.51	2.81	1.99	2.67	1.76	0.23	0.81
$\frac{h}{\Delta} = 1.6$	2.04	1.30	1.12	0.51	0.51	0.56	0.53	0.89
$\frac{h}{\Delta} = 1.3$	1.27	0.91	0.76	0.72	0.65	0.58	1.26	0.92

Table 3: Kinematic test case. Convergence ratio θ_1 and θ_2 respectively for $\Delta/L = 0.025$ and $\Delta/L = 0.0125$

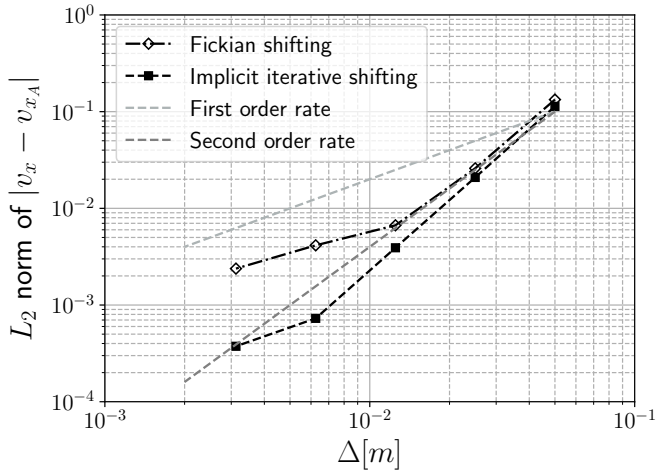


Figure 15: TGV with ALE-SPH solver. Convergence analysis, L_2 norm of $|v_x - v_{xA}|$, particle shifting technique comparison.

4.3. Taylor Green vortex with ALE-SPH solver

Differently from the results presented in Section 4.2, here, a viscous Taylor-Green vortex is simulated adopting the ALE-SPH solver described in Section 2.2, with the aim of evaluating the effects of the IIPS algorithm, embedded in a numerical scheme for the Navier-Stokes equations. For this test case there is an analytical solution for velocity and pressure which is the following:

$$\begin{aligned} v_{xA} =& U e^{-8\pi t/Re} cos(2\pi x_i/L) sin(2\pi y_i/L) \\ v_{yA} =& U e^{-8\pi t/Re} sin(2\pi x_i/L) cos(2\pi y_i/L) \\ p_A =& \frac{\rho}{4} e^{-16\pi t/Re} \left[cos(4\pi x_i/L) + cos(4\pi y_i/L)\right] \end{aligned} \tag{51}$$

where v_{xA} and v_{yA} are the analytical velocity components and p_A the analytical pressure. For this specific case it has been set $U = 1$ m/s, $\rho = 1000$ kg/m^3, the size of the squared domain L equal to 1 m $\times$ 1 m, and the kinematic viscosity $\nu = 0.01$ m^2/s, is used to compute the Reynolds number, $Re = 100$. In the following simulations, the field quantities are reconstructed at the interface using the Monotonic Upstream-Centred Scheme for Conservation Laws (MUSCL) [53], no artificial viscosity is used and the laminar viscous forces in the momentum equation are discretized as reported in [31]. The solver updates Equations (10) in time using a third-order Runge-Kutta scheme. In the above-mentioned solver, a Fickian-based PST was already implemented, [33], with the correction term introduced in the transport velocity $\boldsymbol{v}_0$ formulation, with the addition of a strong background pressure. This model has been used as a reference point to assess the improvements that the novel IIPS algorithm can produce while solving the full set of Navier-Stokes equations. The IIPS method has been implemented in the numerical SPH-ALE solver, following the *Proceduce B* in Algorithm 3, described in Section 4.2 with $L_\infty(\nabla C)_{thr} = 0.001/h$. It is relevant to highlight that the IIPS procedure can be applied in others particle methods in a similar way.

