Diss. ETH No. 19566

Discretization-Corrected PSE Operators For Adaptive Multiresolution Particle Methods

A dissertation submitted to
ETH Zurich

for the degree of
Doctor of Sciences

presented by
Birte Schrader
Dipl.-Ing. Maschinenbau, University of Stuttgart
born on August 17, 1979
citizen of Germany

accepted on the recommendation of
Prof. Ivo F. Sbalzarini
Prof. Patrick Jenny
Prof. Georges-Henri Cottet

2011

Declaration of Authorship

This thesis is a presentation of my own original research work. An exception to this is the work presented in section 5.2.2 and in section 5.3. Section 5.2.2 describes the details of the numerical method introduced in chapter 5. Most of these were devised by my coworker Sylvain Reboux. He also added them to the implementation of the method. They rendered the method robust and compatible. Section 5.3 describes numerical experiments and benchmarks which are necessary to assess the numerical method introduced in chapter 5. These test cases have been chosen by my coworker Sylvain Reboux. He also ran the simulations and postprocessed the results. I include these two sections into my thesis to provide a coherent picture of the method invented.
Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature.
The work was done under the guidance of Professor Ivo F. Sbalzarini at ETH Zurich, Switzerland. I hereby declare that this thesis has not been submitted before to any institution for assessment purposes.

Zurich, February 4, 2011 Birte Schrader

Most of the work presented in sections 2.2 to 2.4 and in chapter 3 has been published [79]. The work presented in sections 2.5 to 2.8 and in chapter 5 is submitted for publication [73].

Abstract

Mathematical models in the form of differential equations can often not be solved analytically, but need to be simulated by means of a numerical method. Particle methods are mesh-free schemes that use independent or pairwise interacting particles to represent the physical properties of a system. This allows for natural adaptivity in complex or deforming geometries. The Lagrangian frame of reference when tracking the particles during a simulation renders particle methods particularly successful in the area of fluid mechanics. Their numerical stability in advection-dominated problems is superior to that of descriptions in an Eulerian frame of reference.

The general particle strength exchange (PSE) operators [31] approximate derivatives on scattered particle locations to any desired order of accuracy. Convergence, however, is limited by the discretization error resulting from an inherent numerical quadrature.

In this thesis, we introduce a consistent discretization correction framework for PSE operators. With this correction, the operators yield the desired rate of convergence for any resolution, both on uniform Cartesian and irregular particle distributions, as well as near boundaries. The discretization correction allows setting the kernel width to arbitrarily small values for constant interparticle spacing. We show that, on uniform Cartesian particle distributions, this leads to a seamless transition from discretization-corrected (DC) PSE operators to classical finite-difference stencils. We further identify relationships between DC PSE operators and operators used in corrected smoothed particle hydrodynamics, reproducing kernel

particle methods, differential reproducing kernel and moving least squares approximations, and vorticity redistribution schemes. We analyze the presented DC PSE operators with respect to their accuracy, rate of convergence, computational efficiency, numerical dispersion, numerical diffusion, and stability and compare to uncorrected (UC) PSE operators and, whenever appropriate, to standard finite-difference stencils. Several benchmarks form the basis for a discussion of the operators.
Benchmarking produces an operator assessment that is highly dependent on the problems considered. We therefore introduce three objective, problem-independent measures for the assessment of operator qualities and demonstrate their use in operator choice and deeper understanding of the influence of an operator's parameters on its properties.
Finally, we formulate and test a numerical method for the solution of transport problems that is based on DC PSE operators and that benefits from their strengths. It unites the concepts of Lagrangian particle methods and self-organization of particles driven by particle-particle interactions. Multiresolution requirements are satisfied by means of a monitor function that is used to scale the particle-particle interactions. This, and the advection of the particles, renders the method naturally adaptive. Particle insertions and removals guarantee dynamic adaptation of the local particle densities and therefore the total number of particles. All computations are local, such that the method is well suited for parallelization on distributed-memory machines.

Kurzfassung

Mathematische Modelle in Form von Differentialgleichungen können häufig nicht analytisch gelöst werden sondern müssen mit Hilfe numerischer Methoden simuliert werden. Partikelmethoden benutzen keine Gitter zur Diskretisierung des Kontinuums, sondern voneinander unabhängige oder paarweise interagierende Partikel um die lokalen physikalischen Systemeigenschaften zu repräsentieren. Dies ermöglicht unter Anderem eine unkomplizierte Anpassung an komplexe oder sich verformende Geometrien. Die Lagrangesche Betrachtungsweise, d.h. das Verfolgen der Partikel während einer Simulation, macht Partikelmethoden besonders auf dem Gebiet der Strömungsmechanik attraktiv. In konvektionsdominierten Problemen weisen Partikelmethoden eine höhere numerische Stabilität auf als Methoden, die auf der Eulerschen, d.h. ortsgebundenen, Betrachtungsweise basieren. Die allgemeinen *particle strength exchange* (PSE) Operatoren nähern mit beliebiger Genauigkeit Ableitungen von Funktionen an, deren Werte nur auf im Raum verstreuten Punkten (Partikeln) gegeben sind. Die Konsistenzordnung wird jedoch durch einen durch numerische Quadratur entstandenen Diskretisierungsfehler herabgesetzt.
In dieser Dissertation führen wir ein konsistentes System zur Diskretisierungskorrektur von PSE Operatoren ein. Mit Hilfe dieser Korrektur erzielen die Operatoren die gewünschte Konsistenzordnung für alle räumlichen Auflösungen, sowohl auf äquidistanten kartesischen und unregelmässigen Partikelverteilungen als auch in der Nähe von Rändern. Die Diskretisierungskorrektur erlaubt es, bei konstantem Partikelabstand die charakteristische Breite des Kerns der Operatoren beliebig schmal zu wählen. Wir

zeigen, dass dies auf kartesischen Partikelverteilungen zu einem nahtlosen Übergang zwischen *discretization-corrected* (DC) PSE Operatoren und klassischen Differenzenquotienten der Finite-Differenzen Verfahren führt. Desweiteren identifizieren wir die Beziehungen zwischen DC PSE Operatoren und Operatoren, die in *corrected smoothed particle hydrodynamics*, *reproducing kernel particle methods*, *differential reproducing kernel* und *moving least squares* Approximationen, sowie in *vorticity redistribution schemes* benutzt werden. Wir untersuchen die vorgestellten DC PSE Operatoren hinsichtlich ihrer Genauigkeit, Konsistenzordnung, Effizienz, numerischer Dispersion, numerischer Diffusion und Stabilität und vergleichen sie sowohl mit den unkorrigierten PSE Operatoren als auch, wenn angemessen, mit herkömmlichen Differenzenquotienten. Wir besprechen die Operatoren auf Basis mehrerer Testprobleme.
Die Beurteilung von Operatoren hängt stark von der Auswahl der Testprobleme ab. Deshalb führen wir drei objektive problemunabhängige Maße zur Beurteilung der Qualität der Operatoren ein. Wir führen ihren Nutzen für die Auswahl geeigneter Operatoren vor und veranschaulichen mit ihrer Hilfe wie die Eigenschaften des Operators von seinen freien Parametern abhängen.
Schließlich formulieren und testen wir eine numerische Methode für die Lösung von Transportgleichungen, die auf DC PSE Operatoren basiert. Sie vereinigt Konzepte Lagrangescher Partikelmethoden mit adaptiver Selbstorganisation der Partikel, die auf paarweisen Interaktionen basiert. Die Anforderungen, die durch das Vorhandensein mehrerer Skalen auftreten, werden mit Hilfe einer Überwachungsfunktion befriedigt, die dazu dient die Interaktionspotentiale zu skalieren. Eine Partikeleinführungs- und entfernungsstrategie garantiert, dass sich sowohl die lokale Partikeldichte als auch die Gesamtzahl Partikel den Anforderungen dynamisch anpasst. Alle Berechnungen sind lokaler Natur, so dass sich die Methode für die Parallelisierung auf *distributed-memory* Mehrprozessorensystemen eignet.

Acknowledgments

Foremost, I thank my supervisor Ivo Sbalzarini. He was always encouraging both questions and ideas, open for discussions and suggestions, and helping when problems arose. While giving me freedom in my daily work as well as in the choice of the focus of my investigations, he was completely supportive and enthusiastic for my research. I very much appreciate this. I enjoyed working in his group and watching it grow and evolve. Setting up a truly interdisciplinary, amiable, and functioning team, he offered us great working conditions. Finally, I appreciate his generosity and flexibility in letting me finish this thesis from home. It would not have been possible otherwise. Thank you!

I also want to thank all of my past and current fellow workers of the MOSAIC group. Each of them contributed to the fact that we had an exceptionally positive group culture. I am feeling very lucky to have had such good-humored office mates, first-rate SOLA co-runners, and valuable discussion partners. Wherever appropriate they may feel included in the plural "we" that I use throughout this thesis. Big thanks go to Christian and Jo, who welcomed me into the group when I joined in autumn 2006. A perfect start. Over and over, I am impressed by the amount of Christian's knowledge ranging from scientific facts published in the fifties (including author names and year) up to Homer's philosophy (no, not the Greek) concerning any given situation. I am always amazed by Jo's capability to get the big picture in talks I am totally lost in. He also seems to have a sixth sense when it comes to detecting problems in new ideas or presented solutions. As he questions everything, discussing hypotheses or

conclusions with him usually proves demanding but worthwhile. Thanks to Jo, however, conversation during lunch always turned to football, giving me some time to relax. I was lucky to have Greg as my first office mate. He combines thorough knowledge both in mathematics and on evolution. He is always helping when approached and possesses exceptional googling skills for finding literature or software gadgets. For years, I then happily worked in the same office as Janick. I like his friendly and attentive way and I appreciated having his proficient advice on algorithms, programming, and life in Switzerland. Furthermore, I thank Omar for his patience in fixing linking problems, cluster maintenance, and handling PPM requests, Ömer for continuing the Maxwell project, and Nela and Oliwia for bringing their liveliness and energy into the group, and Andrea who was never tired of helping in quests of all kinds.
I particularly thank Sylvain Reboux for a productive and enjoyable collaboration. I highly value his understanding of mathematics and fluid mechanics and his composed and witty personality. Without him, this thesis would not be the same.
Special thanks go to Jörg Petrasch who showed to me the offer for this PhD position and encouraged me to apply and to Anthony Leonard who encouraged me to continue my research projects at an early stage.
Finally, I am very grateful to my co-examiners Georges-Henri Cottet and Patrick Jenny for reviewing my thesis. I am feeling honored that they are interested in my work.

Contents

List of Figures

List of Tables

Introduction

Background

Computer simulations have become a powerful tool in science, on a par with theory and experiments. They render hidden dynamics observable and provide full control over all parameters, allowing us to test hypotheses or models. Simulations can also be used for educational purposes and can substitute experiments where performing an experiment is impossible for practical, theoretical, or ethical reasons.
A scientific simulation is the implementation of a model. It is hence of great importance for a simulation to represent the model adequately. In many areas, such as engineering, natural sciences, and economics, models (or fundamental laws) are often formulated as differential equations. Only few differential equations can be solved analytically. As a consequence, scientific simulations are closely linked to the development of numerical methods and to numerical analysis. The goal is to find approximate numerical solutions with guaranteed error bounds at reasonable computational cost. This thesis focusses on the approximation of differential operators in particle methods. Particle methods are mesh-free schemes that use particles to represent the physical properties of a system. This allows for natural adaptivity in complex or deforming geometries. The Lagrangian frame of reference when tracking the particles during a simulation renders particle methods particularly successful in the area of fluid mechanics. Their numerical stability in advection-dominated problems is superior to that of descriptions in an Eulerian frame of reference.

In continuum particle methods the smooth approximation of a function f in the weak formulation is

$$f_\delta^h(\boldsymbol{x},t) = \sum_p \omega_p(t)\zeta_\delta(\boldsymbol{x}-\boldsymbol{x}_p), \tag{1}$$

where the index p runs over all particles, ω_p is the particle weight or strength, $\boldsymbol{x}_p$ is the particle position, and ζ_δ is the mollification kernel or cutoff function with characteristic length δ,

$$\zeta_\delta(\boldsymbol{z}) = \frac{1}{\delta^n}\zeta\left(\frac{\boldsymbol{z}}{\delta}\right),$$

and

$$\int_{\mathbb{R}^n} \zeta(\boldsymbol{z})\mathrm{d}\boldsymbol{z} = 1.$$

Here and in the following, n denotes the number of spatial dimensions. The particle strengths ω_p are usually chosen as $\omega_p = v_p f(\boldsymbol{x}_p)$, where $v_p = h_p^n$ is the particle volume and h_p the local interparticle spacing. With certain continuity requirements on f and ζ, the error $f - f_\delta^h$ is bounded [72].
Consider the transport equation

$$\frac{\partial f}{\partial t} + \nabla\cdot(f\boldsymbol{u}) = g, \tag{2}$$

where $\boldsymbol{u}$ denotes a velocity and g a source term. The evolution of the particle strengths ω_p, the particle volumes v_p, and the particle positions $\boldsymbol{x}_p$ is then defined by

$$\frac{\mathrm{d}\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{x}_p,t), \qquad \frac{\mathrm{d}\omega_p}{\mathrm{d}t} = v_p g\big|_{\boldsymbol{x}_p,t}, \qquad \frac{\mathrm{d}v_p}{\mathrm{d}t} = v_p\nabla\cdot\boldsymbol{u}\big|_{\boldsymbol{x}_p,t}. \tag{3}$$

This can be derived in a straightforward way by taking the material derivatives of the fields $\omega(\boldsymbol{x},t) = v(\boldsymbol{x},t)f(\boldsymbol{x},t)$ and $v(\boldsymbol{x},t)$, where v is the volume per particle, with respect to the velocity field $\boldsymbol{u}$ and substituting the transport equation (2).
In the strong formulation, the function approximation is

$$f_\delta^h(\boldsymbol{x},t) = \sum_p f_p(t)\zeta\left(\frac{\boldsymbol{x}-\boldsymbol{x}_p}{h}\right), \tag{4}$$

and the equations of motion are

$$\frac{\mathrm{d}\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{x}_p, t), \qquad \frac{\mathrm{d}f_p}{\mathrm{d}t} = g\big|_{\boldsymbol{x}_p,t} - f_p \nabla \cdot \boldsymbol{u}\big|_{\boldsymbol{x}_p,t}. \tag{5}$$

For irregular particle distributions, however, equation (4) is a poor approximation of the function f, unless the cutoff function ζ inherently accounts for the inhomogeneous particle spacings.

Independent of whether the weak or strong formulation is used, the approximation of derivatives plays a key role. If the source term g depends on derivatives of the field f, as is for example the case in diffusion with $g = \nabla^2 f$, or if the divergence of $\boldsymbol{u}$ is unknown, field derivatives need to be evaluated at particle locations in order to evolve ω_p and v_p according to equations (3) or f_p according to equations (5). Derivative approximations are hence crucial for solving the governing equations.

Diffusion terms are commonly approximated using particle strength exchange (PSE) operators [48], which were originally designed by Degond and Mas-Gallic [26, 27]. The conservative PSE operators approximate the Laplacian of f at the position of particle p as

$$\nabla^2 f(\boldsymbol{x}_p) \approx \sum_q v_q (f_q - f_p) \eta_\varepsilon(\boldsymbol{x}_p - \boldsymbol{x}_q),$$

where η_ε is an appropriate kernel function, scaled to characteristic width ε.[1] Later, Eldredge and coworkers [31] developed general PSE operators for the approximation of arbitrary differential operators.

Problem Statement and Motivation

For large-enough kernel widths ε the derivative approximation using PSE operators converges with $O(\varepsilon^r)$. The consistency of PSE operators, however, is limited by a discretization error resulting from numerical quadrature over the particle locations. This discretization error is constant or of negative order and dominates for small kernel widths ε. It can be reduced only by increasing the number of particles in the support of the scaled kernel function η_ε, resulting in the *overlap condition* $h < a\varepsilon^q$, $q > 1$. The

[1] The conservative derivative approximation in SPH [48, equation (24)] can be considered a special case of a PSE operator with $\varepsilon = \delta = h$ and $\eta = \nabla^2 \zeta$.

resulting approximation is then of order $O(h^{r/q})$.
In order to obtain a consistent $O(h^r)$-approximation, that is to reduce the total number of particles required to reach a given level of accuracy, a discretization correction can be performed. Several variants of discretization-corrected (DC) PSE operators have been used in state-of-the-art simulations [9, 39, 43, 70, 76]. However, while the numerical properties of the original, uncorrected (UC) PSE operators are well known,

(i) an analysis of DC PSE operators and

(ii) a classification of their position in the field of differential operators

are missing. Also the key question of

(iii) whether the additional computational cost for the discretization correction is amortized by the gain in accuracy

is yet unanswered. In case of a positive answer, it should also be interesting to investigate the use of DC PSE operators

(iv) in multiresolution simulations, and

(v) in Lagrangian particle methods for the strong formulation.

Point (iv) is of particular interest as particle methods are increasingly used for multiscale simulations. If we find DC PSE operators to be valuable in multiresolution problems, they could be used in a straightforward way to compute derivatives on multiresolution particle configurations and to interpolate the particle properties onto a mesh, e.g., for efficiently solving the Poisson equation. This could simplify the design of accurate and efficient multiresolution particle methods.
Why is it interesting to consider point (v)? Lagrangian particle methods based on the weak formulation of the field approximation are normally more attractive than those based on the strong formulation because they are robust in handling field discontinuities. However, dynamic particle management, including particle insertion and removal, is simplified by the strong formulation since particle volumes are disregarded. Dynamic particle management is particularly advantageous for fast evolving fields with multiple scales; where small scales appear, particles are inserted, where small structures are dissipated, particles can be removed. As nonuniform particle distributions usually deteriorate the accuracy of the strong field approximations for UC cutoff functions ζ, discretization correction is expected to be beneficial.

Outline

In this thesis, we address the five questions presented above. In addition, we introduce performance measures for diffusion operators, providing a useful tool for developing a deeper understanding of the role of the tunable operator parameters. We demonstrate that the measures are useful for the selection of efficient operators.
This thesis is structured as follows:

Chapter 1 We briefly review PSE operators as a class of discretized integral operators that can be applied to scattered data in order to approximate derivatives.

Chapter 2 We present a discretization-correction framework for PSE operators that renders the operators pointwise consistent while keeping the number of particles required for operator evaluation constant. We analyze the impact of discretization correction on operator properties such as accuracy, numerical diffusion and dissipation, and stability in explicit time stepping. Some fundamental tests are presented to compare basic features of the corrected operators to those of the original operators and standard finite-difference stencils. The performance of these three types of operators in benchmark simulations forms the basis for an assessment of DC PSE operators.

Chapter 3 Various numerical methods provide a large selection of operators for approximating field derivatives. For certain parameter choices DC PSE operators become equivalent or similar to other operators. Aiming for a unifying description, we discuss the relations to operators used in finite-difference methods, corrected smoothed particle hydrodynamics, reproducing kernel particle methods, differential reproducing kernel and moving least squares approximations, and vorticity redistribution schemes. We prove that certain classical finite-difference stencils are special cases of DC PSE operators.

Chapter 4 Diffusion is a fundamental transport phenomenon. Diffusion operators hence often play a key role in numerical simulations. Besides PSE operators, there are many other possible operators for approximating a diffusion term in particle methods. Which one is best suited for a

given problem? How big are the differences? How does a certain parameter influence the properties of a specific operator? These questions, if asked, are usually tackled by testing a set of different operators on a set of benchmark problems. The answers then naturally depend on the choice of the benchmark problems and conclusions outside the scope of the test problems are speculative. We therefore present three measures that enable an predictive rating of operator properties. All presented measures are by design independent of the problem to be solved. We illustrate their potential and limitations in a parameter study and a case study.

Chapter 5 We introduce a self-organizing adaptive multiresolution particle method for solving transport problems in a Lagrangian frame of reference. It combines the idea of Lagrangian particle methods with that of particle self-organization from particle-particle interaction potentials. This allows for natural adaptation to multiresolution fields in the strong formulation of the transport problem. The discretization correction presented and assessed in chapter 2 enables straightforward and accurate simulations. Based on the solution of benchmark problems we compare the presented method to a remeshed Lagrangian particle method and a multiresolution wavelet particle method.

Chapter 6 We close this thesis by summarizing its results and conclusions and by putting forward ideas for subsequent work.

CHAPTER

ONE

Particle Strength Exchange Operators

1.1 Introduction

In Lagrangian particle methods, such as smoothed particle hydrodynamics (SPH) or vortex methods, continuous flow fields are discretized over scattered particle locations. The particles follow the flow in a Lagrangian way. They need not satisfy any connectivity constraints. That is, they are not required to form a structured or unstructured mesh. In order to accurately and efficiently approximate the viscosity or diffusion term in vortex methods for viscous flow, particle strength exchange (PSE) operators were introduced by Degond and Mas-Gallic 1989. They can handle both isotropic [27] and anisotropic diffusion [26].

The PSE operators were preceded by methods specifically designed to simulate diffusion in vortex methods. Kuwahara and Takami [50] account for viscous effects by spreading (or smoothing) the vorticity when computing the velocity. They point out that – due to superposition of vortices – their "equations are not strictly exact because they are incompatible with the Navier-Stokes equations". Thus, they state that they introduce an artificial viscosity, not the real one. In the random walk method, introduced by Chorin [17], the particles (vortex positions) are displaced by Gaussian random steps in order to simulate diffusion. The dynamics of the numerical particles hence reflect the dynamics of Brownian motion. After convergence of viscous splitting methods for the Navier-Stokes equations had

been proven by Beale and Majda [5], convergence of the random walk vortex method was proven by Marchioro and Pulvirenti [60], Goodman [40], and Long [58]. Nevertheless, a large amount of noise is introduced by the random walk such that large numbers of particles are required for high accuracy [34]. The core spreading (or core expanding) algorithm – a deterministic way of simulating diffusion – as described in Leonard's review [55] has been shown to approximate the wrong equations [41]. A consistent deterministic displacement scheme for simulating diffusion was introduced by Degond and Mustieles [28] one year after the PSE operators were published. They define an equivalent convection velocity for the diffusion process, but remark that this method may be less accurate and more expensive than other methods for the Navier-Stokes and the heat equations and may be more suited for problems in kinetic theory of plasma physics. Another deterministic approach to simulating the viscous term in vortex methods is the exchange of vorticity between particles or the redistribution of vorticity among particles. Among others, the works of Raviart [72], Choquin and Huberson [15], Cottet and Mas-Gallic [24], and Fishelov [33] follow this approach. In some methods, the quantity carried by the particles is interpolated onto a mesh at each time step after the particles have been advected. The diffusion term – or the amount of strength exchanged – may then be computed on the mesh using classical finite-difference stencils as done, for example, by Bergdorf and coworkers [9]. The advantage of this approach is that the particles do not form clusters or holes in the course of the simulation. Also, the construction of neighbor lists becomes unnecessary. The interpolation procedure, also called remeshing, can be combined with the diffusion step of the simulation [85]. In Free Lagrangian Methods [36], a finite difference scheme is constructed for the (irregular) particle positions using a Voronoi diagram. The PSE operators were developed following the idea of exchanging vorticity between particles. Thus, using PSE operators particles are neither displaced nor do their volumes change in order to represent diffusion. The amount of strength they exchange is based on an approximation of the diffusive flux, while the total amount of vorticity is conserved. The resulting method has been shown to be superior to the random walk method [16]. In 2002, Eldredge and coworkers generalized PSE operators to estimate any spatial derivative of a field carried by particles [31].

The classical PSE operators are briefly described in §1.3 and the generalized operators are recollected in §1.4. We begin by recapitulating the

multiindex notation in §1.2, which will be used throughout this thesis.

1.2 Multiindex Notation

For compactness of notation, we use the following multiindex notation: for a multiindex $\boldsymbol{\alpha}$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, and a vector $\boldsymbol{x} = (x_1, \ldots, x_n)^{\mathrm{T}} \in \mathbb{R}^n$,

$$|\boldsymbol{\alpha}| = \sum_{i=1}^{n} \alpha_i, \qquad \boldsymbol{x}^{\boldsymbol{\alpha}} = \prod_{i=1}^{n} x_i^{\alpha_i}, \qquad \boldsymbol{\alpha}! = \prod_{i=1}^{n} \alpha_i!,$$

and a sum over all indices $\boldsymbol{\alpha}$ for which $|\boldsymbol{\alpha}| = k$ is written as $\sum_{|\boldsymbol{\alpha}|=k}$. Derivatives of a multivariate field $f(\boldsymbol{x}) \in \mathbb{R}$, $\boldsymbol{x} \in \mathbb{R}^n$, are expressed as

$$D^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{\partial^{|\boldsymbol{\beta}|} f(\boldsymbol{x})}{\partial x_1^{\beta_1} \partial x_2^{\beta_2} \ldots \partial x_n^{\beta_n}}.$$

1.3 The Classical PSE Operators for Diffusion

Degond and Mas-Gallic [26, 27] designed PSE operators for both the case of isotropic and anisotropic diffusion. In the case of isotropic diffusion, the approximated diffusion operator $L(\cdot)$ is defined by

$$Lf(\boldsymbol{x}) = \nabla \cdot (\nu(\boldsymbol{x}) \nabla f(\boldsymbol{x})) \;=\; \sum_{|\boldsymbol{\alpha}|=1} \left[D^{\boldsymbol{\alpha}} \nu(\boldsymbol{x}) D^{\boldsymbol{\alpha}} f(\boldsymbol{x}) + \nu(\boldsymbol{x}) D^{2\boldsymbol{\alpha}} f(\boldsymbol{x}) \right]. \tag{1.1}$$

Here, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_n) \in \mathbb{N}^n$ is a multiindex, n is the number of spatial dimensions, f is the field to be diffused, and ν is the viscosity (or diffusion coefficient). In the case of anisotropic viscosity, the viscosity field is represented by the positive symmetric tensor $\boldsymbol{\nu} \in \mathbb{R}^{n \times n}$. The diffusion operator $L(\cdot)$ is then defined by

$$Lf(\boldsymbol{x}) = \nabla \cdot (\boldsymbol{\nu}(\boldsymbol{x}) \nabla f(\boldsymbol{x})) \;=\; \sum_{i,j=1}^{n} D^{\boldsymbol{e}_i} \left[\nu_{ij}(\boldsymbol{x}) D^{\boldsymbol{e}_j} f(\boldsymbol{x}) \right] \tag{1.2}$$

where ν_{ij} are the elements of the matrix $\boldsymbol{\nu}$, and $\boldsymbol{e}_i$ is the unit vector along dimension i. The isotropic and the anisotropic case are discussed separately below.

1.3.1 The Case of Isotropic Diffusion

The PSE operators $Q^h(\cdot)$ approximate the isotropic diffusion operator $L(\cdot)$ using the information about the field $f(\boldsymbol{x})$ and the viscosity $\nu(\boldsymbol{x})$ carried by particles of characteristic spacing h.
The PSE operator for isotropic diffusion is defined as

$$Q_h f(\boldsymbol{x}) = \frac{1}{\varepsilon^2} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(f(\boldsymbol{x}_p) - f(\boldsymbol{x}) \right) \mu(\boldsymbol{x}, \boldsymbol{x}_p) \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p), \tag{1.3}$$

where $\boldsymbol{x}_p$ and v_p are the position and the volume of particle p, respectively, and $\mathcal{N}(\boldsymbol{x})$ comprises all particles contained in an r_c-neighborhood around $\boldsymbol{x}$. The cutoff radius r_c is typically chosen such that the neighborhood $\mathcal{N}(\boldsymbol{0})$ coincides (with a certain accuracy) with the support of the scaled kernel function

$$\eta_\varepsilon(\boldsymbol{z}) = \varepsilon^{-n} \eta(\boldsymbol{z}/\varepsilon),$$

where ε is the kernel width.
For conservation of mass, $\mu(\boldsymbol{x}, \boldsymbol{y})$ is set to a symmetric mean of $\nu(\boldsymbol{x})$ and $\nu(\boldsymbol{y})$, for example, $\mu(\boldsymbol{x}, \boldsymbol{y}) = (\nu(\boldsymbol{x}) + \nu(\boldsymbol{y}))/2$. The kernel function η is chosen such that the moment conditions

$$\begin{aligned} Z^{\boldsymbol{\alpha}} &= Y^{\boldsymbol{\alpha}}, \ 1 \le |\boldsymbol{\alpha}| \le r+1, \\ Z^{\boldsymbol{\alpha}*} &< \infty, \ |\boldsymbol{\alpha}| = r+2, \end{aligned} \tag{1.4}$$

where

$$\begin{aligned} Z^{\boldsymbol{\alpha}} &= \int_{\mathbb{R}^n} \boldsymbol{z}^{\boldsymbol{\alpha}} \eta(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = \int_{\mathbb{R}^n} \left(\frac{\boldsymbol{z}}{\varepsilon} \right)^{\boldsymbol{\alpha}} \eta_\varepsilon(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}, \\ Z^{\boldsymbol{\alpha}*} &= \int_{\mathbb{R}^n} |\boldsymbol{z}^{\boldsymbol{\alpha}} \eta(\boldsymbol{z})| \, \mathrm{d}\boldsymbol{z} = \int_{\mathbb{R}^n} \left| \left(\frac{\boldsymbol{z}}{\varepsilon} \right)^{\boldsymbol{\alpha}} \eta_\varepsilon(\boldsymbol{z}) \right| \mathrm{d}\boldsymbol{z}, \\ Y^{\boldsymbol{\alpha}} &= \begin{cases} 2, & \boldsymbol{\alpha} = 2\boldsymbol{e}_i, i = 1, \ldots, n, \\ 0, & \text{else}, \end{cases} \end{aligned} \tag{1.5}$$

are satisfied.
The error $\epsilon(\boldsymbol{x}) = Q_h f(\boldsymbol{x}) - Lf(\boldsymbol{x})$ can be split into two parts and bounded as follows:

1. The *mollification error* ϵ_ε is the difference between the continuous integral operator $Q(\cdot)$ and the exact operator $L(\cdot)$ applied on f:

$$\begin{aligned}\epsilon_\varepsilon(\boldsymbol{x}) &= Qf(\boldsymbol{x}) - Lf(\boldsymbol{x}) \\ &= \frac{1}{\varepsilon^2}\int_{\mathbb{R}^n} \left(f(\boldsymbol{y}) - f(\boldsymbol{x})\right)\mu(\boldsymbol{x},\boldsymbol{y})\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\mathrm{d}\boldsymbol{y} \\ &\quad - \nabla\cdot\left(\nu(\boldsymbol{x})\nabla f(\boldsymbol{x})\right).\end{aligned}$$

   Given that the moment conditions (1.4) are satisfied, the mollification error can be bounded as

$$|\epsilon_\varepsilon(\boldsymbol{x})| \leq C\varepsilon^r,$$

   where C includes the norms $\|f\|_{r+2,\infty,\mathbb{R}^n}$, $\|\nu\|_{r+1,\infty,\mathbb{R}^n}$, and the modified moments $Z^{\boldsymbol{\alpha}*}$, $|\boldsymbol{\alpha}| = r+2$, as defined in (1.5). This result is derived by Taylor-expanding $f(\boldsymbol{y})$ and $\mu(\boldsymbol{x},\boldsymbol{y})$ around $\boldsymbol{x}$.

2. The *saturation error* or *discretization error* ϵ_h represents the difference between the discrete operator $Q_h(\cdot)$ defined in equation (1.3) and its continuous form $Q(\cdot)$:

$$\begin{aligned}\epsilon_h(\boldsymbol{x}) &= Q_h f(\boldsymbol{x}) - Qf(\boldsymbol{x}) \\ &= \frac{1}{\varepsilon^2}\sum_{p\in\mathcal{N}(\boldsymbol{x})} v_p\left(f(\boldsymbol{x}_p) - f(\boldsymbol{x})\right)\mu(\boldsymbol{x},\boldsymbol{x}_p)\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p) \\ &\quad - \frac{1}{\varepsilon^2}\int_{\mathbb{R}^n}\left(f(\boldsymbol{y}) - f(\boldsymbol{x})\right)\mu(\boldsymbol{x},\boldsymbol{y})\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\mathrm{d}\boldsymbol{y}.\end{aligned}$$

   The error can be bounded as

$$|\epsilon_h(\boldsymbol{x})| \leq C\frac{h^m}{\varepsilon^{m+1}}$$

   for $f,\nu \in W^{m,p}(\mathbb{R}^n)$ and a kernel function $\eta \in W^{m,1}(\mathbb{R}^n)$, $m > n$, whose support when centered at $\boldsymbol{x}$ should be covered by the particles

in the neighborhood $\mathcal{N}(\boldsymbol{x})$.[1] The constant C depends on f and ν as well as on η and m, more specifically on the norm $\|\eta\|_{m,1,\mathbb{R}^n}$. Letting h or ε tend to zero, the discretization error tends to zero if the ratio $c = h/\varepsilon$ tends to zero as well – faster than $h^{1/(m+1)}$ or $\varepsilon^{1/m}$. To shrink the discretization error with the mollification error as ε^r, the ratio c needs to be decreased as $h^{(r+1)/(m+r+1)}$ or $\varepsilon^{(r+1)/m}$. Keeping the ratio $c = h/\varepsilon$ constant, the error bound grows with decreasing kernel width ε (or interparticle spacing h).[2] Obviously, whenever the support of the kernel function is not covered by the neighborhood around $\boldsymbol{x}$, an additional, constant error occurs when approximating $Q(\cdot)$ by $Q_h(\cdot)$. This error depends on r_c/ε as well as on f and ν. It can be reduced to the desired accuracy by increasing r_c/ε.

1.3.2 The Case of Anisotropic Diffusion

The anisotropic diffusion operator $L(\cdot)$, defined in equation (1.2), is approximated by the PSE operator $Q^h(\cdot)$ defined by

$$Q_h f(\boldsymbol{x}) = \frac{1}{\varepsilon^4} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(f(\boldsymbol{x}_p) - f(\boldsymbol{x})\right) \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \times \sum_{i,j=1}^{n} \mu_{ij}(\boldsymbol{x}, \boldsymbol{x}_p)(\boldsymbol{x} - \boldsymbol{x}_p)^{\boldsymbol{e}_i + \boldsymbol{e}_j},$$

[1] We denote by $W^{m,p}(\Omega)$ the Sobolev space $W^{m,p}(\Omega) = \{u \in L^p(\Omega);\, D^{|\boldsymbol{\alpha}|}u \in L^p(\Omega),\, \forall \boldsymbol{\alpha} \in \mathbb{N}^n,\, |\boldsymbol{\alpha}| \leq m\}$. We provide $W^{m,p}(\Omega)$ with the norm

$$\|u\|_{m,p,\Omega} = \left(\sum_{|\boldsymbol{\alpha}| \leq m} \|D^{\boldsymbol{\alpha}} u\|^p_{L^p(\Omega)} \right)^{1/p}.$$

[2] We remark that the commonly used and infinitely often differentiable Gaussian kernel function is not part of $W^{\infty,1}(\mathbb{R}^n)$ since its derivatives $D^{\boldsymbol{\alpha}}\eta$, $|\boldsymbol{\alpha}| = \infty$, do not belong to the space $L^1(\mathbb{R}^n)$. Nevertheless, for the Gaussian kernel function, $\eta \in W^{m,1}(\mathbb{R}^n)$ is true for an arbitrarily large integer m. The value of $\|\eta\|_{m,1,\mathbb{R}^n}$, however, grows faster than exponentially for large m; the bound Cc^m for large m therefore grows as we let m grow. Settling for a smaller integer m will result in a tighter bound for the discretization error.

where

$$\boldsymbol{\mu}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2}\left[\boldsymbol{\nu}(\boldsymbol{x}) + \boldsymbol{\nu}(\boldsymbol{y}) - \frac{\mathrm{tr}(\boldsymbol{\nu}(\boldsymbol{x}) + \boldsymbol{\nu}(\boldsymbol{y}))}{n+2}\boldsymbol{I}\right]$$

and $\boldsymbol{I}$ is the identity matrix in $\mathbb{R}^{n\times n}$.
The conditions on the kernel function η then are

$$\begin{aligned} Z^{\boldsymbol{\alpha}} &= Y^{\boldsymbol{\alpha}},\ 3 \le |\boldsymbol{\alpha}| \le r+3, \\ Z^{\boldsymbol{\alpha}*} &< \infty,\ |\boldsymbol{\alpha}| = r+4, \end{aligned}$$

where

$$Y^{\boldsymbol{\alpha}} = \begin{cases} 1, & \boldsymbol{\alpha} = 2\boldsymbol{e}_i + 2\boldsymbol{e}_j,\ i,j = 1,\ldots,n,\ i \ne j, \\ 3, & \boldsymbol{\alpha} = 4\boldsymbol{e}_i, i = 1,\ldots,n, \\ 0, & \text{else.} \end{cases}$$

The error analysis is identical to the one given for the isotropic operator in §1.3.1.

1.4 The General PSE Operators

For the construction of general PSE operators, Eldredge and coworkers [31] approximate any spatial derivative $D^{\boldsymbol{\beta}} f(\boldsymbol{x})$ of a (sufficiently smooth) field f by an integral operator $Q^{\boldsymbol{\beta}}(\cdot)$ defined by

$$Q^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{1}{\varepsilon^{|\boldsymbol{\beta}|}} \int_{\mathbb{R}^n} \left(f(\boldsymbol{y}) \pm f(\boldsymbol{x})\right) \eta_\varepsilon^{\boldsymbol{\beta}}(\boldsymbol{x}-\boldsymbol{y}) \mathrm{d}\boldsymbol{y}. \tag{1.6}$$

The scaled operator kernel

$$\eta_\varepsilon^{\boldsymbol{\beta}}(\boldsymbol{z}) = \frac{1}{\varepsilon^n} \eta^{\boldsymbol{\beta}}\left(\frac{\boldsymbol{z}}{\varepsilon}\right)$$

is chosen such as to fulfill the continuous moment conditions

$$Z^{\boldsymbol{\alpha}} = Y^{\boldsymbol{\alpha}},\ \alpha_{\min} \le |\boldsymbol{\alpha}| \le r + |\boldsymbol{\beta}| - 1, \tag{1.7}$$

$$Z^{\boldsymbol{\alpha}*} < \infty,\ |\boldsymbol{\alpha}| = r + |\boldsymbol{\beta}|, \tag{1.8}$$

where

$$Y^{\boldsymbol{\alpha}} = \begin{cases} (-1)^{|\boldsymbol{\beta}|}\boldsymbol{\beta}!, & \boldsymbol{\alpha} = \boldsymbol{\beta}, \\ 0, & \text{else}, \end{cases} \qquad \alpha_{\min} = \begin{cases} 0, & |\boldsymbol{\beta}| \text{ odd}, \\ 1, & |\boldsymbol{\beta}| \text{ even}. \end{cases} \tag{1.9}$$

The sign in equation (1.6) is chosen positive for odd $|\boldsymbol{\beta}|$ and negative for even $|\boldsymbol{\beta}|$.
The integral operator defined in (1.6) is then discretized by midpoint quadrature over the particles, thus,

$$Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{1}{\varepsilon^{|\boldsymbol{\beta}|}} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(f(\boldsymbol{x}_p) \pm f(\boldsymbol{x}) \right) \eta_\varepsilon^{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{x}_p). \tag{1.10}$$

The error $\varepsilon(\boldsymbol{x}) = Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) - Lf(\boldsymbol{x})$ can again be split into the mollification error and the discretization error as described for the original PSE operator in §1.3.1. The discretization error is bounded by $Ch^m/\varepsilon^{m+|\boldsymbol{\beta}|-1}$ for even $|\boldsymbol{\beta}|$ and $Ch^m/\varepsilon^{m+|\boldsymbol{\beta}|}$ for odd $|\boldsymbol{\beta}|$.

CHAPTER
TWO

Discretization Correction of PSE Operators

2.1 Introduction

Both the original and the generalized PSE operators involve two numerical approximation errors: the mollification error and the discretization error. The mollification error is of order ε^r, where ε is the kernel width and the exponent r is defined by the kernel design. The discretization error from the numerical quadrature over the particle locations can be reduced by reducing the interparticle spacing h faster than the kernel width ε. If the ratio $c = h/\varepsilon$ is kept constant, the discretization error limits the convergence as depicted in figure 2.1. In order for the operator to be consistent, h and ε have to satisfy the condition $c = h/\varepsilon \to 0$ as h and ε tend to zero [20, 72]. This leads to an *overlap condition* of the type $h \leq a\varepsilon^q$, $0 < a < 1$. The value of q depends on the desired order of accuracy r. It is, however, always greater than one. Highly accurate approximations thus typically require large numbers of particles, both in total and in the operator support: as the kernel width ε is decreased, the interparticle spacing needs to be increased, causing the total number of particles in the simulation to grow as ε^{-qn} in $\mathbb{R}^n$. Simultaneously, the number of particles in the operator support grows as ε^{1-q}. The computational cost for evaluating the operator at all particle positions therefore roughly grows

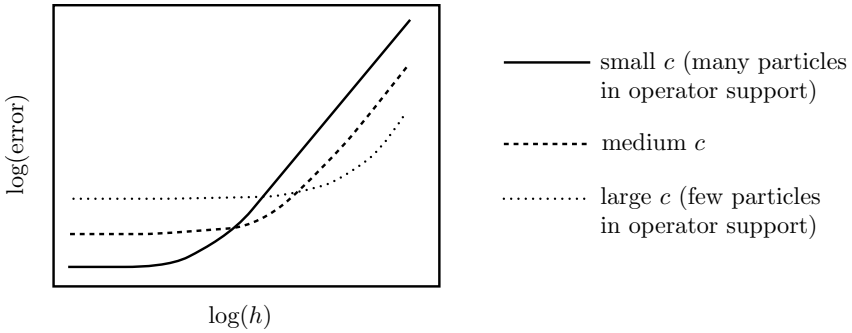

Figure 2.1: Cartoon convergence plots for PSE operators.

as $\varepsilon^{-n\left(q+\frac{q-1}{n}\right)}$. If one could keep the ratio c constant, it would only grow as ε^{-n}.

Discretization correction of the PSE operators aims at removing the discretization error. The removal renders the overlap condition unnecessary. Then, r^{th}-order accuracy is attained for any ratio c. The idea of discretization correction is to replace the continuous moment conditions – equations (1.8) for the general PSE operators – by corresponding discrete moment conditions. Using such discretization-corrected (DC) kernels ensures convergence over the entire range of resolutions.

To the best of our knowledge, discretization correction for particle methods was first described by Cottet and coworkers [21] for interpolation kernels. Shankar and van Dommelen [80] presented a DC redistribution scheme which handles the viscous term in vortex methods. DC PSE operators have been used in many state-of-the-art simulations: Hieber and Koumoutsakos [43], for example, used it for a second-order approximation of the Laplacian on symmetric particle distributions. Bergdorf and coworkers [9] pointed out the possibility of discretization correction of the anisotropic diffusion operator derived by Degond and Mas-Gallic [26]. Sbalzarini and coworkers used DC PSE operators to approximate the Laplacian with second-order accuracy [76] on uniform Cartesian particle distributions. Poncet [70] used the original and the DC anisotropic diffusion operators in vortical ring simulations and compared the results to classical finite difference (FD) stencils. His corrected operator is of order one on arbi-

trary particle distributions and of order two on symmetric ones. Golia and coworkers [39] formulated two different discretization corrections for PSE operators to estimate the gradient and the Laplacian of a field. Their corrected operators guarantee second-order accuracy on symmetric particle distributions and first or zeroth order on arbitrary particle distributions. Some authors avoid the computational overhead of discretization correction by reinitializing the particles on a uniform Cartesian grid (a procedure called *remeshing*) at each time step, as for example Bergdorf and coworkers [9], or every few time steps, as for example Koumoutsakos [47]. Doing so, the correction has to be determined only once at the very beginning of the simulation (or beforehand) and can be reused from then on. Other authors combined diffusion and remeshing into a single kernel [85] to further reduce the computational cost.

In all of these previous applications of DC PSE operators, however, little attention has been paid to their numerical properties. Even though the uncorrected (UC) PSE operators have been analyzed thoroughly by their inventors, general analysis of DC PSE operators was missing in the literature. Also, it was unclear whether the additional computational cost for discretization-correcting the operators is amortized by their gain in accuracy.

In this chapter, we present a formal framework for DC PSE operators. This enables us to derive expressions for their overall approximation errors. We perform a full stability analysis based on the dispersive and diffusive properties of the operators and show the convergence rates and computational efficiencies of the operators on several test problems. Pointing out similarities and differences between DC and UC operators in all cases shows that a separate analysis of the DC operators is valuable. Accommodating Lagrangian particle methods with and without remeshing, our analysis considers particles distributed both irregularly and on uniform Cartesian grids. We also discuss boundary effects. Providing helpful information for the potential user, the influences of the free operator parameters (ratio $c = h/\varepsilon$, cutoff radius r_c, and order of accuracy r) on the computational efficiency and the operators' numerical properties are discussed in all cases. We address the critical point of computational efficiency when solving advection-diffusion problems of various Péclet numbers with Lagrangian schemes using DC PSE operators, Lagrangian schemes using UC PSE operators, Lagrangian FD schemes, and Eulerian FD schemes. We discuss the consistency of DC PSE operators in multiresolution methods

and methods based on strong formulations. Finally, we highlight how the discretization correction of a DC PSE operator can be reused for the approximation of other types of derivatives or functions. We show how to modify these function approximation such that they are interpolating.