The convergence analysis has been performed using $\Delta/L = 0.05, 0.025, 0.0125, 0.00625, 0.003125$ and the value of h/Δ has been kept constant and equal to 2. The results of the L_2 norm of the error of

Δ/L	Fickian shifting θ	Implicit iterative shifting θ
0.05		
0.025	2.36	2.44
0.0125	1.96	2.41
0.00625	0.69	2.42
0.003125	0.80	0.95

Table 4: TGV with ALE-SPH solver. Convergence ratio of L_2 norm of $||v_x - v_{xA}||$, (Figure 15).

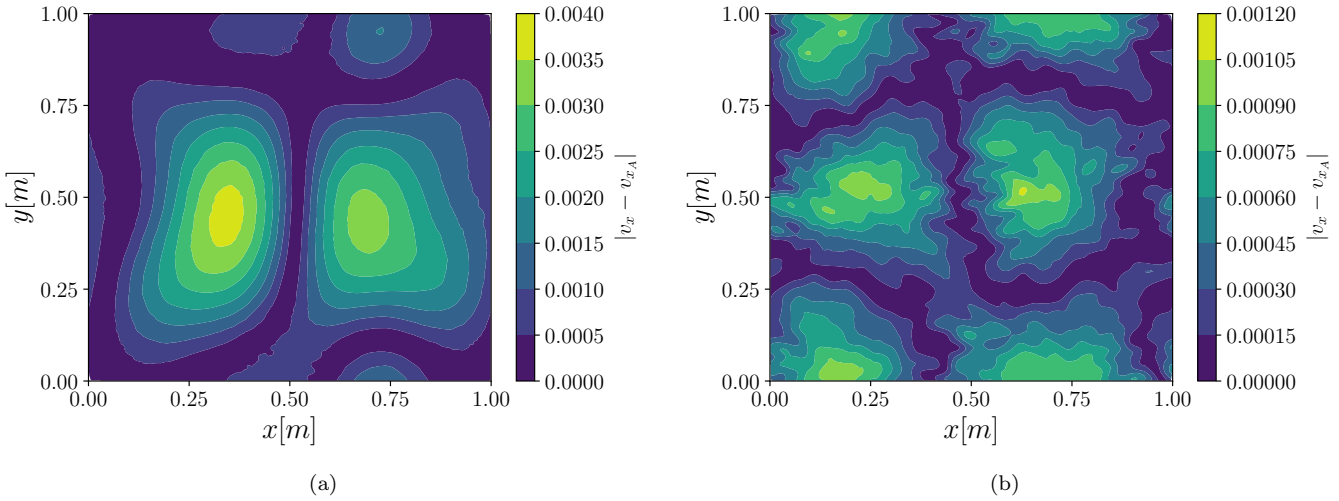


Figure 16: TGV with ALE-SPH solver. Error of the x component of the velocity. $\Delta/L = 0.00625$, physical time $t = 2$ s. Particle shifting comparison (a) Fickian shifting and (b) Implicit iterative shifting.

the x velocity component, $|v_x - v_{xA}|$, are shown in the Figure 15 and reported in Table 4. The higher-than-second-order convergence is maintained, and it can be seen a gain of almost one order of magnitude obtained while reaching saturation.

Figure 16 shows $|v_x - v_{xA}|$, with the resolution $\Delta/L = 0.00626$. The comparison between the different particle shifting methods shows that the IIPS is able to improve the accuracy of the ALE-SPH algorithm.

For the same resolution $\Delta/L = 0.00626$ the L_∞ norm of ∇C is shown in Figure 17, the trend is similar to the ones presented in Figure 11: the IIPS algorithm is able to reduce the maximum $L_\infty(\nabla C)$ by two orders of magnitude, in comparison with the one obtained with the Fickian shifting.