2.2 Discretization Correction

Using the PSE operator (1.10) to evaluate the derivative $D^{\boldsymbol{\beta}} f(\boldsymbol{x})$ involves two approximations: the mollification error $Q^{\boldsymbol{\beta}} f(\boldsymbol{x}) - D^{\boldsymbol{\beta}} f(\boldsymbol{x})$ and the discretization error $Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) - Q^{\boldsymbol{\beta}} f(\boldsymbol{x})$. These two error terms are usually treated separately. Here, we establish the idea of discretization correction by directly considering the overall error $\epsilon(\boldsymbol{x}) = Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) - D^{\boldsymbol{\beta}} f(\boldsymbol{x})$. An expression for this error can be derived by expanding the field f in the operator definition (1.10) into a Taylor series around $\boldsymbol{x}$ and subtracting $D^{\boldsymbol{\beta}} f(\boldsymbol{x})$:

$$\epsilon(\boldsymbol{x}) = \left(\frac{(-1)^{|\boldsymbol{\beta}|}}{\boldsymbol{\beta}!} Z_h^{\boldsymbol{\beta}}(\boldsymbol{x}) - 1 \right) D^{\boldsymbol{\beta}} f(\boldsymbol{x}) + \sum_{\substack{|\boldsymbol{\alpha}|=1 \\ \boldsymbol{\alpha} \neq \boldsymbol{\beta}}}^{\infty} \frac{(-1)^{|\boldsymbol{\alpha}|}}{\boldsymbol{\alpha}!} \varepsilon^{|\boldsymbol{\alpha}|-|\boldsymbol{\beta}|} Z_h^{\boldsymbol{\alpha}}(\boldsymbol{x}) D^{\boldsymbol{\alpha}} f(\boldsymbol{x}) + \epsilon_0 \,, \quad (2.1)$$

with

$$\epsilon_0 = \begin{cases} 2\varepsilon^{-|\boldsymbol{\beta}|} Z_h^{\mathbf{0}}(\boldsymbol{x}) f(\boldsymbol{x}) \,, & |\boldsymbol{\beta}| \text{ odd} \,, \\ 0 \,, & |\boldsymbol{\beta}| \text{ even} \,, \end{cases} \quad (2.2)$$

and the discrete moments $Z_h^{\boldsymbol{\alpha}}$ defined as

$$Z_h^{\boldsymbol{\alpha}}(\boldsymbol{x}) = \frac{1}{\varepsilon^n} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon} \right)^{\boldsymbol{\alpha}} \eta^{\boldsymbol{\beta}} \left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon} \right) . \quad (2.3)$$

Note that the definition of the discrete moments $Z_h^{\boldsymbol{\alpha}}$ corresponds to the definition (1.5) of the continuous ones $Z^{\boldsymbol{\alpha}}$; the term $\int_{\mathbb{R}^n} d\boldsymbol{z}$ is replaced by $\sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p$.
Directly considering the overall error enables deriving a consistent discretization-correction framework in which the kernel function $\eta^{\boldsymbol{\beta}}$ is designed such that all error terms of order $s < r$ vanish. This can be done by requiring the

discrete moments $Z_h^{\boldsymbol{\beta}}$ to satisfy the conditions

$$Z_h^{\boldsymbol{\alpha}} = \begin{cases} (-1)^{|\boldsymbol{\beta}|}\boldsymbol{\beta}!\,, & \boldsymbol{\alpha} = \boldsymbol{\beta}\,, \\ 0\,, & \boldsymbol{\alpha} \neq \boldsymbol{\beta}\,,\ \alpha_{\min} \leq |\boldsymbol{\alpha}| \leq |\boldsymbol{\beta}| + r - 1\,, \end{cases} \tag{2.4}$$

with $\alpha_{\min}$ as defined in equation (1.9). These conditions for the discrete moments are analogous to the conditions (1.8) for the continuous ones. The requirement for the modified $(r + |\boldsymbol{\beta}|)^{\text{th}}$ moment to be finite is superfluous for the discrete moments as their support is finite and the kernel function bounded by the other conditions. If the kernel function $\eta^{\boldsymbol{\beta}}$ fulfills the discrete conditions (2.4), we call it (and the corresponding operator) discretization-corrected.
An UC kernel $\eta^{\boldsymbol{\beta}}$ that satisfies the continuous moment conditions (1.8) fulfills the discrete moment conditions only in the limit $c \to 0$, where the difference between the discrete moments and the continuous ones vanishes. Depending on the desired accuracy, UC kernels may still be satisfactory for small enough c. Error terms proportional to ε^s with $s < r$ do, however, exist and can become dominant for high resolutions when $c = h/\varepsilon$ is kept constant.
We restrict the number of possible DC PSE kernels by constructing them according to the template

$$\eta^{\boldsymbol{\beta}}(\boldsymbol{z}, \boldsymbol{x}) = \left(\sum_{|\boldsymbol{\gamma}| = \alpha_{\min}}^{|\boldsymbol{\beta}| + r - 1} a_{\boldsymbol{\gamma}}(\boldsymbol{x}) \boldsymbol{z}^{\boldsymbol{\gamma}} \right) \exp\left(-|\boldsymbol{z}|^2\right) = K(\boldsymbol{z}, \boldsymbol{x}) \exp\left(-|\boldsymbol{z}|^2\right) \tag{2.5}$$

with a polynomial correction function $K(\boldsymbol{z}, \boldsymbol{x})$, where $\boldsymbol{x}$ is the position where the operator is evaluated and $\boldsymbol{z}$ is a location in a local coordinate system with origin at $\boldsymbol{x}$. We choose this template for its simplicity and similarity to the kernel functions proposed for UC PSE operators [31]. Generally, kernels with a small number of sign changes have better accuracy and robustness [37, 59]. Moreover, the nonvanishing discrete moments $Z_h^{\boldsymbol{\alpha}}$, $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| + r$, should be as small as possible in order to minimize the leading error term.
The unknown coefficients $a_{\boldsymbol{\gamma}}(\boldsymbol{x})$ in the kernel (2.5) are determined by solving a linear system of equations. This is obtained by substituting the

discrete moments of the kernel template (2.5) into the conditions (2.4):

$$\sum_{|\boldsymbol{\gamma}|=\alpha_{\min}}^{|\boldsymbol{\beta}|+r-1} a_{\boldsymbol{\gamma}}(\boldsymbol{x}) w_{\boldsymbol{\alpha},\boldsymbol{\gamma}}(\boldsymbol{x}) = \begin{cases} (-1)^{|\boldsymbol{\beta}|}\boldsymbol{\beta}!, & \boldsymbol{\alpha}=\boldsymbol{\beta}, \\ 0, & \boldsymbol{\alpha}\neq\boldsymbol{\beta}, \end{cases} \forall \boldsymbol{\alpha},\ \alpha_{\min} \leq \boldsymbol{\alpha} \leq |\boldsymbol{\beta}|+r-1, \quad (2.6)$$

with the weights

$$w_{\boldsymbol{\alpha},\boldsymbol{\gamma}}(\boldsymbol{x}) = \frac{1}{\varepsilon^{|\boldsymbol{\alpha}+\boldsymbol{\gamma}|+n}} \sum_{p\in\mathcal{N}(\boldsymbol{x})} v_p \left(\boldsymbol{x}-\boldsymbol{x}_p\right)^{\boldsymbol{\alpha}+\boldsymbol{\gamma}} \exp\left(-|\frac{\boldsymbol{x}-\boldsymbol{x}_p}{\varepsilon}|^2\right). \quad (2.7)$$

This linear system consists of m equations for m unknowns, where

$$m = \binom{|\boldsymbol{\beta}|+r-1+n}{n} - \alpha_{\min}. \quad (2.8)$$

The multiindices $\boldsymbol{\alpha}$ identify the equations (rows), and the multiindices $\boldsymbol{\gamma}$ identify the unknown coefficients (columns). While the continuous moment conditions are often redundant [31], this is not the case for the discrete ones and the full set of m equations must be considered. The condition number of this linear system is determined by the particle distribution and the choice of the kernel function template [32].
For uniform particle distributions, the coefficients $a_{\boldsymbol{\gamma}}$ are independent of $\boldsymbol{x}_p$. The linear system (2.6) then needs to be solved only once, and the same kernel can be used at all particle positions. Moreover, for symmetric particle distributions all odd moments of even function terms and even moments of odd function terms vanish. All coefficients $a_{\boldsymbol{\gamma}}$ for which $\boldsymbol{\gamma}+\boldsymbol{\beta}$ contains odd elements can therefore be set to zero *a priori*. The system of equations can then be reduced to size

$$m_{\text{symm}} = \binom{\frac{|\boldsymbol{\beta}|-n_{\text{odd}}}{2} + \lceil\frac{r}{2}\rceil - 1 + n}{n} - \delta_{0,n_{\text{odd}}} \quad (2.9)$$

by removing all rows and columns where $\boldsymbol{\alpha}+\boldsymbol{\beta}$ or $\boldsymbol{\gamma}+\boldsymbol{\beta}$ contain odd ele-

ments.[1] Here, n_{odd} is the number of odd elements of $\boldsymbol{\beta}$ and δ is Kronecker's delta.

For nonuniform particle distributions, the weights (2.7) are functions of $\boldsymbol{x}$. In this case, a different linear system has to be solved at every particle position $\boldsymbol{x}_p$. The coefficients $a_{\boldsymbol{\gamma}}(\boldsymbol{x})$ of DC PSE kernels may thus be different on different particles and have to be recomputed whenever particles have moved. For small m (low-order kernels for derivatives of small degree in low-dimensional spaces), this can efficiently be done using a closed-form expression for the coefficients $a_{\boldsymbol{\gamma}}(\boldsymbol{x})$ as a function of the weights $w_{\boldsymbol{\gamma},\boldsymbol{\alpha}}(\boldsymbol{x})$. In addition, the matrix of weights is symmetric if the moment conditions are properly ordered, and it typically contains several identical entries since $w_{\boldsymbol{i},\boldsymbol{j}}(\boldsymbol{x}) = w_{\boldsymbol{k},\boldsymbol{l}}(\boldsymbol{x})\,\forall \boldsymbol{x}$ if $\boldsymbol{i}+\boldsymbol{j} = \boldsymbol{k}+\boldsymbol{l}$.

Since DC PSE operators on nonuniformly distributed particles depend on the particle position, particle-particle interactions are not necessarily symmetric any more. This increases the computational cost of a simulation by up to a factor of two and impairs the exact conservativeness of the UC full-space PSE operators [68]. For first-order accurate operators, this can be remedied by using different correction functions for each pair of particles [52–54]. These correction functions are averages of the kernel correction functions at the two particle positions. Preserving symmetry for higher-order DC PSE operators is, to our knowledge, an open problem. A possible workaround is the use of a "midpoint derivative" for the quantification of the strength exchange between two particles p and q: instead of approximating the derivatives at both particle positions, $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$, we approximate it at position $\boldsymbol{x}_{p,q} = \boldsymbol{x}_{q,p} = (\boldsymbol{x}_p + \boldsymbol{x}_q)/2$ and use this estimation for the exchange between these two particles. Note that, since we do not know the value of $f(\boldsymbol{x}_{p,q})$, we cannot approximate the derivative with a PSE operator as defined in equation (1.10). Alternatively, we can use

[1] Equation (2.9) is derived by first finding the number m_i of weak n-combinations of $i \in \mathbb{N}$, $i + |\boldsymbol{\beta}|$ even, with n_{odd} odd parts at the locations of the odd elements of $\boldsymbol{\beta}$. The number m_i is equal to the number of (strong) n-combinations of $j = 2n + i - n_{\text{odd}}$ with only even parts. One can rewrite j as a sum of $j/2$ twos. Thus, m_i can be regarded as the number of permutations of a set consisting of $(n-1)$ commas and $(j/2-n)$ plus signs to be placed between the $j/2$ twos:

$$m_i = \binom{j/2-1}{n-1}.$$

Summing m_i over all even (odd) i, $\alpha_{\min} \leq i \leq |\boldsymbol{\beta}| + r - 1$, for even (odd) $|\boldsymbol{\beta}|$ yields m_{symm}.

the operator $Q^{\boldsymbol{\beta}}_{h,\text{midpoint}}$ defined by

$$Q^{\boldsymbol{\beta}}_{h,\text{midpoint}} f(\boldsymbol{x}) = \frac{1}{\varepsilon^{|\boldsymbol{\beta}|}} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p f(\boldsymbol{x}_p) \eta^{\boldsymbol{\beta}}_{\varepsilon}(\boldsymbol{x} - \boldsymbol{x}_p),$$

and kernels $\eta^{\boldsymbol{\beta}}$ satisfying conditions (2.4) with $\alpha_{\min} = 0$. The drawback of this procedure is the requirement to solve $kN/2$ instead of N linear systems of equations, where k is the average number of neighbors in $\mathcal{N}(\boldsymbol{x})$ and N is the number of particles.

2.3 Theoretical Analysis

In this section, we study the properties of DC PSE operators in theory. We focus on accuracy and convergence in §2.3.1, on numerical dispersion and numerical diffusion in §2.3.2, and on stability of explicit time stepping in §2.3.3.

2.3.1 Accuracy and Convergence

The error made when approximating the derivative $D^{\boldsymbol{\beta}} f(\boldsymbol{x})$ by $Q^{\boldsymbol{\beta}}_h f(\boldsymbol{x})$ is given in equation (2.1). Using Taylor's theorem, this error can be rewritten with a remainder term instead of an infinite series. Additionally assuming that the conditions (2.4) are perfectly satisfied, the overall error reduces to

$$\epsilon(\boldsymbol{x}) = \varepsilon^r \sum_{|\boldsymbol{\alpha}|=r+|\boldsymbol{\beta}|} \frac{(-1)^{|\boldsymbol{\alpha}|}}{\boldsymbol{\alpha}!} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon} \right)^{\boldsymbol{\alpha}} \eta^{\boldsymbol{\beta}}_{\varepsilon}(\boldsymbol{x} - \boldsymbol{x}_p) \, R^{\boldsymbol{\alpha}}_f(\boldsymbol{x}, \boldsymbol{x}_p),$$

where

$$R^{\boldsymbol{\alpha}}_f(\boldsymbol{x}, \boldsymbol{x}_p) = (D^{\boldsymbol{\alpha}} f)(\boldsymbol{x} + t(\boldsymbol{x}_p - \boldsymbol{x})),\ t \in [0, 1].$$

The error $\epsilon(\boldsymbol{x})$ can thus be bounded as

$$|\epsilon(\boldsymbol{x})| \leq \varepsilon^r C \max_{\substack{|\boldsymbol{\alpha}|=r+|\boldsymbol{\beta}| \\ \boldsymbol{y} \in B_{r_c}[\boldsymbol{x}]}} |D^{\boldsymbol{\alpha}} f(\boldsymbol{y})|,$$

where

$$C = \sum_{|\boldsymbol{\alpha}|=r+|\boldsymbol{\beta}|} \frac{1}{\boldsymbol{\alpha}!} \sum_{p\in\mathcal{N}(\boldsymbol{x})} v_p \left| \left(\frac{\boldsymbol{x}-\boldsymbol{x}_p}{\varepsilon} \right)^{\boldsymbol{\alpha}} \eta_\varepsilon^{\boldsymbol{\beta}} \left(\boldsymbol{x}-\boldsymbol{x}_p\right) \right|.$$

Convergence of order r is therefore guaranteed for any particle distribution for which the moment conditions (2.4) can be met, that is, for which the linear system of equations (2.6) does not become singular. Thus, there have to be enough neighboring particles within the cutoff radius r_c around the position under consideration. There is, however, no condition on the relationship between the kernel width ε and the particle spacing h.
The accuracy of the approximation depends on three things: the spatial resolution given by the kernel width ε, the nature of the field f, and the combination of particle distribution, cutoff radius, and kernel function η.

2.3.2 Wavenumber Modification

We quantify the wavenumber modifications introduced by the operators. Wavenumber modifications manifest themselves as numerical dispersion and numerical diffusion. We analyze the wavenumber modifications of DC PSE operators in three one-dimensional test cases: the linear wave equation, the convection equation, and the diffusion equation.

2.3.2.1 Linear Wave Equation and Convection Equation

We derive the wavenumber modification relations $k_{\mathrm{mod}}(k)$ of DC PSE operators for the one-dimensional linear wave equation

$$\frac{\partial^2 f}{\partial t^2} - u^2 \frac{\partial^2 f}{\partial x^2} = 0 \tag{2.10}$$

and the one-dimensional convection equation

$$\frac{\partial f}{\partial t} + u \frac{\partial f}{\partial x} = 0 \tag{2.11}$$

for speeds $u > 0$.
In order to derive expressions for the wavenumber modification, we compare the dispersion relation $\omega(k)$ of a traveling wave $\exp\left[\hat{\imath}(kx-\omega t)\right]$ obey-

ing the original equation (2.10) or (2.11), respectively, with the dispersion relation obtained when the spatial derivatives are approximated by a PSE operator (1.10). If the wave $\exp[\hat{\imath}(kx - \omega t)]$ is a solution of the original problem, then $\exp[\hat{\imath}(k_{\mathrm{mod}}(k)x - \omega t)]$ solves the spatially discretized equation. For irregular particle distributions, the wavenumber modification depends on the position x.
For the wave equation (2.10), we find the following relation between the original wavenumbers and the modified ones:

$$k_{\mathrm{mod}}(k) = \frac{1}{\varepsilon}\sqrt{Z_h^0(x) - \hat{\eta}_x^{(2)}(k\varepsilon)}, \tag{2.12}$$

where $\hat{\eta}_x^{\beta}(k\varepsilon)$ is defined as a nonuniform analog of the discrete Fourier transform of the kernel η^{β} centered at x, hence,

$$\hat{\eta}_x^{\beta}(k\varepsilon) = \sum_{p\in\mathcal{N}(x)} v_p \exp\left[-\hat{\imath}k(x - x_p)\right] \eta_{\varepsilon}^{\beta}(x - x_p, x).$$

Z_h^0 is the zeroth discrete moment of the kernel $\eta^{(2)}$ as defined in equation (2.3). Taylor-expanding $\hat{\eta}_x^{(2)}$ around $k\varepsilon = 0$ and substituting into equation (2.12) yields

$$k_{\mathrm{mod}}(k) = k\left[\sum_{\alpha=1}^{\infty} \frac{(-\hat{\imath}k\varepsilon)^{\alpha-2}}{\alpha!} Z_h^{\alpha}(x)\right]^{1/2}. \tag{2.13}$$

For the one-dimensional convection equation (2.11), the modified wavenumber is

$$k_{\mathrm{mod}}(k) = -\frac{\hat{\imath}}{\varepsilon}\left[Z_h^0(x) + \hat{\eta}_x^{(1)}(k\varepsilon)\right].$$

After Taylor-expanding around $k\varepsilon = 0$, this becomes

$$k_{\mathrm{mod}}(k) = k\left[-\sum_{\alpha=0}^{\infty} (\delta_{\alpha 0} + 1)\, \frac{(-\hat{\imath}k\varepsilon)^{\alpha-1}}{\alpha!} Z_h^{\alpha}(x)\right], \tag{2.14}$$

where δ_{ij} is the Kronecker delta, and the Z_h^{α} are the α^{th} discrete moments of the kernel $\eta^{(1)}$.
For symmetric particle distributions with even kernel functions η^{β} and even β, or with odd kernel functions η^{β} and odd β, the discrete moments

Z_h^α are zero for all odd $\alpha - \beta$. The effect of the wavenumber modification is thus purely dispersive in these cases. For nonsymmetric particle distributions, there may be nonzero moments Z_h^α for odd $\alpha - \beta$, introducing diffusive terms. The modified wavenumbers then contain imaginary parts that indicate position-dependent amplitude changes. This changes the shape of the original wave, analogously to a diffusion process. For even β, the diffusive terms are typically small compared to the dispersive terms. This is because the first discrete moments Z_h^α that are nonzero by design are the moments Z_h^β and $Z_h^{\beta+r}$.

Uniform Cartesian Particle Distributions On Cartesian particle distributions, the uniform interparticle spacing h limits the spectrum to $k \leq \pi/h$ (Nyquist-Shannon sampling theorem). We therefore restrict our analysis to the interval $k \in [0, \pi/h]$.
Figure 2.2 shows the wavenumber modification for the wave equation (2.10) and the convection equation (2.11), both for UC and DC PSE operators for $c = 0.5$, $c = 0.9$, and $r = 2, 4, 6, 8$. For the DC PSE operators, the kernel template (2.24) is used. For second-order operators ($r = 2$), the curves for UC and DC operators are indistinguishable. For higher orders of accuracy, the DC operators have smaller modifications of the wavenumbers than the UC ones. DC operators thus have lower numerical dispersion than UC PSE operators over the entire range of wavenumbers. Also, the wavenumber modifications decrease with increasing order of accuracy and increasing c, both for DC and UC operators.
The effect of numerical dispersion is illustrated for a two-dimensional advection problem in §2.4.2.1. As expected from the one-dimensional case discussed above, both the order of accuracy r and the ratio c influence the dispersive properties of the operators. Since dispersion decreases with increasing c, DC operators can be made much less dispersive than UC operators, as they allow larger values of c at full rate of convergence (see §2.4.1.1).

Irregular Particle Distributions On asymmetric particle distributions, the modified wavenumbers contain imaginary parts that lead to numerical diffusion. Figures 2.3 and 2.4 show the real and imaginary parts of the modified wavenumbers for the wave equation (2.10) and the convection equation (2.11), respectively. We compare UC and DC (using

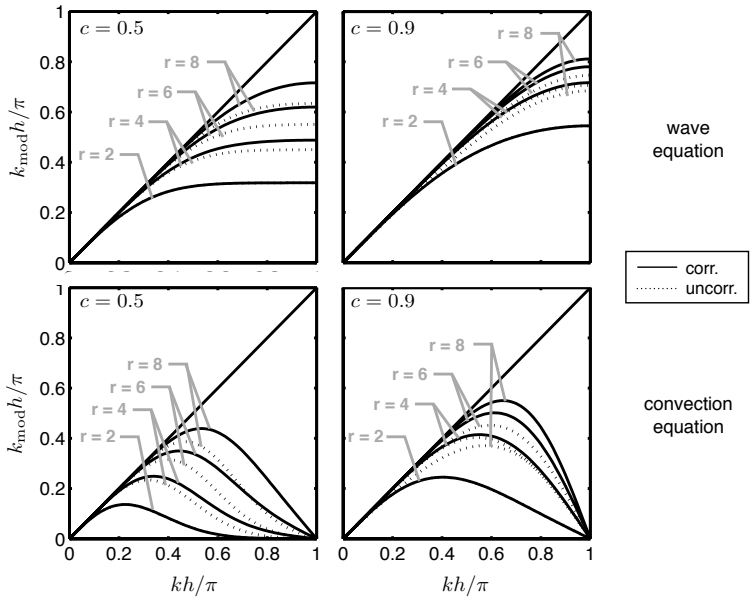

Figure 2.2: Wavenumber modification on uniform Cartesian particle distributions. The modified wavenumbers k_{mod} of the solutions of the one-dimensional wave equation (top) and the one-dimensional convection equation (bottom) are shown for DC (solid lines) and UC (dotted lines) PSE operators of orders $r = 2, 4, 6, 8$ with $c = 0.5$ (left) and $c = 0.9$ (right).

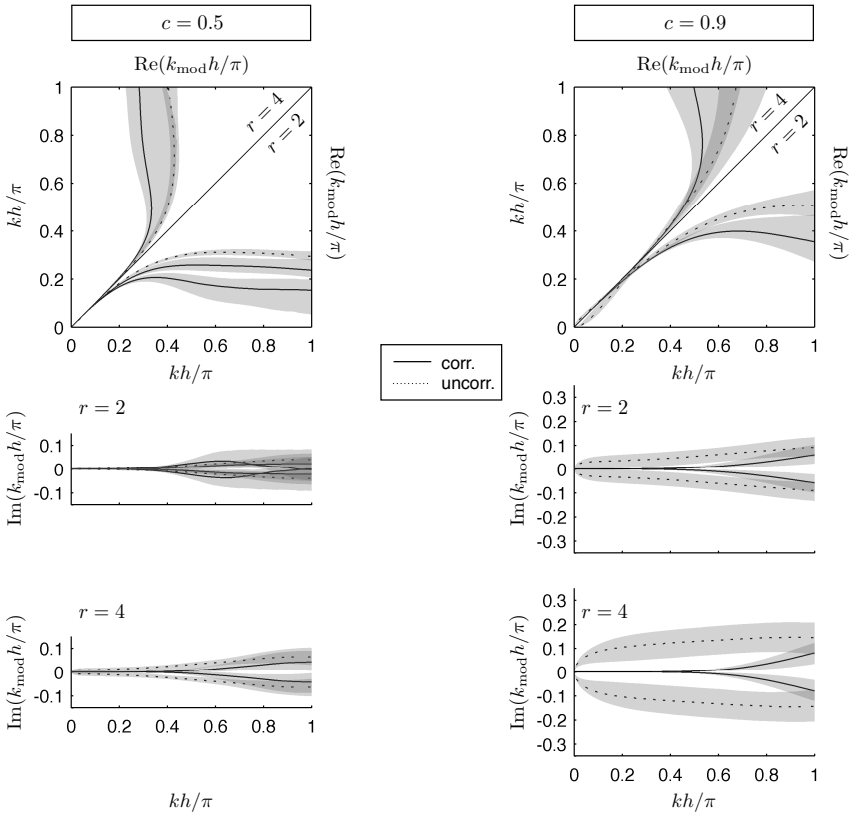

Figure 2.3: Wavenumber modification for the wave equation on random particle distributions. The real and imaginary parts of the modified wavenumbers k_{mod} are shown for DC and UC PSE operators of order $r = 2$ and $r = 4$ for $c = 0.5$ (left) and $c = 0.9$ (right). Each experiment is repeated for 10 000 random particle distributions. The lines report the median and the shaded areas the central 68.2% of the resulting curves.

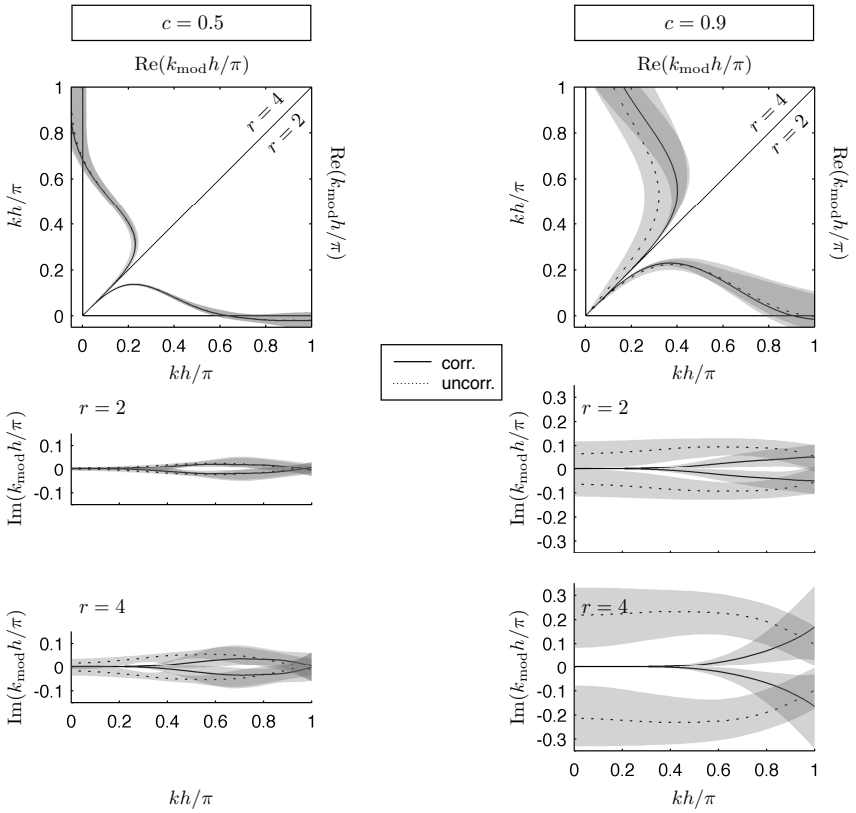

Figure 2.4: Wavenumber modification for the convection equation on random particle distributions. The real and imaginary parts of the modified wavenumbers k_{mod} are shown for DC and UC PSE operators of order $r = 2$ and $r = 4$ for $c = 0.5$ (left) and $c = 0.9$ (right). Each experiment is repeated for 10 000 random particle distributions. The lines report the median and the shaded areas the central 68.2% of the resulting curves.

the kernel template (2.5)) PSE operators evaluated on the center particle of 10 000 random particle distributions for $c = 0.5$ and $c = 0.9$. We report the medians (lines) and the areas covered by the central 68.2% of all curves $k_{\mathrm{mod}}(k)$ (shaded bands). The particle distributions are generated as follows: We subdivide the operator support $[-r_c, r_c]$ into equisized cells of width h and place one particle per cell. The position of each particle inside its cell is sampled from a uniform probability distribution. Particle distributions that lead to linear systems of equations with condition numbers above a certain threshold are resampled. According to the generalized sampling theorem for nonuniform sampling [61], we use the average interparticle spacing as the characteristic h, again limiting the spectrum of wavenumbers to $k \leq \pi/h$.
As in the uniform case, the modification of the real part decreases with increasing r and c. For the wave equation (figure 2.3), the UC operators lead to smaller modifications in $\mathrm{Re}(k)$ than the DC operators. For the convection equation (figure 2.4), however, DC operators show comparable or lower numerical dispersion than UC ones. The imaginary parts (leading to numerical diffusion) are always smaller for DC operators than for UC ones for wavenumbers $k \lesssim c\pi/h$.
For $c = 0.5$ and $r = 2$ for the wave equation (figure 2.3, left), $\mathrm{Re}(k_{\mathrm{mod}}(k))$ of DC operators splits into two curves. The upper curve results from all operators with strictly nonnegative kernel functions (83% of the random distributions tested). In this case, the nonnegative kernels lead to reduced numerical dispersion. The effect on numerical diffusion is, however, negligible.

2.3.2.2 Diffusion Equation

For the one-dimensional diffusion equation

$$\frac{\partial f}{\partial t} - \nu \frac{\partial^2 f}{\partial x^2} = 0\,, \tag{2.15}$$

the dispersion relation is $\omega(k) = -\hat{\imath}\nu k^2$, where $\nu > 0$ is the viscosity (or diffusion constant). The modified dispersion relation $\omega_{\mathrm{mod}}(k) = -\hat{\imath}\nu k_{\mathrm{mod}}(k)^2$ leads to the same modified wavenumber (2.13) as for the wave equation. Alternatively, the modification can also be absorbed into the viscosity as

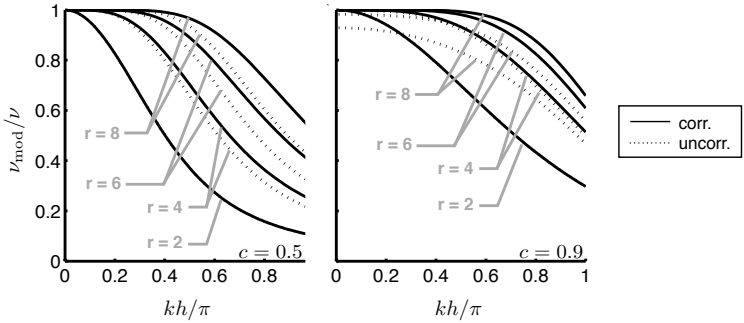

Figure 2.5: Viscosity modification for the diffusion equation on uniform Cartesian particle distributions. The ratio of the modified viscosity to the true viscosity ν_{mod}/ν is shown for DC and UC PSE operators of order $r = 2, 4, 6, 8$ for $c = 0.5$ (left) and $c = 0.9$ (right).

$\omega_{\mathrm{mod}}(k) = -\hat{\imath}\nu_{\mathrm{mod}}(k)k^2$. This leads to the modified viscosity

$$\nu_{\mathrm{mod}}(k) = \frac{\nu}{(k\varepsilon)^2}\left[Z_h^0(x) - \hat{\eta}_x^{(2)}(k\varepsilon)\right] = \nu\sum_{\alpha=1}^{\infty}\frac{(-\hat{\imath}k\varepsilon)^{\alpha-2}}{\alpha!}Z_h^\alpha(x)\,.$$

The modified viscosity for nonuniform particle distributions depends on the position x. $\mathrm{Re}(\nu_{\mathrm{mod}})/\nu > 1$ results in overdiffusion and $\mathrm{Re}(\nu_{\mathrm{mod}})/\nu < 1$ in underdiffusion. The effect of the imaginary part of the modified viscosity is both dispersive and diffusive.

Uniform Cartesian Particle Distributions Figure 2.5 shows ν_{mod}/ν versus kh/π for operators with $c = 0.5$ and $c = 0.9$ on uniform Cartesian particle distributions. All operators lead to increasing underdiffusion for increasing wavenumbers. For DC operators (using the kernel template (2.24)), however, the modified viscosity ν_{mod} is closer to the true viscosity ν than for UC operators. Also, increasing r or c leads to less underdiffusion for DC operators. For UC operators, high r and c lead to underdiffusion even for $kh \to 0$ due to the constant discretization error.

Random Particle Distributions On nonuniform particle distributions, the modified viscosity is complex. Figure 2.6 shows its real and

imaginary parts for DC (using the kernel template (2.5)) and UC operators. We show the medians and the areas covered by the central 68.2% of all curves $\nu_{\mathrm{mod}}(k)/\nu$ for 10 000 random particle distributions as described in §2.3.2.1.
While UC operators show less underdiffusion for larger wavenumbers, they do not reproduce the correct viscosity in the limit $kh \to 0$; their imaginary part approaches infinity and there is a bias in the real part. This inconsistency results from the nonvanishing discretization error. The effect becomes worse with increasing r and c.

2.3.3 Stability

Numerical stability requires that small perturbations, such as round-off errors, decay over time. If the effect $\delta f(t)$ of an error δf_0 introduced at time t_0 can be bounded by $|\delta f(t)| \leq \exp[\lambda(t - t_0)]|\delta f_0|$ with exponent $\lambda < 0$, the system is called exponentially stable. Typically, the amplification of each Fourier mode of δf_0 can be bounded separately by a wavenumber-dependent exponent $\lambda_{\boldsymbol{k}}$. Stability is then determined by $\lambda = \max_{\boldsymbol{k}} \lambda_{\boldsymbol{k}}$.
We consider the stability of numerical solutions of partial differential equations of order $|\boldsymbol{\beta}|$ in space and a in time,

$$\frac{\partial^a f}{\partial t^a} = bD^{\boldsymbol{\beta}} f\,, \tag{2.16}$$

with periodic boundary conditions. We distinguish two cases:

(I) the continuous-time case where the spatial derivative $D^{\boldsymbol{\beta}} f$ is replaced by the PSE approximation $Q_h^{\boldsymbol{\beta}} f$, and

(II) the discrete-time case, where in addition the time derivative is approximated by finite differences.

For the continuous-time case we derive the exponents $\lambda_{\boldsymbol{k}}$ in terms of the modified wavenumbers. For the discrete-time case we provide CFL conditions for different time-stepping schemes.

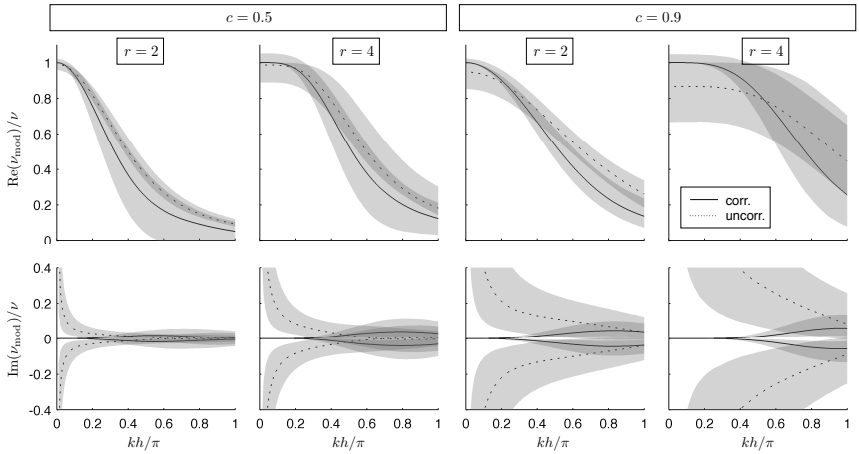

Figure 2.6: Viscosity modification for the diffusion equation on random particle distributions. The real and imaginary parts of ν_{mod}/ν are shown for DC and UC PSE operators of order $r = 2$ and $r = 4$ for $c = 0.5$ (left) and $c = 0.9$ (right). Each experiment is repeated for 10 000 random particle distributions. The lines report the median and the shaded areas the central 68.2% of the resulting curves.

2.3.3.1 Continuous Time

The modified wave $\exp[\hat{\imath}(\boldsymbol{k}_{\mathrm{mod}} \cdot \boldsymbol{x} - \omega t)]$, satisfying equation (2.16) when $D^{\boldsymbol{\beta}} f$ is replaced with $Q_h^{\boldsymbol{\beta}} f$, can be rewritten as

$$\exp\left[\hat{\imath}(\boldsymbol{k}_{\mathrm{mod}} \cdot \boldsymbol{x} - \omega t)\right] = \exp\left\{\mathrm{Re}\left[\left(b(\hat{\imath}\boldsymbol{k}_{\mathrm{mod}})^{\boldsymbol{\beta}}\right)^{1/a}\right] t\right\} \times \exp\left\{\hat{\imath}\left\{\boldsymbol{k} \cdot \boldsymbol{x} - \mathrm{Im}\left[\left(b(\hat{\imath}\boldsymbol{k}_{\mathrm{mod}})^{\boldsymbol{\beta}}\right)^{1/a}\right] t\right\}\right\}.$$

Thus, $\boldsymbol{k}_{\mathrm{mod}}$ is a function of $\boldsymbol{k}$ and the amplitude of the modified wave is $\exp(\lambda_{\boldsymbol{k}} t)$ with

$$\lambda_{\boldsymbol{k}} = \mathrm{Re}\left\{\left[b(\hat{\imath}\boldsymbol{k}_{\mathrm{mod}}(\boldsymbol{k}))^{\boldsymbol{\beta}}\right]^{1/a}\right\} = \begin{cases} \mathrm{Re}\left\{\left[\dfrac{b}{\varepsilon^{|\boldsymbol{\beta}|}} \displaystyle\sum_{|\boldsymbol{\alpha}|=1}^{\infty} \frac{(-\hat{\imath}\varepsilon\boldsymbol{k})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} Z_h^{\boldsymbol{\alpha}}(\boldsymbol{x})\right]^{1/a}\right\}, & |\boldsymbol{\beta}| \text{ even}, \\ \mathrm{Re}\left\{\left[\dfrac{b}{\varepsilon^{|\boldsymbol{\beta}|}} \displaystyle\sum_{|\boldsymbol{\alpha}|=0}^{\infty} (1+\delta_{|\boldsymbol{\alpha}|0}) \frac{(-\hat{\imath}\varepsilon\boldsymbol{k})^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} Z_h^{\boldsymbol{\alpha}}(\boldsymbol{x})\right]^{1/a}\right\}, & |\boldsymbol{\beta}| \text{ odd}. \end{cases}$$

Of the a roots of the term $(\cdot)^{1/a}$, the one with maximum real part is chosen. On a particle distribution of resolution h, the maximum exponent λ is the maximum over all $\lambda_{\boldsymbol{k}}$ with $|\boldsymbol{k}|h \in]0, \sqrt{n}\pi]$. For the one-dimensional wave equation (2.10), convection equation (2.11), and diffusion equation (2.15), the wavenumber-dependent exponents are

$$\lambda_k^{\mathrm{wave}} = u|\mathrm{Im}(k_{\mathrm{mod}}(k))|, \tag{2.17}$$

$$\lambda_k^{\mathrm{conv}} = u\mathrm{Im}(k_{\mathrm{mod}}(k)), \tag{2.18}$$

$$\text{and} \quad \lambda_k^{\mathrm{diff}} = \nu\left[\mathrm{Im}(k_{\mathrm{mod}}(k))^2 - \mathrm{Re}(k_{\mathrm{mod}}(k))^2\right], \tag{2.19}$$

respectively.
Using the expressions for the modified wavenumbers presented above, these exponents imply that on uniform Cartesian particle distributions all operators are exponentially stable for the diffusion equation and neutrally stable

($\lambda_k = 0$, neither error amplification nor decay) for the wave equation and the convection equation. Neutral stability is all that can be achieved using centered operators on the hyperbolic wave and convection equations. Exponential stability on these equations would require upwind schemes.
On irregular particle distributions, all operators lead to instabilities for the wave equation, due to the imaginary part of the modified wavenumber. For the convection equation, most operators result in $\mathrm{Im}(k_{\mathrm{mod}}) > 0$ for some $kh \in]0, \pi]$ and are thus unstable. For the diffusion equation, UC operators are exponentially stable on all, and DC operators on 98%, of the 10 000 randomly generated particle distributions. On 2% of the particle distributions, DC operators led to instabilities. Our results, however, suggest that this is the case only for $c > c_{\max}$, where the critical value $c_{\max}$ depends on the degree of Lagrangian grid distortion. We test this by considering particle distributions that are perturbations of uniform Cartesian distributions. Regularly placed particles with spacing h are perturbed by adding uniform random numbers in $[-\xi/2, \xi/2]$ to their positions. This mimics Lagrangian grid distortion in simulations where the particles are periodically remeshed. The parameter ξ quantifies the degree of distortion. Table 2.1 lists the critical $c_{\max}$ for different ξ for kernels with three different additional conditions on the zeroth-order moment. For each $\xi \in \{0.1, 0.2, \ldots, 1.5\}$, we report the maximum c for which no instabilities occurred over 10^7 random particle distributions. The results reflect the tradeoff between stability and numerical diffusion. Operators that introduce more numerical diffusion (low c) are more robust against Lagrangian grid distortion. We find $c_{\max}$ to be zero for all values of ξ tested if no condition is imposed on the zeroth moment. Of all conditions on the zeroth moment tested, the conditions $Z_h^0 = \{3, 4, 5\}$ result in the largest values for $c_{\max}$.

2.3.3.2 Discrete Time

Discrete-time stability depends on the time-discretization scheme. The maximum CFL numbers guaranteeing stable time integration depend on

Table 2.1: Critical c, $c_{\max}$, for stability of second-order DC PSE operators on the diffusion equation as a function of the degree of Lagrangian grid distortion ξ. We show the results for the additional moment condition $Z_h^0 = \{0, 5, 10\}$.

ξ/h	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
$Z_h^0 = 0$:															
$c_{\max}$	3.4	3.3	3.2	3.1	3.0	2.9	2.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$Z_h^0 = 5$:															
$c_{\max}$	3.4	3.3	3.2	3.1	3.0	2.9	2.8	1.0	0.9	0.7	0.7	0.7	0.6	0.6	0.6
$Z_h^0 = 10$:															
$c_{\max}$	3.4	3.3	3.2	3.1	3.0	2.9	2.8	1.0	0.8	0.3	0.2	0.2	0.2	0.2	0.2

the largest wavenumber-dependent exponents

$$\lambda^*_{\text{conv}} = 2 \max_k \frac{\lambda^{\text{conv}}_k}{uh|k_{\text{mod}}|^2}, \quad \lambda^*_{\text{diff}} = 2 \max_k \frac{\lambda^{\text{diff}}_k}{\nu h^2 |k_{\text{mod}}|^4},$$
$$\lambda^*_{\text{wave}} = 2 \max_k \frac{\lambda^{\text{wave}}_k}{uh|k_{\text{mod}}|^2}. \tag{2.20}$$

Table 2.2 shows the resulting time-step limits for three time-stepping schemes:

(i) first-order forward in time,

(ii) second-order centered in time, and

(iii) first-order backward in time.