5. Conclusions and Perspectives

A novel particle shifting technique, called IIPS, has been presented in this paper. The technique has been developed to be embedded in meshless numerical methods based on kernel basis function. In particular, it is thought to be well suited to the SPH schemes but it can be easily extended to methods based on moving calculation points. The implicit iterative particle shifting has the capability to directly tackle the formation of areas in the domain with non-uniform particle concentration. The new IIPS technique removes the discretisation error obtaining second-order or higher-than-second-order

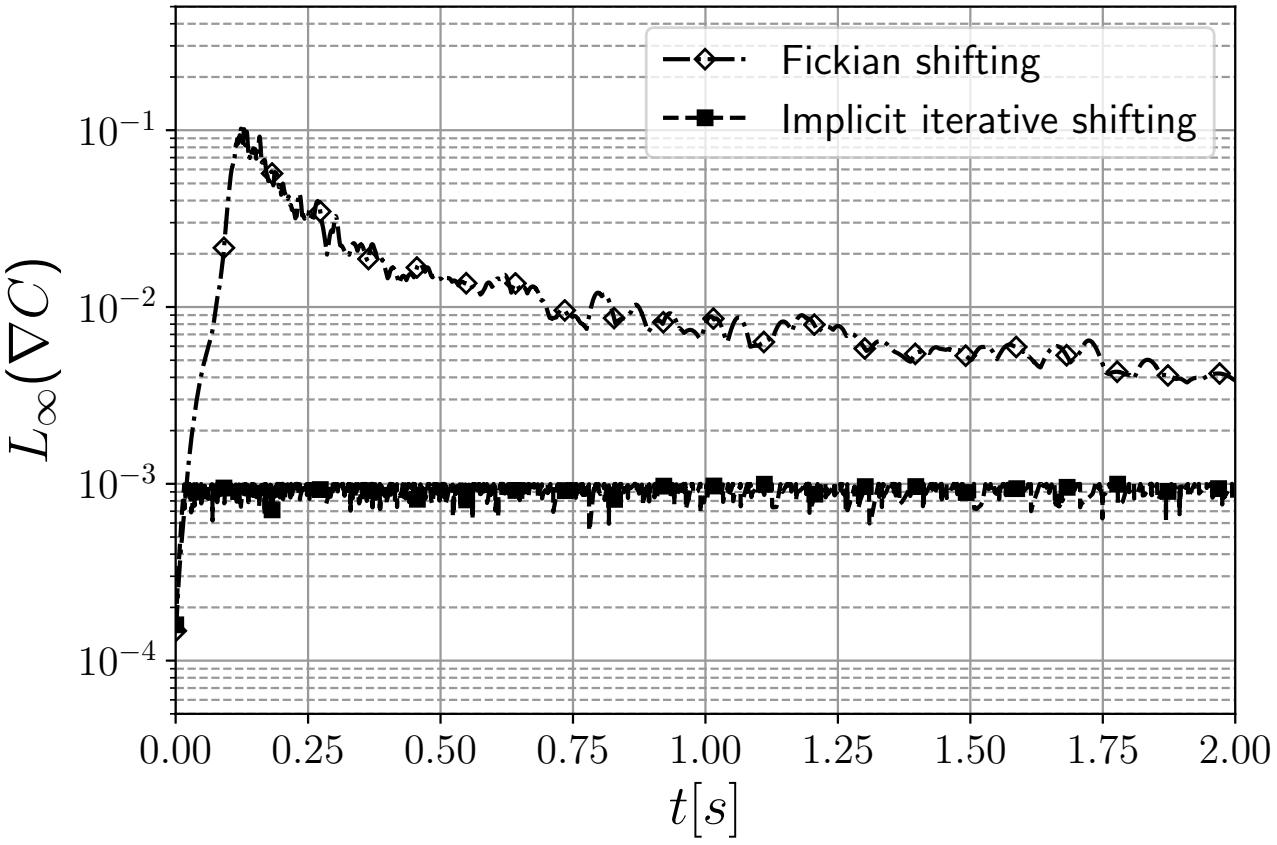


Figure 17: TGV with ALE-SPH solver. $\Delta/L = 0.00625$. Particle shifting comparison $L_\infty(\nabla C)$

convergence without kernel or kernel gradient correction techniques. The IIPS benefits and advantages have been identified by the comparison with an explicit particle shifting technique which represents the state-of-the-art of the shifting methods available in the literature. The methodology introduced an intrinsic difference from explicit shifting being able to link and generate implicit connections between the correction terms for the particle positions in the updated configuration. The particle distribution is a key element for accuracy in an SPH operator, and it is strictly related to the theoretical fundamentals of SPH itself. For this reason, a purely theoretical formulation has been developed trying to solve this weakness, slightly changing the point of view regarding the shifting techniques.