In case (i) we approximate $\partial f/\partial t$ using first-order FT (forward in time) finite differences (explicit Euler scheme)[2]. Von Neumann stability analysis shows that this scheme with time step size Δt is exponentially stable for $|1 + \Delta t b(\hat{\imath} k_{\text{mod}})^\beta| < 1, \forall\, kh \in]0, \pi]$. For the convection equation on symmetric particle distributions, $\lambda^{\text{conv}}_k = 0$ (see §2.3.3.1). Any $\Delta t > 0$ thus leads to instabilities. This is because FTCS (forward in time, central in space) schemes are unstable for hyperbolic partial differential equations. On irregular particle distributions, however, a finite value for λ^*_{conv} is obtained. The more upwind particles are contained in the kernel support, the more likely λ^*_{conv} is negative, leading to exponentially stable explicit Euler time stepping. For the diffusion equation, exponential stability is guaranteed in all cases for $\Delta t < -\lambda^*_{\text{diff}} h^2/\nu$. Figure 2.7 shows the dependence of λ^*_{diff} on the ratio c and the order of accuracy r for UC and DC PSE operators. It can be seen that the underdiffusion introduced for low c and r improves stability. For random particle distributions, a positive upper bound for the time step, i.e. $\lambda^*_{\text{diff}} < 0$, is found for all UC operators and for DC operators with $c < c_{\max}$ (limited Lagrangian grid distortion) and fixed zeroth-order moment (see §2.3.3.1).

[2] We do not consider first-order FT schemes for the wave equation since they would involve two unknown values at future time points.

Table 2.2: Wavenumber-based stability conditions for different time discretization schemes and equations. We write $\min_k / \max_k$ for $\min_{k\in]0,\pi/h]} / \max_{k\in]0,\pi/h]}$.

exponential stability with first-order FT time discretization			
convection eq.	Δt	$<$	$\min_k -2\mathrm{Im}(k_{\mathrm{mod}})/(u\lvert k_{\mathrm{mod}}\rvert^2)$
			$= -\lambda^*_{\mathrm{conv}} h/u$
diffusion eq.	Δt	$<$	$\min_k 2[\mathrm{Re}(k_{\mathrm{mod}})^2 - \mathrm{Im}(k_{\mathrm{mod}})^2]/(\nu\lvert k_{\mathrm{mod}}\rvert^4)$
			$= -\lambda^*_{\mathrm{diff}} h^2/\nu$
neutral stability with second-order CT time discretization			
wave eq.	$\mathrm{Im}(k_{\mathrm{mod}}) = 0$ and $\Delta t \leq \min_k 2/(u\lvert k_{\mathrm{mod}}\rvert)$		
convection eq.	$\mathrm{Im}(k_{\mathrm{mod}}) = 0$ and $\Delta t \leq \min_k 1/(u\lvert k_{\mathrm{mod}}\rvert)$		
diffusion eq.	$\lvert\mathrm{Re}(k_{\mathrm{mod}})\rvert = \lvert\mathrm{Im}(k_{\mathrm{mod}})\rvert$ and $\Delta t \leq \min_k 1/(\nu\lvert k_{\mathrm{mod}}\rvert^2)$		
exponential stability with first-order BT time discretization			
wave eq.	Δt	$>$	$\max_k 2\lvert\mathrm{Im}(k_{\mathrm{mod}})\rvert/(u\lvert k_{\mathrm{mod}}\rvert^2)$
			$= \lambda^*_{\mathrm{wave}} h/u$
convection eq.	Δt	$>$	$\max_k 2\mathrm{Im}(k_{\mathrm{mod}})/(u\lvert k_{\mathrm{mod}}\rvert^2)$
			$= \lambda^*_{\mathrm{conv}} h/u$
diffusion eq.	Δt	$>$	$\max_k 2[\mathrm{Im}(k_{\mathrm{mod}})^2 - \mathrm{Re}(k_{\mathrm{mod}})^2]/(\nu\lvert k_{\mathrm{mod}}\rvert^4)$
			$= \lambda^*_{\mathrm{diff}} h^2/\nu$

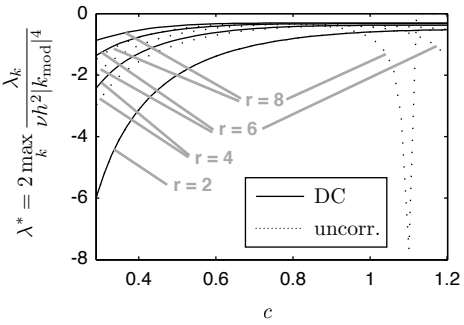

Figure 2.7: Influence of the ratio $c = h/\varepsilon$ and the order r on the stability of PSE solutions of the diffusion equation ($\nu > 0$). We show the maxima $\lambda^*_{\mathrm{diff}}$ of the normalized (dimensionless) wavenumber-dependent exponents for uniform Cartesian particle distributions for UC and DC PSE operators.

In case (ii) we consider the second-order CT (central in time) leapfrog scheme ($a = 1$)

$$f(x, t + \Delta t) = f(x, t - \Delta t) + 2\Delta t Q_h^\beta f(x, t) \tag{2.21}$$

and the second-order CT scheme ($a = 2$)

$$f(x, t + \Delta t) = 2f(x, t) - f(x, t - \Delta t) + \Delta t^2 Q_h^2 f(x, t)\,. \tag{2.22}$$

The conditions guaranteeing neutral stability for the one-dimensional wave, convection, and diffusion equations are listed in table 2.2. On uniform Cartesian particle distributions neutrally stable solutions of the wave and convection equations can be obtained with all operators. Highly dispersive operators again lead to less restrictive conditions. On random particle distributions, however, none of the tested operators yields stable solutions. For the diffusion equation, none of the operators tested is stable, neither on regular nor on random particle distributions. This is because the first condition in table 2.2 is never satisfied for CT schemes.
In case (iii) we use first-order BT (backward in time) differences, leading to an implicit time integration scheme. Such schemes have a *lower* bound for the time step required for exponential stability. This limits the accuracy that can be achieved since the time step can not be lowered arbitrarily in order to reduce the approximation error. This is because the BT scheme introduces additional numerical diffusion or dispersion. The neutral stability of the continuous-time case can thus be turned into exponential stability by taking large enough time steps. These lower bounds are listed in table 2.2. As expected, they are complementary to the upper bounds of FT schemes: Instead of an upper bound $\Delta t < -\lambda^* h/u$, we now have a lower bound $\Delta t > \lambda^* h/u$. On uniform Cartesian particle distributions, λ^*_{wave} is zero and exponential stability is guaranteed for all positive time steps. On irregular particle distributions, λ^*_{wave} is positive. From equations (2.20), (2.17), and (2.13) we see that for operators with nonzero Z_h^1, λ^*_{wave} grows as $h^{-1/2}$ when h tends to zero. This implies that the time step Δt can not be decreased at the same rate as the interparticle spacing h. In addition, λ^*_{wave} grows with decreasing c for UC operators. For the convection equation on irregular particle distributions, the lower bound for the time step can be positive or negative, depending on the sign of Z_h^0. As Z_h^0 approaches zero, $|\lambda^*_{\text{conv}}|$ grows. Solving the diffusion

equation with a BT scheme on uniform Cartesian particle distributions is unconditionally stable ($\lambda^*_{\text{diff}} \leq 0$) for all operators. On irregular particle distributions the same is the case for all UC operators and for all DC operators with $c < c_{\max}$ (limited Lagrangian grid distortion) and fixed zeroth-order moment (see §2.3.3.1).

2.4 Numerical Experiments

2.4.1 Fundamental Tests

We present numerical experiments that illustrate the rate of convergence, overall error, and computational efficiency of DC PSE operators. We compare DC PSE operators to UC PSE operators and to classical FD stencils. This section comprises four parts: In §2.4.1.1, convergence and computational efficiency are tested on uniform Cartesian particle distributions. The test problems are taken from Eldredge and coworkers [31] in order to demonstrate the fundamental characteristics of the operators. In §2.4.1.2 convergence is tested on irregular particle distributions and in §2.4.1.3 we assess boundary effects on finite-sized domains. §2.4.2.2 presents the operators' efficiencies in practical applications by studying a two-dimensional Lagrangian advection-diffusion benchmark.

2.4.1.1 Uniform Cartesian Particle Distributions

We present numerical experiments that demonstrate the rate of convergence and computational efficiency of DC PSE operators on uniform Cartesian particle distributions, and we compare the DC PSE operators to the corresponding limit FD stencils.

Rate of Convergence We consider the test case of evaluating the second derivative of the one-dimensional function

$$f(x) = \frac{1}{\sqrt{\pi\sigma^2}} \exp\left(-\frac{x^2}{\sigma^2}\right), \ \sigma = 0.05, \tag{2.23}$$

at the locations of all particles in the domain $[-0.5, 0.5]$. The particles are arranged with equidistant spacing h and the ratio $c = h/\varepsilon$ is kept constant

as the resolution is increased. We use the kernel function template

$$\eta^{\beta}(z) = z^{(\beta \bmod 2)} \left(\sum_{\gamma=0}^{m_{\text{symm}}-1} a_{\gamma} |z|^{\gamma} \right) \exp\left(-|z|^2\right), \tag{2.24}$$

which typically yields higher accuracy for one-dimensional operators on symmetric particle distributions than the more general template (2.5).
Figure 2.8 shows the L_2 norm of the relative[3] error in the approximation (1.10) of the derivative for different c and orders $r = 2$ (top row) and $r = 4$ (bottom row). The cutoff radii are $r_c = 3.5\varepsilon$ ($r = 2$) and $r_c = 5.5\varepsilon$ ($r = 4$) for the UC operators, and $r_c = 2\varepsilon$ ($r = 2$) and $r_c = 3\varepsilon$ ($r = 4$) for the DC operators. The error of order $s = -2$ for very high resolutions is due to limited machine precision and numerical extinction. It is not a feature of the operators.
The UC operators yield the desired rate of convergence over a wide range of resolutions if the ratio c is small (e.g. $c = 0.5$, which is the ratio used by Eldredge and coworkers). For larger values of c (curves for $c = 0.9$, $c = 1.0$, and $c = 1.4$), the error reaches a constant plateau at lower resolutions. This is due to the constant discretization error of the quadrature. For small c, the density of quadrature points is higher and the discrete moment conditions are hence closer to the continuous ones. The minimum error level $E_{\min}$ for the UC operators is given by the error term (2.1) that contains the discrete moment Z_h^2. This moment increases with increasing c, as shown in the right-most panels of figure 2.8. The minimum error level decreases exponentially with c. This is due to the discretization error decreasing exponentially for an integrand following the template (2.5) or (2.24) [11]. The discontinuities in $E_{\min}$ for small c and $r = 2$ are a result of the cutoff radius being fixed in terms of the kernel width ε: At the discontinuities, the number of particles in the kernel support changes, leading to a jump in Z_h^2 and, therefore, in $E_{\min}$. This effect becomes negligible for larger c and larger r_c. We also observe that $E_{\min}$ of the UC operators increases with increasing order r. This is confirmed by results for order $r = 6$ (not shown). The DC operators theoretically yield the desired rate of convergence for all orders and resolutions. The condition number

[3] We normalize the error with the L_∞ norm of the exact solution in the domain of interest. We do not show the L_∞ norm of the relative errors since it shows the same behavior as the L_2 norm.

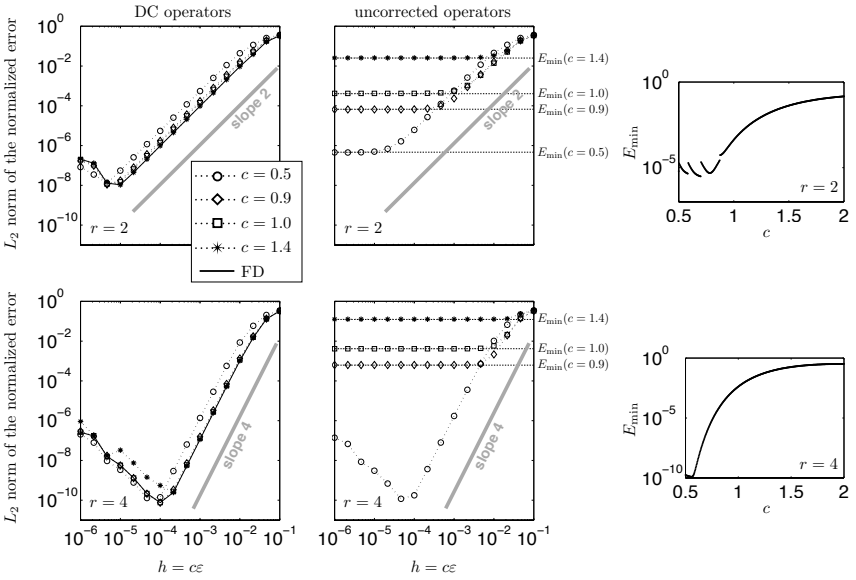

Figure 2.8: Convergence of the DC PSE operator approximating the second derivative of equation (2.23) on uniform Cartesian particle distributions. We show the convergence of the relative error for DC (left) and UC (center) PSE operators of orders $r = 2$ (top) and $r = 4$ (bottom). The panels on the right show the dependence of the minimum error level $E_{\min}$ of the UC operators on the ratio $c = h/\varepsilon$.

of the linear system of equations (2.6), however, increases with increasing c and r. For $r = 6$ and $c = 1.4$, Gauss elimination with partial pivoting is not sufficient any more and pre-conditioning might become necessary.
We also compare the DC PSE operators to the corresponding limiting FD stencils (solid lines in figure 2.8). For $r = 2$, these are obtained for $c \to \infty$ (see chapter 3) and for $r = 4$ for $r_c = 2h$ and $c = 1$. The error of the FD stencils mostly coincides with that of the DC PSE operators for $c = 1.4$, where the DC PSE operators effectively have the same support as the FD stencils.

Computational Efficiency We quantify the computational efficiency of the operators through the computational cost needed to reach a certain error level. Table 2.3 reports the computational costs of the two-dimensional advection case described in §2.4.2.1. We report the CPU time required for a single evaluation of the operators on all particles (t_1) and the total time needed to reach an L_2 error of $< 1\%$ of the initial pulse height in the whole domain at final time $T = 0.5$ (t_{all}). All timings were done using MATLAB 7.6.0.324 (R2008a) on a 2 GHz Intel Core Duo processor with 1 GB RAM. For all operators, we convolve the square matrix of precomputed kernel weights with the square matrix of field values f at the particle locations. As an implementation- and machine-independent measure, we also list the ratio $N = k/h^2$, where k is the number of neighboring particles (nonzero entries in the stencil). The time $t_1(N)$ is expected to be in $O(N)$.
For UC operators with $c = 0.55$, t_{all} decreases considerably with increasing order r, as already reported by Eldredge and coworkers [31]. This is true despite the fact that the cutoff radius increases from $r_c = 3.5\varepsilon$ to 5.5ε when going from $r = 2$ to $r = 4$. Increasing c to 0.9 for UC operators leads to higher efficiency since the number of neighbors within the fixed cutoff radius $r_c = 3.5\varepsilon$ decreases. As shown in figure 2.8, however, convergence of such operators is impaired.
It can be seen from table 2.3 that the computational efficiency of DC PSE operators is always higher than that of the corresponding UC operators. Furthermore, increasing c for DC operators improves their efficiency (through reducing k) without hampering convergence. In the limit $c \to \infty$, we recover the classical compact FD stencil as the most efficient operator. Compact FD stencils involve the least number of neighbors k at full accuracy and rate of convergence. They do, however, also provide the lowest

Table 2.3: CPU time needed for solving the two-dimensional test case of §2.4.2.1 on uniform Cartesian particle distributions to an final error of 1%. The time t_1 is required for a single evaluation of the operator on all particles, whereas t_{all} is the time needed to reach the target error level. The computational cost is governed by the implementation-independent ratio $N = k/h^2$, where k is the number of particles within the kernel support (nonzero entries in the stencil) and h the interparticle spacing.

kernel type	c	r	r_c	N	t_1 in ms	t_{all} in s
UC operator	0.55	2	3.5ε	6 611 570	329.	82.5
UC operator	0.90	2	3.5ε	840 145	37.7	5.77
DC operator	0.55	2	2ε	1 974 145	86.2	20.2
DC operator	0.90	2	2ε	165 035	15.3	1.98
DC operator	1.40	2	2ε	68 787	4.32	0.445
FD stencil		2		29 210	3.79	0.360
UC operator	0.55	4	5.5ε	1 211 890	66.5	16.3
UC operator	0.90	4	5.5ε	219 074	12.9	1.50
DC operator	0.55	4	3ε	304 978	15.6	3.10
DC operator	0.90	4	3ε	50 013	3.61	0.321
DC operator	1.40	4	3ε	11 464	1.50	0.0932
FD stencil		4		7643	1.51	0.0937

amount of regularization on noisy or discontinuous fields.

2.4.1.2 Irregular Particle Distributions

We assess the rate of convergence and the computational efficiency of DC PSE operators on irregular particle distributions. In order for the linear system (2.6) to be fully determined, each particle must have at least $k_{\min}$ neighbors, where $k_{\min} = m$ is the number of different multiindices $\boldsymbol{\alpha} \in \mathbb{N}^n$ with $|\boldsymbol{\alpha}| = \alpha_{\min}, \ldots, |\boldsymbol{\beta}| + r - 1$. All results presented in this section are obtained using the kernel function template (2.5).

Rate of Convergence We consider the convergence of the relative error when approximating the first derivative along x of the two-dimensional Gaussian pulse

$$f(x, y) = \frac{1}{\pi\sigma^2} \exp\left[-\frac{(x - x_0)^2 + (y - y_0)^2}{\sigma^2}\right], \quad \sigma = 0.1, \tag{2.25}$$

at position $\boldsymbol{x} = (x_0 + \sigma, y_0)$ using DC PSE operators of orders $r = 2, 4, 6$ on four different irregular particle distributions. The ratio $c = h/\varepsilon$ is kept constant at $c = \{0.5, 0.9, 1.4\}$, while the particle arrangements are scaled according to the desired resolution. All particle volumes are set to the average volume h^2. In real-world applications, where particle volumes evolve according to the flow field, better estimates can be used.
The results are summarized in figure 2.9. It can be seen that the UC operators diverge due to the discretization errors, whereas the DC PSE operators yield the desired rate of convergence in all cases. For $r = 6$, the error term of order $s = -1$ in equation (2.2) dominates at high resolutions, due to numerical inaccuracies when computing the coefficients $a_{\boldsymbol{\gamma}}(\boldsymbol{x})$. This underlines the importance of satisfying the discrete moment conditions as accurately as possible.

Computational Efficiency On irregular particle distributions, the DC kernel function $\eta^{\boldsymbol{\beta}}(\boldsymbol{z})$ becomes a field of kernel functions, $\eta^{\boldsymbol{\beta}}(\boldsymbol{z}, \boldsymbol{x})$, thus requiring the solution of a different linear system of equations for each position $\boldsymbol{x}$ where the operator is to be evaluated (see §2.2). Moreover, the kernels have to be recomputed whenever the particles have moved. The computational cost then becomes comparable to that of the vorticity

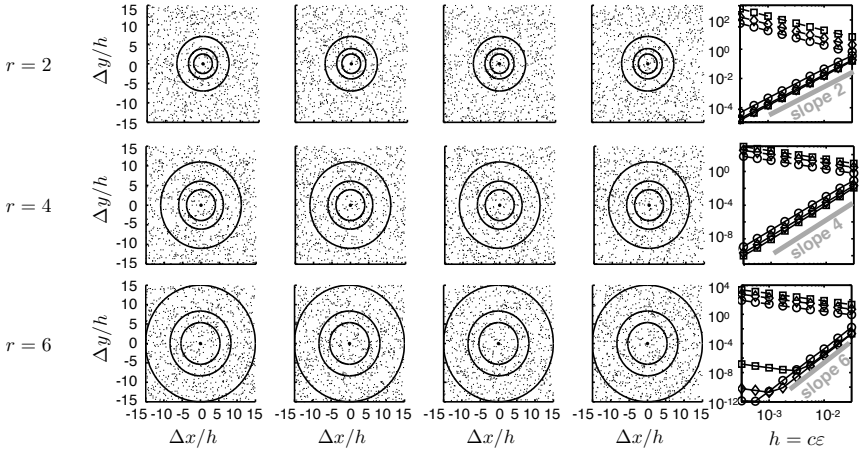

Figure 2.9: Convergence of DC PSE operators (solid lines) and of UC PSE operators (dashed lines) of order $r = 2$, 4, 6 (top to bottom) on random particle distributions. The right panels show the maximum of the relative error when approximating the first derivative of the Gaussian pulse (2.25) at the center particles of four random particle distributions for $c = 0.5$ ($\bigcirc$), $c = 0.9$ ($\Diamond$), and $c = 1.4$ ($\square$). The considered particle distributions, with center particles at positions $\boldsymbol{x} = (x_0 + \sigma, y_0)$, are shown in the four panels on the left. The circles show the supports of the operators for the different values of c. The cutoff radii are $r_c = 3.5\varepsilon, 5.5\varepsilon, 7.5\varepsilon$ for $r = 2, 4, 6$, respectively.

redistribution method [80], see section 3.5, or the discrete PSE scheme for the diffusion step in vortex methods proposed by Poncet [70]. For a detailed discussion of the computational cost of DC PSE operators on moving particles, we refer to §2.4.2.2.
If the particle distribution is not arbitrary, but the result of a known convective velocity field, the kernels need not necessarily be recomputed at every time step, but can be adapted based on the known velocity. Starting from the initial particle distribution and the corresponding DC PSE operators, one can define a mapping from the old to the new particle positions. The operators can then be adapted in analogy to the variable vortex blob method [25, 44]. For nonlinear mappings, however, additional error terms appear. For high orders of accuracy, we thus expect velocity-based operator adaptation to be computationally more expensive than reinitializing the particles at every time step.

2.4.1.3 Boundary Effects

The accuracy of UC full-space PSE operators usually deteriorates near boundaries of the computational domain. Eldredge and coworkers [31] therefore derived one-sided integral operators with significantly improved accuracy near boundaries. Such a special treatment is not necessary when using DC PSE operators since they do not rely on regularity or symmetry of the particle distribution. The skewed particle distributions near boundaries are simply treated as irregular particle distributions, and the corresponding DC PSE operators are constructed by solving the resulting linear systems of equations.
We demonstrate that the presence of boundaries does not affect the order of accuracy of DC PSE operators by considering a two-dimensional Gaussian pulse (2.25) leaving the computational domain through an absorbing boundary, see figure 2.10. Convergence of the resulting relative errors is shown in figure 2.11 for UC full-space, UC one-sided, and DC PSE operators approximating the directional derivative $\boldsymbol{d} \cdot \nabla f$ of the Gaussian pulse centered at $(x_0, y_0) = (0.35, 0.45)$ in a computational domain of size $[0, 0.5]^2$. For the one-sided UC operators we only consider particles in the upwind direction of $\boldsymbol{d}$. It can be seen from figure 2.11 that the UC full-space operators diverge in all cases (dominating error of order $s = -1$) and should not be used near boundaries. The UC one-sided operators show constant L_∞ errors (bottom row of panels) for orders $r = 1$ to $r = 3$

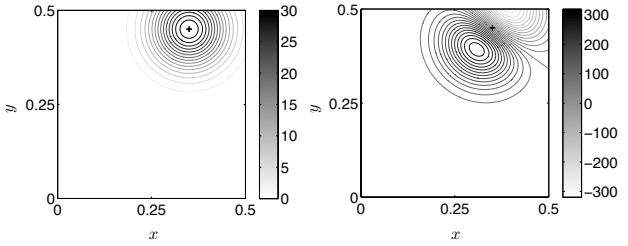

Figure 2.10: Test case used to assess the convergence of the operators near boundaries. The left panel shows the contour lines of a Gaussian pulse (2.25) centered at $(x_0, y_0) = (0.35, 0.45)$ in a domain $[0, 0.5]^2$. The right panel shows the contour lines of the analytical solution of the directional derivative along $\boldsymbol{d} = (0.588, 0.809)$. The + marks the center of the pulse.

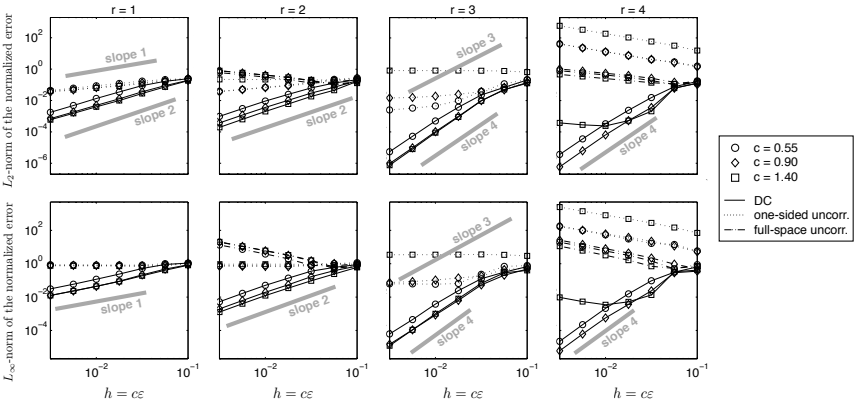

Figure 2.11: Convergence of the UC full-space, UC one-sided, and DC PSE operators approximating the directional derivative of the Gaussian pulse shown in figure 2.10. We show convergence of both the L_2 (top row of panels) and L_∞ norms of the relative errors for operators of design orders $r = 1, 2, 3, 4$ (left to right) and different c. UC full-space operators are shown for even orders r only.

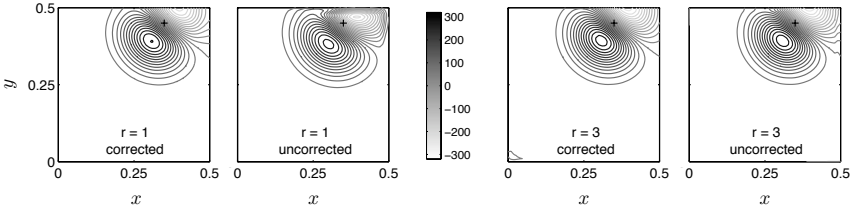

Figure 2.12: Visual comparison of the results obtained with UC one-sided operators and DC PSE operators for the test case in figure 2.10. We show contour lines of the numerically computed approximations of the directional derivative of the Gaussian pulse for $r = 1$ (left) and $r = 3$ (right). Both cases use $h = 0.01$ and c is $c = 0.55$. The + marks the center of the pulse.

and diverge for $r = 4$. The DC PSE operators yield the desired rates of convergence in both the L_2 (upper panels) and the L_∞ norms until numerical inaccuracies in the determination of the kernel coefficients start to dominate the error ($r = 4$ with $c = 1.4$, solid line with square symbols). Figure 2.12 visually compares the solutions obtained using UC one-sided and DC PSE operators of orders $r = 1$ and $r = 3$. The example shown is computed using $h = 0.01$ and $c = 0.55$. It can be seen that the UC operators lead to distortions and kinks in the iso-lines near boundaries, preventing convergence of the L_∞ error.

2.4.2 The Operators in Simulations

2.4.2.1 On the Dispersive Wavenumber Modification: A Two-Dimensional Advection Simulation with an Eulerian Frame of Reference

As done by Eldredge and coworkers [31], we demonstrate the numerical dispersion induced by the wavenumber modification of the operators by simulating the advection of a two-dimensional Gaussian pulse (2.25) in direction $\boldsymbol{d} = (0.588, 0.809)$ with unit speed, thus,

$$\frac{\partial f}{\partial t} + \boldsymbol{d} \cdot \nabla f = 0\,.$$

We approximate $\boldsymbol{d} \cdot \nabla f$ using both UC and DC PSE operators, as well as classical FD stencils. For the DC operators we use the kernel function template (2.5).
Figures 2.13 and 2.14 show the simulation results for second order ($r = 2$) and fourth-order ($r = 4$) operators, respectively, on uniform Cartesian particle distributions ($h = 0.02$). The panels show the pulse at times $t = 0$ and $t = 0.5$ in the domain $[-0.1, 1]^2$ with homogeneous Dirichlet boundary conditions. Time integration is done using the leapfrog scheme (2.21) with a time step of $\Delta t = 0.5h$ for the simulations of order $r = 2$, and $\Delta t = 20h^2$ for the simulations of order $r = 4$.
The results visually illustrate that the order of accuracy r and the ratio c both influence the dispersive properties of the operators. Dispersion decreases with increasing r and c. Since DC PSE operators allow larger values of c at full rate of convergence, their numerical dispersion can be reduced down to the level of the FD stencils. This is not possible for UC operators, where the solution gradually deteriorates for increasing c. For low c, PSE operators introduce distortion along direction $\boldsymbol{d}$, whereas the distortion introduced by FD is oriented along the coordinate axes. This is particularly evident for $r = 2$.

2.4.2.2 On Computational Efficiency: A Lagrangian Advection-Diffusion Test Case

We demonstrate the value of DC PSE operators in a more complex test case with moving particles: a two-dimensional Lagrangian advection-diffusion simulation. Again, we compare DC PSE operators to UC PSE operators and finite differences. Details of the test case and the numerical schemes are described in appendix A.1. We assess the computational cost of DC PSE operators by measuring the CPU times of simulations for the Péclet numbers Pe = {1, 10, 100, 1000} to final relative errors of less than {0.1%,0.05%,0.05%,0.1%}, respectively. These error levels are chosen such that they can be reached also with UC operators (see §2.4.1.1). We only test second-order accurate operators. The particles are either remeshed to regular Cartesian positions at every time step, or they are never remeshed. The latter is possible because the velocity field does not lead to holes in the particle distribution. FD stencils are evaluated using the connectivity information of the mesh, whereas the PSE simulations use neighbor lists (we use Verlet lists [83]) even if remeshing is done at every time step. In

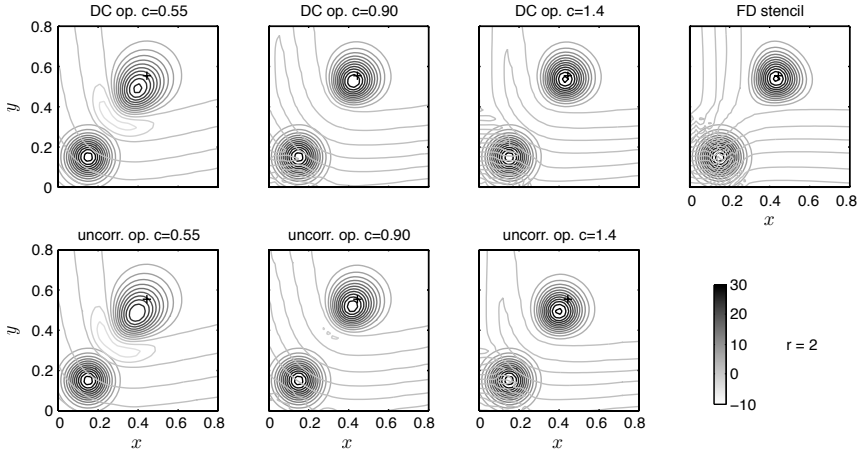

Figure 2.13: Effect of numerical dispersion for second-order operators. The contour lines of an advected Gaussian pulse (2.25) are shown at times $t = 0$ and $t = 0.5$. At $t = 0$, the pulse is centered at $\boldsymbol{x} = (0.15, 0.15)$. For $t > 0$, it is propagated along the direction $\boldsymbol{d} = (0.588, 0.809)$ at unit speed. The $+$ marks the center of the pulse in the exact solution at time $t = 0.5$.

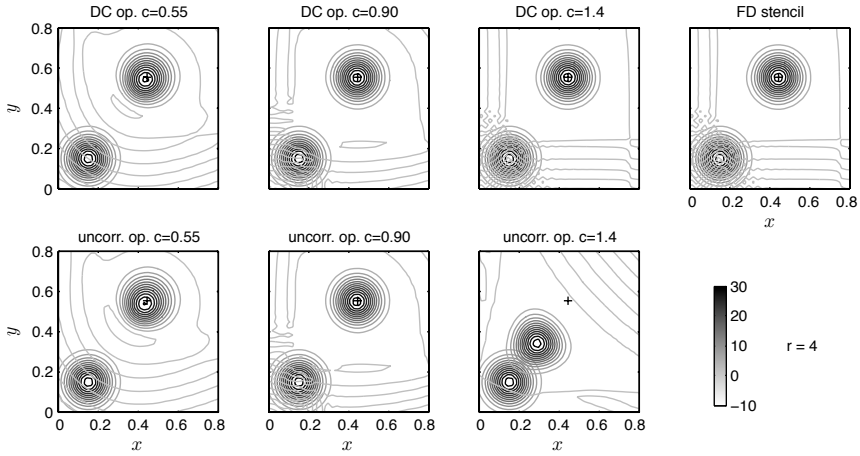

Figure 2.14: Effect of numerical dispersion for fourth-order operators. The contour lines of an advected Gaussian pulse (2.25) are shown at times $t = 0$ and $t = 0.5$. At $t = 0$, the pulse is centered at $\boldsymbol{x} = (0.15, 0.15)$. For $t > 0$, it is propagated along the direction $\boldsymbol{d} = (0.588, 0.809)$ at unit speed. The $+$ marks the center of the pulse in the exact solution at time $t = 0.5$.

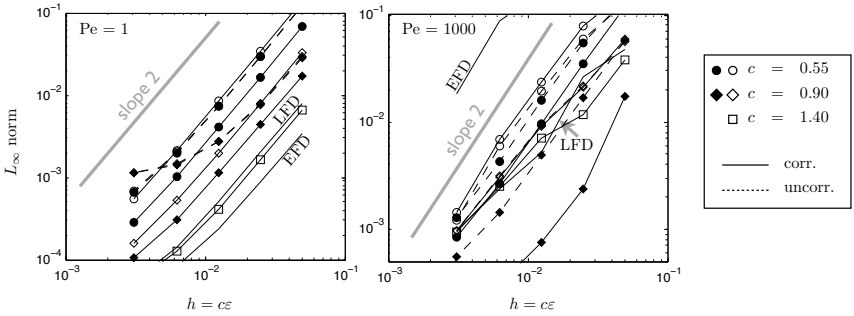

Figure 2.15: Convergence plots for the advection-diffusion simulation for Pe=1 (left) and Pe = 1000 (right). Empty markers denote simulations with remeshing, while filled markers denote simulations without remeshing. The FD simulations are shown as solid lines without markers. LFD refers to Lagrangian FD and EFD to Eulerian FD.

all cases, the Verlet lists are recomputed at every time step.
We show the convergence plots for Péclet numbers 1 and 1000 in figure 2.15. All results are summarized in table 2.4. As expected, Eulerian FD show the best computational efficiency for low Péclet numbers. For high Péclet numbers (100 and 1000), Lagrangian methods are more efficient. In all cases, DC PSE operators are more efficient than UC ones. DC PSE operators can also outperform Lagrangian FD (see appendix A.1.3.3) at high Péclet numbers (100 and 1000), since they do not require remeshing. If remeshing is done at every time step, FD stencils are preferable. The larger the Péclet number, however, the more the remeshing error dominates over the error introduced by the diffusion operator. This can be seen by comparing the resolution h that is required to reach the target error level.
On irregular particle distributions (i.e., without remeshing), DC operators require that the correction function is recomputed for each particle at every time step. This additional computational cost accounts for 85 to 90% of the total CPU time. It is, however, amortized by the gain in accuracy of DC operators in all cases except Pe = 1: the target error level can be reached with coarser resolution. This coarser resolution also allows larger time steps. In cases where the advection error dominates, higher

Table 2.4: CPU time t_{CPU} (in seconds) and interparticle spacing (or mesh resolution) h required for solving the two-dimensional advection-diffusion problem described in appendix A.1 for $\text{Pe} = \{1, 10, 100, 1000\}$ with a relative error of $\{0.1\%, 0.05\%, 0.05\%, 0.1\%\}$, respectively. We compare DC PSE operators with UC ones and FD for $r = 2$ and $\Delta t \lesssim 0.15 h^2 \text{Pe}$. Values are based on interpolations from simulations with $h \in \{3.125, 6.25, 12.5, 25.0, 50.0\} \cdot 10^{-3}$.

				Pe = 1		Pe = 10		Pe = 100		Pe = 1000	
	op.	c	r_c	$10^3\,h$	t_{CPU}	$10^3\,h$	t_{CPU}	$10^3\,h$	t_{CPU}	$10^3\,h$	t_{CPU}
remeshing	UC	0.55	3.5ε	3.99	36 300	4.85	5020	5.18	583.	2.87[a]	739.[a]
	UC	0.90	3.5ε	2.07[a]	21 300[a]	4.86	2160	5.40	213.	3.18	203.
	DC	0.55	2ε	4.27	10 900	4.89	2077	5.09	274.	2.67[a]	405.[a]
	DC	0.90	2ε	8.74	256.	7.34	191.	5.30	121.	3.20	89.1
	DC	1.40	2ε	19.5	6.87	7.60	120.	5.24	86.3	3.24	69.4
no remeshing	UC	0.55	3.5ε	4.10	31 600	5.02	4280	5.73	381.	2.73[a]	915.[a]
	UC	0.90	3.5ε	2.24[a]	144 000[a]	7.72	289.	10.2	13.9	4.79	37.1
	DC	0.55	2ε	6.17	10 100	7.29	1600	7.55	210.	3.50	533.
	DC	0.90	2ε	11.7	533.	15.6	53.1	23.9	1.49	14.9	1.21
Lagrangian FD				17.6	0.716	6.80	11.0	4.55	8.70	3.22	4.41
Eulerian FD				26.8	< 0.02	9.80	0.268	2.37[a]	16.7[a]	0.883[a]	41.0[a]

[a] Value based on extrapolation.

resolution might therefore again be required, making DC PSE operators less efficient.

2.5 DC PSE Operators in Multiresolution Simulations

DC PSE operators can straightforwardly be used in simulations with particles of different sizes. In this case, the operator as defined in equation (1.10) has to be based on a position-dependent kernel width $\varepsilon(\boldsymbol{x})$ instead of a globally constant kernel width ε, hence,

$$Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{1}{\varepsilon(\boldsymbol{x})^{|\boldsymbol{\beta}|}} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(f(\boldsymbol{x}_p) \pm f(\boldsymbol{x}) \right) \eta^{\boldsymbol{\beta}}_{\varepsilon(\boldsymbol{x})}(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x}).$$

Taylor-expanding around $\boldsymbol{x}$ and subtracting the exact derivative $D^{\boldsymbol{\beta}} f(\boldsymbol{x})$ yields the error

$$\begin{aligned}
\epsilon(\boldsymbol{x}) &= Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) - D^{\boldsymbol{\beta}} f(\boldsymbol{x}) \\
&\leq \varepsilon(\boldsymbol{x})^r \|f\|_{r+|\boldsymbol{\beta}|,\infty,B_{r_c(\boldsymbol{x})}[\boldsymbol{x}]} \sum_{|\boldsymbol{\alpha}|=r+|\boldsymbol{\beta}|} \frac{1}{\boldsymbol{\alpha}!} \\
&\quad \times \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left| \left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon(\boldsymbol{x})} \right)^{\boldsymbol{\alpha}} \eta^{\boldsymbol{\beta}}_{\varepsilon(\boldsymbol{x})}(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x}) \right|,
\end{aligned}$$

given that the discrete moments

$$Z_h^{\boldsymbol{\alpha}}(\boldsymbol{x}) = \frac{1}{\varepsilon(\boldsymbol{x})^n} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon(\boldsymbol{x})} \right)^{\boldsymbol{\alpha}} \eta^{\boldsymbol{\beta}} \left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon(\boldsymbol{x})}, \boldsymbol{x} \right)$$

fulfill the moment conditions (2.4). Thus, if the cutoff radius $r_c(\boldsymbol{x})$ grows as $\varepsilon(\boldsymbol{x})$, the approximation is consistent with an error of order $O(\varepsilon(\boldsymbol{x})^r)$.

2.6 DC PSE Operators in Strong Formulations

DC PSE operators can be constructed in a straightforward way for function approximations in strong formulations, as in equation (4). The difference between the particle-based approximations in weak and strong descriptions

is the existence of a particle volume for the former case. Thus, in the latter case, particles can also be seen as (moving) data points.
Strong descriptions are usually less robust towards discontinuities. For the construction of a DC PSE operator, however, the absence of volumes is inconsequential: omitting the volumes in definition (2.3) of the discrete moments leads to DC PSE operators for strong formulations.

2.7 Versatility of DC PSE Operators

As discussed in §2.2 and §2.4.1, the use of DC operators on changing, nonuniform particle distributions is costly due to the solution of the linear system of equations (2.6) for each particle and each time step. If the evaluation of several different differential operators is required, e.g. the gradient and the Laplacian, the total cost can be significantly reduced by reusing the solutions of the linear systems of equations. The possibility of doing so becomes obvious by rewriting the kernel function template (2.5) and the moment conditions (2.4). For simplicity, we omit the volumes v_p in the following. They are implicitly contained in the resulting kernel function through the discretization correction. The DC PSE operator then reads

$$Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{1}{\varepsilon(\boldsymbol{x})^{|\boldsymbol{\beta}|}} \sum_{p \in \mathcal{N}(\boldsymbol{x})} \left(f(\boldsymbol{x}_p) \pm f(\boldsymbol{x})\right) \eta_\varepsilon^{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x}).$$

With the kernel function template (2.5) expressed as

$$\eta_\varepsilon^{\boldsymbol{\beta}}(\boldsymbol{z}, \boldsymbol{x}) = \left[\boldsymbol{p}\left(\frac{\boldsymbol{z}}{\varepsilon(\boldsymbol{x})}\right) \boldsymbol{a}^{\mathrm{T}}(\boldsymbol{x})\right] \exp\left(-\frac{|\boldsymbol{z}|^2}{\varepsilon(\boldsymbol{x})^2}\right), \tag{2.26}$$

where the row vector $\boldsymbol{p}(\boldsymbol{z}/\varepsilon) \in \mathbb{R}^m$ contains the m monomials $\{(\boldsymbol{z}/\varepsilon)^{\boldsymbol{\gamma}}\}_{\alpha_{\min} \leq |\boldsymbol{\gamma}| \leq |\boldsymbol{\beta}|+r-1}$ and $\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{x}) \in \mathbb{R}^m$ is the column vector of unknown coefficients, the discrete moment conditions (2.4) can be expressed as

$$\boldsymbol{A}(\boldsymbol{x})\boldsymbol{a}^{\mathrm{T}}(\boldsymbol{x}) = \boldsymbol{b}^{\mathrm{T}} \tag{2.27}$$

with

$$\begin{aligned}\boldsymbol{A}(\boldsymbol{x}) &= \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{B}(\boldsymbol{x}) \in \mathbb{R}^{m\times m},\\ \boldsymbol{B}(\boldsymbol{x}) &= \boldsymbol{E}(\boldsymbol{x})\boldsymbol{V}(\boldsymbol{x}) \in \mathbb{R}^{k\times m},\\ \boldsymbol{b} &= (-1)^{|\boldsymbol{\beta}|} D^{\boldsymbol{\beta}}\boldsymbol{p}(\boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{0}} \in \mathbb{R}^{m},\end{aligned}$$

where $\boldsymbol{V}(\boldsymbol{x})$ is the Vandermonde matrix associated with the set of monomials $\boldsymbol{p}((\boldsymbol{x}-\boldsymbol{x}_p)/\varepsilon(\boldsymbol{x}))$ and the set of points $\{\boldsymbol{x}_p\}_{p\in\mathcal{N}(\boldsymbol{x})}$, and $\boldsymbol{E}(\boldsymbol{x})$ is a diagonal matrix determined by the kernel's window function:

$$\begin{aligned}\boldsymbol{V}(\boldsymbol{x}) &= \begin{pmatrix} p_1(\boldsymbol{z}_1/\varepsilon) & p_2(\boldsymbol{z}_1/\varepsilon) & \cdots & p_m(\boldsymbol{z}_1/\varepsilon)\\ p_1(\boldsymbol{z}_2/\varepsilon) & p_2(\boldsymbol{z}_2/\varepsilon) & \cdots & p_m(\boldsymbol{z}_2/\varepsilon)\\ \vdots & \vdots & \ddots & \vdots\\ p_1(\boldsymbol{z}_k/\varepsilon) & p_2(\boldsymbol{z}_k/\varepsilon) & \cdots & p_m(\boldsymbol{z}_k/\varepsilon)\end{pmatrix} \in \mathbb{R}^{k\times m}, \qquad (2.28)\\ \boldsymbol{E}(\boldsymbol{x}) &= \operatorname{diag}\left(\left\{\exp\left(-|\boldsymbol{z}_i|^2/(2\varepsilon^2)\right)\right\}_{i=1}^{k}\right) \in \mathbb{R}^{k\times k},\end{aligned}$$

where $\{\boldsymbol{z}_i\}_{i=1}^{k} = \{\boldsymbol{x}-\boldsymbol{x}_p\}_{p\in\mathcal{N}(\boldsymbol{x})}$ is the set of vectors pointing from all particles within the neighborhood of $\boldsymbol{x}$ to $\boldsymbol{x}$.
The matrix $\boldsymbol{A}(\boldsymbol{x})$ contains information about the spatial distribution of the particles $\{\boldsymbol{x}_p\}$ around $\boldsymbol{x}$, while the right-hand side $\boldsymbol{b}$ determines the approximation properties of the kernels. Because the diagonal elements of $\boldsymbol{E}(\boldsymbol{x})$ are strictly positive, the invertibility of $\boldsymbol{A}$ depends only on that of the Vandermonde matrix $\boldsymbol{V}$. If $\boldsymbol{V}$ is not invertible, the issue can usually be solved by inserting or displacing some particles. If $\boldsymbol{V}$ is invertible, $\boldsymbol{A}$ is symmetric and positive definite (as the product of a real matrix and its transpose) and can efficiently be inverted using, for example, Cholesky decomposition. Nevertheless, this operation represents most of the computational cost of solving the linear system of equations (2.27) for the unknown kernel coefficients $\boldsymbol{a}(\boldsymbol{x})$. Solving this system for multiple right-hand sides $\boldsymbol{b}$ in order to compute derivatives of different orders, however, comes at little additional cost.[4]
If both even and odd derivatives are to be computed, setting $\alpha_{\min}$ to zero

[4] Note that the order of accuracy r of the operators depends both on the number of moment conditions m and on the order $|\boldsymbol{\beta}|$ of the derivative that is approximated. If the moments up to order $\alpha_{\max}$ satisfy the moment conditions, the order of accuracy of an operator approximating a $|\boldsymbol{\beta}|^{\text{th}}$ derivative is $r_{|\boldsymbol{\beta}|} = \alpha_{\max} - |\boldsymbol{\beta}|$.

for both cases is beneficial because then both systems (2.27) have the same left-hand side. Setting $\alpha_{\min}$ to zero is unnecessary for the even derivative and, since there is one additional moment condition to be satisfied, the minimum number of neighbors grows by one. The increase in computational cost due to the larger number of neighbors, however, is typically small compared to the cost reduction of finding $[\boldsymbol{A}(\boldsymbol{x})]^{-1}$ once instead of twice. The value of b_0, the value imposed on the zeroth moment, can be chosen freely for the approximation of the even derivative. It cancels out because of the difference term in the definition of the operator. This extra degree of freedom of DC PSE operators can be useful in practice as it enables enlarging the stability region of the operator in diffusion problems, see §2.3.3.
Setting b_0 to zero has the advantage that the operator can also be evaluated at off-particle locations. The operator then becomes equivalent to

$$Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{1}{\varepsilon(\boldsymbol{x})^{|\boldsymbol{\beta}|}} \sum_{p \in \mathcal{N}(\boldsymbol{x})} f(\boldsymbol{x}_p) \eta_\varepsilon^{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x}). \tag{2.29}$$

In fact, setting the zeroth-order moment to zero and evaluating the operators at off-particle locations makes DC PSE a particle-analog of derivative-reproducing kernel (DRK) Galerkin collocation methods [14, 84, 88], which are conceptually related to Moving Least-Squares (MLS) schemes [7, 51].