Through static and kinematic numerical experiments, the IIPS algorithm increases the SPH accuracy for the test function by more than one order of magnitude and reduces the gradient of particle concentration by almost two orders of magnitude, without a drastic computational time overhead. Moreover, the portion of the discretization error arising due to irregular particle distribution can be completely eliminated from the SPH differential operators restoring the theoretical convergence rate.

The Taylor Green flow, has been simulated at Re=100, adopting an ALE-SPH algorithm, where the IIPS has been activated when the particle disorder rises above a predefined level. The results for the TGV are in agreement with the static and kinematic numerical experiments and confirms that superiority of the proposed formulation, in comparison with the explicit, non-iterative shifting formulation.

The IIPS represents a completely novel formulation, it has been mainly proposed as a starting point for future investigation, both theoretical and numerical, it is at the first stages of development. Future works will be conducted aiming to introduce the IIPS procedure in a consistent way in a solver for governing equations and extend its applicability for flows with bounded domains, free surfaces and multi-phase applications.

References

[1] C. Altomare, J. M. Domínguez, A. J. C. Crespo, T. Suzuki, I. Caceres, and M. Gómez-Gesteira, "Hybridization of the wave propagation model swash and the meshfree particle method sph for real coastal applications," *Coastal Engineering Journal*, vol. 57, no. 4, pp. 1550024-1-1550024–34, 2015.

[2] A. Amicarelli, J.-C. Marongiu, F. Leboeuf, J. Leduc, and J. Caro, "Sph truncation error in estimating a 3d function," *Computers & Fluids*, vol. 44, no. 1, pp. 279–296, 2011.

[3] M. Antuono, P. Sun, S. Marrone, and A. Colagrossi, "The δ-ale-sph model: An arbitrary lagrangian-eulerian framework for the δ-sph model with particle shifting technique," *Computers & Fluids*, vol. 216, p. 104 806, 2021.

[4] B. Bouscasse, A. Colagrossi, S. Marrone, and A. Souto-Iglesias, "Sph modelling of viscous flow past a circular cylinder interacting with a free surface," *Computers & Fluids*, vol. 146, pp. 190–212, 2017.

[5] H. H. Bui, R. Fukagawa, K. Sako, and S. Ohno, "Lagrangian meshfree particles method (sph) for large deformation and failure flows of geomaterial using elastic–plastic soil constitutive model," *International journal for numerical and analytical methods in geomechanics*, vol. 32, no. 12, pp. 1537–1570, 2008.

[6] A. D. Chow, B. D. Rogers, S. J. Lind, and P. K. Stansby, "Incompressible sph (isph) with fast poisson solver on a gpu," *Computer Physics Communications*, vol. 226, pp. 81–103, 2018.

[7] A. D. Chow, B. D. Rogers, S. J. Lind, and P. K. Stansby, "Numerical wave basin using incompressible smoothed particle hydrodynamics (isph) on a single gpu with vertical cylinder test cases," *Computers & Fluids*, vol. 179, pp. 543–562, 2019.

[8] P. W. Cleary, M. Prakash, R. Das, and J. Ha, "Modelling of metal forging using sph," *Applied Mathematical Modelling*, vol. 36, no. 8, pp. 3836–3855, 2012.

[9] A. Colagrossi, B. Bouscasse, M. Antuono, and S. Marrone, "Particle packing algorithm for sph schemes," *Computer Physics Communications*, vol. 183, no. 8, pp. 1641–1653, 2012.

[10] J. Domınguez, C. Altomare, J. Gonzalez-Cao, and P. Lomonaco, "Towards a more complete tool for coastal engineering: Solitary wave generation, propagation and breaking in an sph-based model," *Coastal Engineering Journal*, vol. 61, no. 1, pp. 15–40, 2019.

[11] D. Duque, "Particle method for phase separation on membranes," *Microfluidics and Nanofluidics*, vol. 22, no. 9, pp. 1–10, 2018.

[12] M. W. Evans, F. H. Harlow, and E. Bromberg, "The particle-in-cell method for hydrodynamic calculations," LOS ALAMOS NATIONAL LAB NM, Tech. Rep., 1957.

[13] T. Fonty, M. Ferrand, A. Leroy, A. Joly, and D. Violeau, "Mixture model for two-phase flows with high density ratios: A conservative and realizable sph formulation," *International Journal of Multiphase Flow*, vol. 111, pp. 158–174, 2019.

[14] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars," *Monthly notices of the royal astronomical society*, vol. 181, no. 3, pp. 375–389, 1977.

[15] D. Hietel, K. Steiner, and J. Struckmeier, "A finite-volume particle method for compressible flows," *Mathematical Models and Methods in Applied Sciences*, vol. 10, no. 09, pp. 1363–1382, 2000.

[16] M. L. Hosain, J. Domınguez, R. B. Fdhila, and K. Kyprianidis, "Smoothed particle hydrodynamics modeling of industrial processes involving heat transfer," *Applied Energy*, vol. 252, p. 113 441, 2019.

[17] H. Hu and P. Eberhard, "Thermomechanically coupled conduction mode laser welding simulations using smoothed particle hydrodynamics," *Computational Particle Mechanics*, vol. 4, no. 4, pp. 473–486, 2017.

[18] E. J. Kansa, "Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—i surface approximations and partial derivative estimates," *Computers & Mathematics with applications*, vol. 19, no. 8-9, pp. 127–145, 1990.

[19] A. Khayyer, H. Gotoh, H. Falahaty, and Y. Shimizu, "An enhanced isph–sph coupled method for simulation of incompressible fluid–elastic structure interactions," *Computer Physics Communications*, vol. 232, pp. 139–164, 2018.

[20] A. Khayyer, H. Gotoh, and Y. Shimizu, "A projection-based particle method with optimized particle shifting for multiphase flows with large density ratios and discontinuous density fields," *Computers & Fluids*, vol. 179, pp. 356–371, 2019.

[21] A. Khayyer, H. Gotoh, and Y. Shimizu, "Comparative study on accuracy and conservation properties of two particle regularization schemes and proposal of an optimized particle shifting scheme in isph context," *Journal of Computational Physics*, vol. 332, pp. 236–256, 2017.

[22] S. Koshizuka and Y. Oka, "Moving-particle semi-implicit method for fragmentation of incompressible fluid," *Nuclear science and engineering*, vol. 123, no. 3, pp. 421–434, 1996.

[23] S. Lind, R. Xu, P. Stansby, and B. D. Rogers, "Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves," *Journal of Computational Physics*, vol. 231, no. 4, pp. 1499–1523, 2012.

[24] G.-R. Liu and M. B. Liu, *Smoothed particle hydrodynamics: a meshfree particle method*. World scientific, 2003.

[25] M. Liu and G. Liu, "Smoothed particle hydrodynamics (sph): An overview and recent developments," *Archives of computational methods in engineering*, vol. 17, no. 1, pp. 25–76, 2010.

[26] J.-C. Marongiu, F. Leboeuf, J. Caro, and E. Parkinson, "Free surface flows simulations in pelton turbines using an hybrid sph-ale method," *Journal of Hydraulic Research*, vol. 48, no. S1, pp. 40–49, 2010.

[27] S. Marrone, M. Antuono, A. Colagrossi, G. Colicchio, D. Le Touzé, and G. Graziani, "δ-sph model for simulating violent impact flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 13-16, pp. 1526–1542, 2011.