2.8 A DC Operator for Interpolation between Two Scattered Sets of Particles

Using above operators (2.29) that can be evaluated at off-particle locations, it is also possible to approximate the function f itself at arbitrary locations between particles. We can thus construct particle-particle interpolation schemes where the linear system of equations that has the same left-hand side as the one for the derivative approximations, see §2.7 for details. Interpolating f_p from one set of irregularly distributed particles to another, however, in addition requires the kernel to satisfy the Kronecker delta property at the particle locations. Otherwise, the interpolated field f is a smoothed version of the original one.
For a noninterpolating approximation of f at any collocation point $\boldsymbol{x}$, one can use the kernel function $\eta_\varepsilon^{\boldsymbol{0}}$ as in equation (2.26) whose unknown

coefficients $\boldsymbol{a}(\boldsymbol{x})$ are to be determined by solving the linear system of equations (2.27) with $\boldsymbol{b} = \boldsymbol{p}(\boldsymbol{0})$, $\alpha_{\min} = 0$. Interpolating kernel functions that fulfill the Kronecker delta property, that is, kernel functions ζ for which

$$f^h(\boldsymbol{x}_p) = \sum_q f_q \zeta_\varepsilon(\boldsymbol{x}_p - \boldsymbol{x}_q, \boldsymbol{x}_p) = f_p \text{ for all } p,$$

can also be constructed by reusing the same matrix $\boldsymbol{A}$ and its Cholesky decomposition (or inverse). Following the idea introduced by Chen and coworkers [14], we obtain interpolating kernels by expressing ζ as the sum of the noninterpolating kernel $\bar{\zeta}$ and a correction function $\hat{\zeta}$, thus:

$$\zeta_\varepsilon(\boldsymbol{z}, \boldsymbol{x}) = \bar{\zeta}_\varepsilon(\boldsymbol{z}, \boldsymbol{x}) + \hat{\zeta}_a(\boldsymbol{z}, \boldsymbol{x}).$$

The noninterpolating kernels $\bar{\zeta}_\varepsilon(\boldsymbol{z}, \boldsymbol{x})$, based on a template as in equation (2.26), are obtained by solving the system (2.27) with the right-hand side

$$\bar{\boldsymbol{b}}^T(\boldsymbol{x}) = \boldsymbol{p}(\boldsymbol{0}) - \sum_p \boldsymbol{p}\left(\frac{\boldsymbol{x} - \boldsymbol{x}_p}{\varepsilon(\boldsymbol{x})}\right) \hat{\zeta}_a(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x})$$

with smooth correction functions

$$\hat{\zeta}_a(\boldsymbol{z}, \boldsymbol{x}) = \hat{\zeta}\left(\frac{\boldsymbol{z}}{a(\boldsymbol{x})}, \boldsymbol{x}\right)$$

satisfying $\hat{\zeta}_a(\boldsymbol{x}_p - \boldsymbol{x}_q, \boldsymbol{x}_p) = \delta_{pq}$, where δ_{pq} is the Kronecker delta. The resulting kernels ζ_ε satisfy the moment conditions for $\boldsymbol{b}^T = \boldsymbol{p}(\boldsymbol{0})$, which ensures that the approximation is consistent, as well as the Kronecker delta property $\zeta_\varepsilon(\boldsymbol{x}_p - \boldsymbol{x}_q, \boldsymbol{x}_p) = \delta_{pq}$, which ensures that the approximation is interpolating. This can be seen by expressing ζ_ε as

$$\begin{aligned}\zeta_\varepsilon(\boldsymbol{z}, \boldsymbol{x}) &= \boldsymbol{p}\left(\frac{\boldsymbol{z}}{\varepsilon(\boldsymbol{x})}\right) \boldsymbol{a}^{\mathrm{T}}(\boldsymbol{x}) \exp\left(-\frac{|\boldsymbol{z}|^2}{\varepsilon(\boldsymbol{x})^2}\right) + \hat{\zeta}_a(\boldsymbol{z}, \boldsymbol{x}) \\ &= \boldsymbol{p}\left(\frac{\boldsymbol{z}}{\varepsilon(\boldsymbol{x})}\right) [\boldsymbol{A}(\boldsymbol{x})]^{-1} \bar{\boldsymbol{b}}^T(\boldsymbol{x}) \exp\left(-\frac{|\boldsymbol{z}|^2}{\varepsilon(\boldsymbol{x})^2}\right) + \hat{\zeta}_a(\boldsymbol{z}, \boldsymbol{x})\end{aligned}$$

$$= \boldsymbol{p}\left(\frac{\boldsymbol{z}}{\varepsilon(\boldsymbol{x})}\right)[\boldsymbol{A}(\boldsymbol{x})]^{-1}\left(\boldsymbol{p}(\boldsymbol{0}) - \sum_r \boldsymbol{p}\left(\frac{\boldsymbol{x}-\boldsymbol{x}_r}{\varepsilon(\boldsymbol{x})}\right)\hat{\zeta}_a(\boldsymbol{x}-\boldsymbol{x}_r,\boldsymbol{x})\right)$$
$$\times \exp\left(-\frac{|\boldsymbol{z}|^2}{\varepsilon(\boldsymbol{x})^2}\right) + \hat{\zeta}_a(\boldsymbol{z},\boldsymbol{x}).$$

For $\boldsymbol{z} = \boldsymbol{x}_p - \boldsymbol{x}_q$ and $\boldsymbol{x} = \boldsymbol{x}_p$, $\zeta_\varepsilon(\boldsymbol{z},\boldsymbol{x})$ becomes

$$\zeta_\varepsilon(\boldsymbol{x}_p - \boldsymbol{x}_q, \boldsymbol{x}_p) = \boldsymbol{p}\left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{\varepsilon(\boldsymbol{x}_p)}\right)[\boldsymbol{A}(\boldsymbol{x}_p)]^{-1}\left(\boldsymbol{p}(\boldsymbol{0}) - \sum_r \boldsymbol{p}\left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_r}{\varepsilon(\boldsymbol{x}_p)}\right)\delta_{pr}\right)$$
$$\times \exp\left(-\frac{|\boldsymbol{x}_p - \boldsymbol{x}_q|^2}{\varepsilon(\boldsymbol{x}_p)^2}\right) + \delta_{pq}$$
$$= \delta_{pq}.$$

A simple choice for the correction functions $\hat{\zeta}$ is a smooth function of finite support in the unit ball. Setting $a(\boldsymbol{x})$ to a value smaller than the distance to the nearest neighbor of $\boldsymbol{x}$ yields $\hat{\zeta}_a(\boldsymbol{x}-\boldsymbol{x}_p,\boldsymbol{x}) = 0$, $\boldsymbol{x} \neq \boldsymbol{x}_p$. Normalization assures that $\hat{\zeta}_a(\boldsymbol{0},\boldsymbol{x}) = 1$.[5] With some assumptions about the smoothness of f and the regularity of the particle distribution, bounds for the local approximation errors can be expressed in terms of the local interparticle spacing h_p, and the magnitude of the derivatives of f. Chen and coworkers [14] report the following error estimate for the interpolant $f^h(\boldsymbol{x}) = \sum_p f(\boldsymbol{x}_p)\zeta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p,\boldsymbol{x})$:

$$\left|f - f^h\right|_{\boldsymbol{x}=\boldsymbol{x}_p} \leq C h_p^r \, |f|_{r,\infty,B_{r_c(\boldsymbol{x}_p)}} \tag{2.30}$$

with the semi-norm $|f|_{r,\infty,\Omega} = \max_{|\boldsymbol{\alpha}|=r} \|D^{\boldsymbol{\alpha}} f\|_{L^\infty(\Omega)}$.

2.9 Conclusions

We have presented and analyzed a systematic framework for discretization correction of general integral PSE operators [31] for approximating n-dimensional spatial derivatives of any degree. This was made possible by considering the total approximation error, thus combining the mollifica-

[5] As Wang and coworkers [84], we take $\hat{\zeta}$ to be the quartic spline with cutoff radius 1 for the simulations presented in §5.3. We choose $a(\boldsymbol{x}) = 0.9 \min_p |\boldsymbol{x} - \boldsymbol{x}_p|$, $\boldsymbol{x} \neq \boldsymbol{x}_p$.

tion and discretization errors. The discrete moments of the kernel function then naturally appeared in the error expansion, and DC operators could be constructed based on the corresponding discrete moment conditions.
We have analyzed DC PSE operators with respect to their accuracy, computational efficiency, rate of convergence, and stability on regular and irregular particle distributions as well as near boundaries. We have demonstrated that DC PSE operators achieve the desired rate of convergence in all cases, whereas the convergence of UC operators is hampered by the constant discretization error. The computational efficiency is mainly determined by the cutoff radius of the operators and the ratio c. DC PSE operators are more efficient than UC ones since they reach the same level of accuracy with smaller cutoff radii. Moreover, the relaxed overlap condition of DC PSE operators allows higher values of c, reducing the number of particles within the operator support. For advection-dominated problems, the efficiency of DC PSE operators can exceed that of Lagrangian FD schemes because they require less frequent remeshing and hence are more accurate (less remeshing error). In these cases, the computational cost of determining the position-dependent correction functions is amortized by this gain in accuracy. If remeshing is done at every time step, the remeshing error dominates the error of the diffusion operator and FD schemes are more efficient.
DC operators introduce less numerical diffusion and dispersion into the solution than UC ones. In addition, the numerical diffusion and dispersion of DC PSE operators can be further reduced by increasing c, without affecting the rate of convergence. We have derived analytical expressions for the modified wavenumbers of the one-dimensional wave, convection, and diffusion equations both for uniform Cartesian and random particle distributions. Based on these wavenumbers, we presented CFL-like stability conditions for different time stepping schemes.
The main practical limitation of the presented DC PSE operators lies in determining the DC kernel. The coefficients of the DC kernel have to be determined by solving a linear system of equations. We found the approximation error to be sensitive to numerical inaccuracies in the coefficients of the DC kernel. These numerical inaccuracies typically grow with increasing order of convergence, limiting the practical use of high-order operators. Also, the exact conservativeness of UC full-space PSE operators is lost for DC PSE operators on nonuniform particle distributions and near boundaries for $r > 1$. Finally, the computational cost of computing the DC

kernel functions on irregular particle distributions is significant, since the correction function has to be determined for each particle individually. The possibility of reusing the solution of the linear system of equations to construct DC kernels that approximate also other derivatives or interpolate between sets of scattered particles, though, may lessen the significance of this additional cost on more complex problems.

Taken together, our results provide a framework for discretization correction of PSE-type operators. As opposed to UC PSE operators, DC operators yield the design rate of convergence over the entire range of resolutions as well as on irregular particle distributions and near boundaries. Moreover, they offer more freedom in choosing optimal kernel parameters due to the relaxed overlap condition. This can lead to accuracy and efficiency exceeding those of Lagrangian FD schemes.

CHAPTER

THREE

Relatives of DC PSE Operators

The discretization correction framework introduced in the previous chapter enables us to show the relationships between DC PSE operators and other operators used to approximate derivatives. We particularly highlight the connections to FD stencils, corrected SPH [10, 46, 52–54, 68, 71], operators used in reproducing kernel particle methods (RKPM) [56, 57], differential reproducing kernel (DRK) [84, 88] and moving least squares (MLS)[6, 51, 67] approximations, and operators used in vorticity redistribution methods [38, 80].
For certain parameter choices, DC PSE operators become equivalent or similar to operators used in other numerical methods. We prove that certain classical FD stencils are limit cases of DC PSE operators for $c \to \infty$.

3.1 Finite Differences

For uniform Cartesian particle distributions with spacing h and a finite operator support of radius r_c, the discrete integral operator (1.10) can be written as

$$Q_h^{\boldsymbol{\beta}} f(\boldsymbol{x}) = \frac{c^n}{\varepsilon^{|\boldsymbol{\beta}|}} \sum_{|\boldsymbol{k}|^2=0}^{\lfloor r_c^2/h^2 \rfloor} \left(f(\boldsymbol{x}+\boldsymbol{k}h) \pm f(\boldsymbol{x})\right) \eta^{\boldsymbol{\beta}}(-c\boldsymbol{k}), \quad \boldsymbol{k} \in \mathbb{Z}^n . \qquad (3.1)$$

Using the kernel template (2.5), the value of the DC kernel function at $-c\boldsymbol{k}$ is

$$\eta^{\boldsymbol{\beta}}(-c\boldsymbol{k}) = \left(\sum_{\substack{|\boldsymbol{\gamma}|=\alpha_{\min} \\ \boldsymbol{\beta}+\boldsymbol{\gamma} \text{ even}}}^{|\boldsymbol{\beta}|+r-1} a_{\boldsymbol{\gamma}} (-c\boldsymbol{k})^{\boldsymbol{\gamma}} \right) \exp\left(-c^2|\boldsymbol{k}|^2\right) \tag{3.2}$$

and the discrete moments become

$$Z_h^{\boldsymbol{\alpha}} = c^n \sum_{|\boldsymbol{k}|^2=0}^{\lfloor r_c^2/h^2 \rfloor} \sum_{\substack{|\boldsymbol{\gamma}|=\alpha_{\min} \\ \boldsymbol{\beta}+\boldsymbol{\gamma} \text{ even}}}^{|\boldsymbol{\beta}|+r-1} a_{\boldsymbol{\gamma}} (c\boldsymbol{k})^{\boldsymbol{\alpha}+\boldsymbol{\gamma}} \exp\left(-c^2|\boldsymbol{k}|^2\right) . \tag{3.3}$$

Here, "$\boldsymbol{\beta}+\boldsymbol{\gamma}$ even" stands for all multiindices $\boldsymbol{\gamma}$ for which $\boldsymbol{\beta}+\boldsymbol{\gamma}$ contains only even elements. All other $\boldsymbol{\gamma}$ need not be considered since the corresponding coefficients $a_{\boldsymbol{\gamma}}$ can priorly be set to zero (see §2.2).
The DC PSE operators for $c \to \infty$ can be derived from equations (3.1) to (3.3) and the moment conditions (2.4): the unknown coefficients $a_{\boldsymbol{\gamma}}$ of the kernel function (3.2) are determined by inserting the discrete moments (3.3) into the conditions (2.4) and solving the resulting linear system of equations. The resulting kernel function is then used in definition (3.1) and the limit $c \to \infty$ is taken.
For the second-order accurate DC PSE operator approximating the first derivative along dimension i ($r = 2$, $\boldsymbol{\beta} = \boldsymbol{e}_i$), for example, the only unknown coefficient $a_{\boldsymbol{e}_i}$ is determined by the equation

$$c^n \sum_{|\boldsymbol{k}|^2=0}^{\lfloor r_c^2/h^2 \rfloor} a_{\boldsymbol{e}_i} c^2 k_i^2 \exp\left(-c^2|\boldsymbol{k}|^2\right) = -1 \qquad (\boldsymbol{\alpha} = \boldsymbol{\beta} = \boldsymbol{e}_i).$$

The equations for $\boldsymbol{\alpha} = \boldsymbol{0}$, $\boldsymbol{\alpha} = \boldsymbol{e}_j$, $j \neq i$, and $|\boldsymbol{\alpha}| = 2$ vanish since they are automatically satisfied by the symmetry of the particle distribution. The DC kernel function can thus be written as

$$\eta^{\boldsymbol{e}_i}(-c\boldsymbol{k}) = \frac{k_i \exp\left(-c^2|\boldsymbol{k}|^2\right)}{c^{n+1} \sum_{|\boldsymbol{l}|^2=0}^{\lfloor r_c^2/h^2 \rfloor} l_i^2 \exp\left(-c^2|\boldsymbol{l}|^2\right)} . \tag{3.4}$$

Using this kernel, the operator (3.1) becomes

$$Q_h^{\boldsymbol{e}_i} f(\boldsymbol{x}) = \frac{\sum_{|\boldsymbol{k}|^2=0}^{\lfloor r_c^2/h^2 \rfloor} \left(f(\boldsymbol{x}+\boldsymbol{k}h) + f(\boldsymbol{x})\right) k_i \exp\left(-c^2|\boldsymbol{k}|^2\right)}{h \sum_{|\boldsymbol{k}|^2=0}^{\lfloor r_c^2/h^2 \rfloor} k_i^2 \exp\left(-c^2|\boldsymbol{k}|^2\right)} \,. \tag{3.5}$$

This is a FD stencil with extent and weights that can be adjusted by the choice of the cutoff radius r_c and the ratio c. Letting $c \to \infty$ yields

$$\lim_{c\to\infty} Q_h^{\boldsymbol{e}_i} f(\boldsymbol{x}) = \frac{f(\boldsymbol{x}+\boldsymbol{h}_i) - f(\boldsymbol{x}-\boldsymbol{h}_i)}{2h} \,, \quad \boldsymbol{h}_i = h\boldsymbol{e}_i \,, \tag{3.6}$$

for any value of $r_c \geq h$. This is the classical centered difference stencil for the first derivative of f.
Following the same procedure, the second-order DC PSE operator approximating the Laplacian $\nabla^2 f(\boldsymbol{x})$ becomes

$$\begin{aligned} \lim_{c\to\infty} Q_h^{\mathrm{Lap}} f(\boldsymbol{x}) &= \lim_{c\to\infty} \sum_{i=1}^{n} Q_h^{2\boldsymbol{e}_i} f(\boldsymbol{x}) \\ &= \frac{\sum_{i=1}^{n} \left[f(\boldsymbol{x}+\boldsymbol{h}_i) - 2f(\boldsymbol{x}) + f(\boldsymbol{x}-\boldsymbol{h}_i)\right]}{h^2} \end{aligned} \tag{3.7}$$

and the fourth-order DC PSE approximation of the first derivative along $\boldsymbol{e}_i$ yields

$$\lim_{c\to\infty} Q_h^{\boldsymbol{e}_i} f(\boldsymbol{x}) = \frac{-f(\boldsymbol{x}+2\boldsymbol{h}_i) + 8f(\boldsymbol{x}+\boldsymbol{h}_i) - 8f(\boldsymbol{x}-\boldsymbol{h}_i) + f(\boldsymbol{x}-2\boldsymbol{h}_i)}{12h} \,. \tag{3.8}$$

The limit FD stencil of the n-dimensional anisotropic diffusion operator of order $r = 2$, introduced by Degond and Mas-Gallic [27] to approximate $\nabla \cdot (\boldsymbol{L}(\boldsymbol{x})\nabla f(\boldsymbol{x}))$, can be found in appendix B. All these classical compact FD stencils can hence be interpreted as DC PSE operators on uniform Cartesian particle distributions with a kernel width ε tending to zero (particles become grid points).
For irregular particle distributions, the DC PSE operators can be made equivalent to FD stencils for irregular meshes. Demkowicz and coworkers [29] proposed simple FD stencils approximating the two-dimensional Laplacian on irregular meshes by solving a linear system of equations consisting of moment conditions analogous to the ones used here for DC PSE

operators. The value $\eta^{\boldsymbol{\beta}}(\boldsymbol{x}-\boldsymbol{x}_p,\boldsymbol{x})$ of the DC PSE kernel is equivalent to the FD stencil weight α_p for particle (mesh node) p.
Wright and Fornberg [87] used radial basis functions to generate compact FD stencils on irregularly distributed nodes. For Cartesian node distributions, they recover the classical one- and two-dimensional FD stencils for first and second derivatives. These stencils were obtained in the limit of the radial basis function becoming flat, thus $c \to 0$. This was possible because the support of the stencils was chosen and fixed beforehand. The DC PSE operator (3.1) becomes equal to the respective FD stencil, equations (3.6) to (3.8), if one fixes r_c to the radius of the stencil. The limit $c \to \infty$ in our case, however, makes it unnecessary to fix r_c beforehand. Instead, compact FD stencils naturally emerge.