[28] J. J. Monaghan, "Simulating free surface flows with sph," *Journal of computational physics*, vol. 110, no. 2, pp. 399–406, 1994.

[29] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annual review of astronomy and astrophysics*, vol. 30, no. 1, pp. 543–574, 1992.

[30] J. J. Monaghan, H. E. Huppert, and M. G. Worster, "Solidification using smoothed particle hydrodynamics," *Journal of Computational Physics*, vol. 206, no. 2, pp. 684–705, 2005.

[31] J. P. Morris, P. J. Fox, and Y. Zhu, "Modeling low reynolds number incompressible flows using sph," *Journal of computational physics*, vol. 136, no. 1, pp. 214–226, 1997.

[32] R. Nestor and N. Quinlan, "Extension of the finite volume particle method to higher order accuracy and viscous flow.," *" SPHERIC-Smoothed Particle Hydrodynamics European Research Interest Community"*., p. 95, 2007.

[33] M. Neuhauser, "Development of a coupled sph-ale/finite volume method for the simulation of transient flows in hydraulic machines," Ph.D. dissertation, Ecully, Ecole centrale de Lyon, 2014.

[34] G. Oger, S. Marrone, D. Le Touzé, and M. De Leffe, "Sph accuracy improvement through the combination of a quasi-lagrangian shifting transport velocity and consistent ale formalisms," *Journal of Computational Physics*, vol. 313, pp. 76–98, 2016.

[35] E. Oñate, S. Idelsohn, O. Zienkiewicz, and R. Taylor, "A finite point method in computational mechanics. applications to convective transport and fluid flow," *International journal for numerical methods in engineering*, vol. 39, no. 22, pp. 3839–3866, 1996.

[36] S. Pineda, J.-C. Marongiu, S. Aubert, and M. Lance, "Simulation of a gas bubble compression in water near a wall using the sph-ale method," *Computers & Fluids*, vol. 179, pp. 459–475, 2019.

[37] D. J. Price and J. J. Monaghan, "Smoothed particle magnetohydrodynamics—iii. multidimensional tests and the· b= 0 constraint," *Monthly Notices of the Royal Astronomical Society*, vol. 364, no. 2, pp. 384–406, 2005.

[38] N. J. Quinlan, "Extensions of the meshless finite volume particle method (fvpm) for static and dynamic free-surface flows," *Computers & Fluids*, vol. 177, pp. 33–45, 2018.

[39] N. J. Quinlan, M. Basa, and M. Lastiwka, "Truncation error in mesh-free particle methods," *International Journal for Numerical Methods in Engineering*, vol. 66, no. 13, pp. 2064–2085, 2006.

[40] M. Rentschler, J. Marongiu, M. Neuhauser, and E. Parkinson, "Overview of sph-ale applications for hydraulic turbines in andritz hydro," *Journal of Hydrodynamics*, vol. 30, no. 1, pp. 114–121, 2018.

[41] M. Russell, A. Souto-Iglesias, and T. Zohdi, "Numerical simulation of laser fusion additive manufacturing processes using the sph method," *Computer Methods in Applied Mechanics and Engineering*, vol. 341, pp. 163–187, 2018.

[42] M. S. Shadloo, A. Zainali, M. Yildiz, and A. Suleman, "A robust weakly compressible sph method and its comparison with an incompressible sph," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 8, pp. 939–956, 2012.

[43] A. Skillen, S. Lind, P. K. Stansby, and B. D. Rogers, "Incompressible smoothed particle hydrodynamics (sph) with reduced temporal noise and generalised fickian smoothing applied to body–water slam and efficient wave–body interaction," *Computer Methods in Applied Mechanics and Engineering*, vol. 265, pp. 163–173, 2013.

[44] D. Sulsky, Z. Chen, and H. L. Schreyer, "A particle method for history-dependent materials," *Computer methods in applied mechanics and engineering*, vol. 118, no. 1-2, pp. 179–196, 1994.

[45] P.-N. Sun, D. Le Touzé, G. Oger, and A.-M. Zhang, "An accurate sph volume adaptive scheme for modeling strongly-compressible multiphase flows. part 1: Numerical scheme and validations with basic 1d and 2d benchmarks," *Journal of Computational Physics*, vol. 426, p. 109 937, 2021.

[46] P. Sun, A. Colagrossi, S. Marrone, M. Antuono, and A.-M. Zhang, "A consistent approach to particle shifting in the δ-plus-sph model," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 912–934, 2019.

[47] P. Sun, A. Colagrossi, S. Marrone, and A. Zhang, "The δplus-sph model: Simple procedures for a further improvement of the sph scheme," *Computer Methods in Applied Mechanics and Engineering*, vol. 315, pp. 25–49, 2017.

[48] N. Tsuruta, A. Khayyer, and H. Gotoh, "A short note on dynamic stabilization of moving particle semi-implicit method," *Computers & Fluids*, vol. 82, pp. 158–164, 2013.

[49] R. Vacondio and B. Rogers, "Consistent iterative shifting for sph methods," 2017.

[50] R. Vacondio, C. Altomare, M. De Leffe, X. Hu, D. Le Touzé, S. Lind, J.-C. Marongiu, S. Marrone, B. D. Rogers, and A. Souto-Iglesias, "Grand challenges for smoothed particle hydrodynamics numerical schemes," *Computational Particle Mechanics*, pp. 1–14, 2020.

[51] R. Vacondio, B. Rogers, P. K. Stansby, and P. Mignosa, "Variable resolution for sph in three dimensions: Towards optimal splitting and coalescing for dynamic adaptivity," *Computer Methods in Applied Mechanics and Engineering*, vol. 300, pp. 442–460, 2016.

[52] R. Vacondio, B. Rogers, P. K. Stansby, P. Mignosa, and J. Feldman, "Variable resolution for sph: A dynamic particle coalescing and splitting scheme," *Computer Methods in Applied Mechanics and Engineering*, vol. 256, pp. 132–148, 2013.

[53] B. Van Leer, "Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method," *Journal of computational Physics*, vol. 32, no. 1, pp. 101–136, 1979.

[54] T. Verbrugghe, V. Stratigaki, C. Altomare, J. Domınguez, P. Troch, and A. Kortenhaus, "Implementation of open boundaries within a two-way coupled sph model to simulate nonlinear wave–structure interactions," *Energies*, vol. 12, no. 4, p. 697, 2019.

[55] J. Vila, "On particle weighted methods and smooth particle hydrodynamics," *Mathematical models and methods in applied sciences*, vol. 9, no. 02, pp. 161–209, 1999.

[56] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.

[57] J. Wurster and M. R. Bate, "Disc formation and fragmentation using radiative non-ideal magnetohydrodynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 486, no. 2, pp. 2587–2603, 2019.

[58] R. Xu, P. Stansby, and D. Laurence, "Accuracy and stability in incompressible sph (isph) based on the projection method and a new approach," *Journal of computational Physics*, vol. 228, no. 18, pp. 6703–6725, 2009.

[59] E. Yang, H. H. Bui, H. De Sterck, G. D. Nguyen, and A. Bouazza, "A scalable parallel computing sph framework for predictions of geophysical granular flows," *Computers and Geotechnics*, vol. 121, p. 103 474, 2020.

[60] V. Zago, G. Bilotta, A. Hérault, R. A. Dalrymple, L. Fortuna, A. Cappello, G. Ganci, and C. Del Negro, "Semi-implicit 3d sph on gpu for lava flows," *Journal of Computational Physics*, vol. 375, pp. 854–870, 2018.

[61] E. H. Zubeldia, G. Fourtakas, B. D. Rogers, and M. M. Farias, "Multi-phase sph model for simulation of erosion and scouring by means of the shields and drucker–prager criteria.," *Advances in Water Resources*, vol. 117, pp. 98–114, 2018.