3.2 (Corrected) Smoothed Particle Hydrodynamics

In PSE operators, the kernel is weighted by a sum or difference of field values, which guarantees conservativeness of DC PSE operators on uniform symmetric particle distributions. The operators used in SPH lack this symmetry. Rather, the kernel is weighted only by the field values evaluated at the particle locations:

$$Q^{\boldsymbol{\beta}}_{h,\mathrm{SPH}} f(\boldsymbol{x}) = \frac{1}{\varepsilon^{|\boldsymbol{\beta}|}} \sum_{p\in\mathcal{N}(\boldsymbol{x})} v_p f(\boldsymbol{x}_p) \eta^{\boldsymbol{\beta}}_{\varepsilon,\mathrm{SPH}}(\boldsymbol{x}-\boldsymbol{x}_p)\,. \tag{3.9}$$

Thus, the operator is equivalent to a PSE operator with a kernel function that has a zeroth moment of zero. This operator achieves r^{th}-order convergence for all kernels $\eta^{\boldsymbol{\beta}}_{\varepsilon,\mathrm{SPH}}$ that satisfy the discrete moment conditions (2.4) for $\alpha_{\min} = 0$. In classical SPH, however, like for the UC PSE operators, the kernel is designed to satisfy continuous moment conditions, and the above ones are not automatically fulfilled.
Johnson and Beissel [46] therefore introduced normalized smoothing functions for SPH. There, $\eta^{\boldsymbol{\beta}}_{\varepsilon,\mathrm{SPH}}$ is normalized with a scalar factor that depends on $\boldsymbol{x}$ such that the first of the conditions (2.4) is met. This is reminiscent of the corrected kernel function (3.4), in which the normalization factor is equal to the discrete moment $Z^{\boldsymbol{e}_i}_h$. Normalized SPH kernels were shown to lead to more accurate derivative estimations than unnormalized ones,

especially near boundaries and on nonuniformly distributed particles [46]. Randles and Libersky [71] have extended the normalization idea to the estimation of the divergence of general tensor fields.
Bonet and Kulasegaram [10] derived second-order accurate SPH kernels for approximating a field, and first-order accurate kernels for approximating its Laplacian. These operators involve a linear correction function for the kernel, similar to the polynomial correction function proposed in §2.2. The coefficients of the correction function are chosen such as to satisfy the conditions (2.4) for $|\boldsymbol{\alpha}| \leq |\boldsymbol{\beta}|$, hence $r = 1$ [10].
Lanson and Vila [52–54] proposed an altered normalization that leads to a conservative scheme for approximating first derivatives. This correction involves the PSE-like idea of replacing $f(\boldsymbol{x}_p)$ in equation (3.9) by the difference $f(\boldsymbol{x}_p) - f(\boldsymbol{x})$, such that the condition for $Z_h^{\mathbf{0}}$ vanishes. The average of the renormalization matrices at $\boldsymbol{x}$ and $\boldsymbol{x}_p$ is used as an additional weight. The resulting operators are first-order accurate and resemble DC PSE operators for even $|\boldsymbol{\beta}|$. There is, however, no trivial conservative generalization of these operators to higher derivatives or higher orders of accuracy.

3.3 Reproducing Kernel Particle Methods & Moving Least Squares Approximations

In RKPM [56, 57], a field $f(\boldsymbol{x})$ is represented using the particle function approximation

$$Q_{h,\mathrm{RKPM}}^{\mathbf{0}} f(\boldsymbol{x}) = \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p f(\boldsymbol{x}_p) \eta_{\varepsilon,\mathrm{RKPM}}(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x}) \,.$$

The order of accuracy is given by the largest integer r for which the interpolation kernel $\eta_{\varepsilon,\mathrm{RKPM}}$ fulfills the moment conditions (2.4) with $\boldsymbol{\beta} = \mathbf{0}$. This is achieved by multiplying a window function by a polynomial correction function, thus forming the kernel η_{RKPM} similarly to the kernel (2.5). Just as corrected SPH operators, RKPM operators are not symmetric, rendering them generally nonconservative. The $\boldsymbol{\beta}^{\mathrm{th}}$ derivative of the field is evaluated with an order of accuracy of $r - |\boldsymbol{\beta}|$ by taking the $\boldsymbol{\beta}^{\mathrm{th}}$ derivative of the continuously differentiable kernel function $\eta_{\varepsilon,\mathrm{RKPM}}$.
When setting all particle volumes to $v_p \equiv 1$ (that is, volumes are disregarded), RKPM become equivalent to moving least squares (MLS) ap-

proximations [6, 7, 35, 51, 67]. In MLS, nonuniform particle distributions (holes and clusters) are thus implicitly accounted for in the kernel function (shape function), in the same manner as they are accounted for by the DC PSE kernel when applied in the strong formulation.

3.4 Differential Reproducing Kernels

DRK [84, 88], designed for problems in computational mechanics, are used to approximate a derivative of a field $f(\boldsymbol{x})$ by

$$Q^{\boldsymbol{\beta}}_{h,\mathrm{DRK}} f(\boldsymbol{x}) = \sum_{p \in \mathcal{N}(\boldsymbol{x})} f(\boldsymbol{x}_p) \eta_{\varepsilon,\mathrm{DRK}}(\boldsymbol{x} - \boldsymbol{x}_p, \boldsymbol{x}),$$

where the kernel fulfills the discrete moment conditions (2.4) of the DC PSE kernels with $\alpha_{\min} = 0$ and $v_p \equiv 1$. They are thus equivalent to DC PSE operators in the strong formulation and a zeroth moment fixed to zero.
Opposed to differentiation in MLS methods or RKPM, the DRK are not built by differentiating the kernels used for the function approximation, but are derived independently via the reproducing conditions.

3.5 Vorticity Redistribution Schemes

Shankar and van Dommelen [80] introduced a DC vorticity redistribution method to approximate the diffusion term in vortex methods for two-dimensional incompressible flows. The diffusion of the vorticity field f, $\mathrm{D}f/\mathrm{D}t = \nu \nabla^2 f$, is approximated by changing the field values

$$f^n(\boldsymbol{x}) = \sum_p F^n_p \phi_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \tag{3.10}$$

at time step n to

$$f^{n+1}(\boldsymbol{x}) = \sum_{p,q} w^n_{pq} F^n_p \phi_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_q) \tag{3.11}$$

at time step $n+1$. The weights w_{pq}^n are chosen such that

$$\sum_q \left(\frac{\boldsymbol{x}_q - \boldsymbol{x}_p}{h_\nu}\right)^{\boldsymbol{\alpha}} w_{pq}^n = \begin{cases} 1, & \boldsymbol{\alpha} = \boldsymbol{0}, \\ 2, & \boldsymbol{\alpha} = 2\boldsymbol{e}_i, \\ 0, & |\boldsymbol{\alpha}| \leq r+2 \text{ (except cases above)} \end{cases} \tag{3.12}$$

is true at each particle p. Here, $h_\nu = \sqrt{\nu \Delta t}$ is half the characteristic diffusion length. This is equivalent to the diffusion operator $Q_h(\cdot)$ defined by

$$Q_h f(\boldsymbol{x}) = \left(f^{n+1}(\boldsymbol{x}) - f^n(\boldsymbol{x})\right)/\Delta t$$

when assuming explicit Euler time stepping. Substituting equations (3.10) and (3.11), expanding $\phi(\boldsymbol{x} - \boldsymbol{x}_q)$ around $\boldsymbol{x} - \boldsymbol{x}_p$, and taking into account conditions (3.12), one can rewrite the definition of the operator as

$$\begin{aligned} Q_h f(\boldsymbol{x}) &= \sum_{p,q} w_{pq}^n F_p^n \sum_{|\boldsymbol{\alpha}|=0}^{r+1} \frac{D^{\boldsymbol{\alpha}} \phi_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p)(\boldsymbol{x}_p - \boldsymbol{x}_q)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}! \Delta t} \\ &\quad + \sum_{p,q} w_{pq}^n F_p^n \sum_{|\boldsymbol{\alpha}|=r+2} \frac{R_{\boldsymbol{\alpha}}^{\phi_\varepsilon}(\boldsymbol{x}_p)(\boldsymbol{x}_p - \boldsymbol{x}_q)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}! \Delta t} - \sum_p F_p^n \frac{\phi_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p)}{\Delta t} \\ &= \nu \sum_p F_p^n \nabla^2 \phi_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) + \sum_{|\boldsymbol{\alpha}|=r+2} \sum_{p,q} w_{pq}^n (\boldsymbol{x}_p - \boldsymbol{x}_q)^{\boldsymbol{\alpha}} F_p^n \frac{R_{\boldsymbol{\alpha}}^{\phi}(\boldsymbol{x}_p)}{\boldsymbol{\alpha}! \Delta t} \\ &= \nu \nabla^2 f(\boldsymbol{x}) + \nu h_\nu^r \sum_{|\boldsymbol{\alpha}|=r+2} \sum_p \frac{F_p^n R_{\boldsymbol{\alpha}}^{\phi_\varepsilon}(\boldsymbol{x}_p)}{\boldsymbol{\alpha}!} \sum_q w_{pq}^n \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}_q}{h_\nu}\right)^{\boldsymbol{\alpha}}, \end{aligned}$$

where $R_{\boldsymbol{\alpha}}^{\phi_\varepsilon}(\boldsymbol{x}_p) = D^{\boldsymbol{\alpha}} \phi_\varepsilon(\boldsymbol{\zeta})$, $\zeta \in [0, |\boldsymbol{x} - \boldsymbol{x}_p|]$, for radially symmetric blob shapes ϕ_ε. Thus, choosing h proportional to h_ν, the operator $Q_h(\cdot)$ converges to $L(\cdot)$ as h tends to zero, meaning that the operator is DC. It is interesting to note that the conditions (3.12) were originally derived by requiring all finite wavenumbers of the Fourier transform of f to be correctly damped. Shankar and van Dommelen remark that these conditions "are similar to the equations obtained when a Taylor series expansion of the exact solution is substituted into a finite difference formula, or to the moment conditions in the particle methods. In fact, consistency of a finite difference scheme requires the same agreement for finite wavenumbers (e.g., [82],

(10.1.3)). For uniform point spacing and [weights w_{pq}^n], the redistribution method is equivalent to an explicit finite difference scheme" [80].
An important difference to the DC PSE operators is that in the vorticity redistribution scheme the linear system of equations for the unknown weights w_{pq}^n is generally not square: the number of unknowns is the number of neighbors of the particle under consideration and the number of equations in (3.12) is determined by the order of accuracy r. Therefore, the system is typically underdetermined. For stability of the redistribution method it is desirable that $w_{pq}^n \geq 0$. In order to find a nonnegative solution to the underdetermined system, Shankar and van Dommelen apply a least maximum solution procedure, and in cases where no acceptable solution exists they insert additional particles until a solution exists. While their method leads to a strictly positive solution, solving the problem using a linear programming approach leads to redistribution only among the minimum number of particles (some weights w_{pq}^n are set to zero). Shankar and van Dommelen argue that "the least maximum procedure tends to spread out the vorticity somewhat better."
While Shankar and van Dommelen formulated the redistribution scheme for arbitrarily high-order accuracy, they tested only the first-order accurate version. Gharakhani [38] introduced and tested a vorticity redistribution scheme of third order. The scheme designed and tested by Gharakhani corresponds to a fourth-order centered finite-difference approximation of the Laplacian and a second-order Runge-Kutta (modified Euler) time integration if applied on a uniform Cartesian particle distribution.

CHAPTER

FOUR

Objective Rating of Operator Properties

4.1 Introduction

Frequently, many different numerical methods exist for the solution of a given computational problem. Their design mostly aims at objectives like ease of implementation, high accuracy, robustness, or low computational cost. Typically, a compromise between different desirable method properties must be found. This results in a pool of methods designed to do essentially the same, but with different strengths and weaknesses. Usually, the goal is to select – potentially under certain constraints – the computationally most efficient method for the task at hand. In order to compare the methods, their performance needs to be quantified. The quantification is usually based on specific test cases. This renders the quantification *problem-dependent*. In this chapter, we explore the properties of diffusion operators used in particle methods and demonstrate a way to model their performance in a *problem-independent* way.

The (approximate) evaluation of the diffusion operator $L(\cdot) = \nabla \cdot (\nu \nabla(\cdot))$ is commonly required in particle methods. In viscous vortex methods, for example, the method of random walks [17] or of deterministic displacements [28] for representing the diffusion of vorticity is now mostly superseded by more accurate methods where vorticity is exchanged or redistributed among particles. These methods, such as resampling methods [23, 72], redistribution methods [1, 38, 75, 80], or methods of particle

strength exchange (PSE) [26, 27, 33, 62], estimate the diffusion operator deterministically based on the field derivatives. Also in smoothed particle hydrodynamics, deterministic diffusion operators predominate [65]. Due to differing design criteria of the aforementioned methods, many different diffusion operators exist. Some operators have free parameters which have been shown to have a big influence on their properties, see §2.3, §2.4 and [16, 31, 69].
We introduce three problem-independent measures to quantify an operator's accuracy, computational cost, and stability in explicit time stepping. These measures enable comparing different operators and ranking their properties. They also allow for parameter studies to assess the influence of tunable operator parameters on the operator performance. We show examples of how these quality measures can be used to predict the computational efficiency of an operator in a simulation. The quality measures are applicable to different types of diffusion operators, as we demonstrate for the examples of PSE and Fishelov diffusion operators, and diffusion operators used in a resampling method and in smoothed particle hydrodynamics. This renders different types of diffusion operators comparable in an objective way without relying on empirical comparisons and provides a tool for both operator choice and operator design.

4.2 The Quality Measures

We introduce problem-independent quality measures such that different operator types or operators with different parameter settings can be objectively compared and the operator performance in specific test cases predicted. We define the important qualities of a diffusion operator as

(i) its accuracy,

(ii) its stability in explicit time-stepping, and

(iii) its computational cost.

In the following, we define three measures reflecting these qualities.
We call the measures introduced here problem-independent because they do not depend on test-case-specific parameters such as the viscosity field ν or the field f the operator is applied to. Also, they do not depend on numerical parameters including spatial and temporal resolution.

4.2.1 Quantifying Accuracy

We define the problem-independent accuracy measures A_s, $s \leq r$, such that the overall error

$$\epsilon(\boldsymbol{x}) = Q_h f(\boldsymbol{x}) - Lf(\boldsymbol{x}) \tag{4.1}$$

is bounded by

$$|\epsilon(\boldsymbol{x})| \leq \sum_{s=-2}^{r} \frac{h^s}{A_s(\boldsymbol{x})} \|f\|_{s+2,\infty,B_{r_c}[\boldsymbol{x}]} \|\nu\|_{s+1,\infty,B_{r_c}[\boldsymbol{x}]}. \tag{4.2}$$

The Sobolev norms $\|\cdot\|_{m,\infty,\Omega}$ are defined as

$$\|\cdot\|_{m,\infty,\Omega} = \max_{\substack{0 \leq |\boldsymbol{\alpha}| \leq m \\ \boldsymbol{y} \in \Omega}} |D^{\boldsymbol{\alpha}}(\cdot(\boldsymbol{y}))|.$$

For the anisotropic case, we substitute the norm $\|\nu\|_{m,\infty,\Omega}$ in error bound (4.2) by

$$\max_{\substack{0 \leq |\boldsymbol{\alpha}| \leq m \\ \boldsymbol{y} \in \Omega}} \|D^{\boldsymbol{\alpha}} \boldsymbol{\nu}(\boldsymbol{y})\|_{\mathrm{F}},$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm.
The accuracies $A_s(\boldsymbol{x})$, $s < r$, should be large compared to A_r, such that the error terms of order $O(h^s)$ are negligible even for small h. Typically, a larger $A_r(\boldsymbol{x})$ implies a smaller error at position $\boldsymbol{x}$.
Closed-form expressions for the measures A_s are given in appendix C.

4.2.2 Quantifying Stability

In order to quantify the stability of a diffusion operator, we define a problem-independent stability measure S such that the time-step limit in explicit Euler time stepping for solving

$$\frac{\partial f(\boldsymbol{x}, t)}{\partial t} = Q_h f(\boldsymbol{x}, t) \tag{4.3}$$

is given by

$$\Delta t < S\frac{h^2}{\nu}. \tag{4.4}$$

That is, the CFL number $\alpha_{\text{CFL}} = \Delta t\nu/h^2$ must be smaller than S if the corresponding operator is used in explicit Euler time stepping. The stability S also indicates trends of the time-step limit when using other explicit methods for time integration. When using Heun's method, for example, the time-step limit is increasing with increasing S (for $S > 0$), but the relationship is not linear and depends on additional characteristics of the operator.
Closed-form expressions for the stability measure S for different diffusion operators are given in appendix C. They are derived assuming constant viscosity.

4.2.3 Quantifying Computational Cost

We define

$$C = V_n \left(\frac{r_c}{h}\right)^n$$

as a problem-independent measure of the computational cost of the operator. Here, V_n is the volume of the n-dimensional unit sphere and r_c is the radius of the operator support. Thus, the measure C merely approximates the number of neighbors of each particle. We do not account for the cost of computing the weights needed for the operator evaluation because lookup-tables are commonly used to avoid function evaluations. Also, the construction of the operator is not taken into account because in some applications it is done just once in the beginning of the computations.

4.3 The Operator Choice

How to choose a suitable operator for a specific problem? Or, given a certain operator type, to which values should the parameters be fixed? In this section, we provide some insight into the influence of the parameters of operators. We first present a parameter study for UC and DC PSE operators. Then, in §4.3.2, we present two examples of operator choices for test problems based on the quality measures introduced above.

4.3.1 Parameter Study: How do the Free Operator Parameters Influence the Quality of the PSE operators?

The PSE diffusion operator $Q_h(\cdot)$ defined in equation (1.3) is determined by the choice of the kernel function η, of the cutoff radius r_c, and of the ratio $c = h/\varepsilon$. One objective here is to demonstrate the effect of these parameters on the operator properties, providing a guideline for their choice. Based on the quality measures introduced in §4.2, we therefore assess the diffusion operators as defined by different parameter sets. The parameters are described in §4.3.1.1 and the results of the parameter study are presented in §4.3.1.2.
The quality measures as presented in §4.2 depend on the underlying spatial distribution of particles. For simplicity, we restrict the present study to uniform Cartesian particle distributions. Also, we only consider operators used for three-dimensional simulations with isotropic viscosity. The quality measures introduced are, however, designed for the general case (any particle distribution, any number of dimensions n).

4.3.1.1 Parameters

In the following, we describe the free parameters and state the ranges within which these parameters are varied in the parameter study.

The Kernel Function η We restrict our study to kernel functions of the form

$$\eta(\boldsymbol{z}) = K_i(z)\phi_j(z), \quad z = |\boldsymbol{z}| = \left[\sum_{k=1}^{n} z_k^2\right]^{1/2}, \tag{4.5}$$

where K_i is called the correction function and ϕ_j the window function. In the present study, we consider ϕ_j and K_i as parameters of the diffusion

operator and test the following options:

$$\begin{aligned}
\phi_1(z) &= \exp\left(-z^2\right), & \phi_2(z) &= \frac{1}{z^{10}+1},\\
\phi_3(z) &= \frac{1}{(1+z^2)^5}, & \phi_4(z) &= \operatorname{sech}(z),\\
\phi_5(z) &= z\operatorname{cosech}(z), & \phi_6(z) &= \exp\left(-z^2\right)\cos(\sqrt{2}z),\\
\phi_7(z) &= \frac{1}{\pi}\sin(\pi z)/\sinh(z), & \phi_8(z) &= \begin{cases}\exp\left(\frac{1}{z^2-R^2}+\frac{1}{R^2}\right), & z<R,\\ 0, & \text{else,}\end{cases}\\
\phi_9(z) &= \begin{cases}1, & z<R,\\ 0, & \text{else,}\end{cases} & \phi_{10}(z) &= \begin{cases}(R-z)/R, & z<R,\\ 0, & \text{else,}\end{cases}
\end{aligned} \tag{4.6}$$

and

$$\begin{aligned}
K_1(z) &= \sum_{k=0}^{m-1} a_k z^k, & K_2(z) &= \sum_{k=0}^{m-1} a_k z^{2k},\\
K_3(z) &= \sum_{k=0}^{m-1} a_k \cos(kz), & K_4(z) &= a_0 + \sum_{k=1}^{m-1} a_k \sin(kz).
\end{aligned} \tag{4.7}$$

Via the coefficients a_k, the correction functions K_i have m degrees of freedom. The number m corresponds to the number of moment conditions (1.4) or (2.4), respectively, that are not fulfilled automatically by symmetry. The coefficients a_k are fixed such that the respective moment conditions are satisfied. For UC operators, we only consider the correction function K_2.

Since the number of moment conditions depends on the desired order of accuracy r, the number m of coefficients a_k in a correction function K_i also depends on r. The choice of the kernel functions thus incorporates the choice of the order of accuracy r.

We remark that for second-order operators ($r = 2$), $m = 1$. In this case, the correction functions K_i, $i = 1, \ldots, 4$, are identical. The selection of window functions ϕ_j is motivated by the kernel functions used or proposed by Cottet and Koumoutsakos [22] ($j = 1, \ldots, 3$) and Maz'ya and Schmidt [63] ($j = 4, \ldots, 7$). The window functions $\phi_8, \ldots, \phi_{10}$ are included for their simplicity and compact support. We always use $R = 3$. The ten window

functions are plotted in figure 4.1.

The Cutoff Radius r_c The cutoff radius r_c defines the size of the closed ball $B_{r_c}[\boldsymbol{x}] = \{\boldsymbol{y} \in \mathbb{R}^n : |\boldsymbol{y} - \boldsymbol{x}| \leq r_c\}$ that defines the neighborhood of a particle at position $\boldsymbol{x}$. Reducing the cutoff radius reduces the cost of operator evaluation as fewer particles are considered neighbors and the sum over neighboring particles in equation (1.3) is shorter. For UC operators, however, reducing the cutoff radius increases the discretization error, because the approximated integral is truncated. A large cutoff radius is expected to increase operator stability and decrease its sensitivity to perturbations in the particle positions. The minimum cutoff radii are $r_{c,\min} = h$ for order $r = 2$ and $r_{c,\min} = 2h$ for order $r = 4$. A smaller cutoff radius cannot be used since the number of particles contained in the ball $B_{r_c}[\boldsymbol{x}]$ would be too small to yield an r^{th}-order approximation. We test a set $R_c = \{r_{c,1} = r_{c,\min}, r_{c,2}, \ldots, r_{c,k} = r_{c,\max}\}$, $r_{c,i} < r_{c,i+1}$ of cutoff radii, where the differences $r_{c,i+1} - r_{c,i}$ are minimized under the constraint that the number of particles in $B_{r_{c,i}}[\boldsymbol{x}]$ is different from the number of particles in $B_{r_{c,i+1}}[\boldsymbol{x}]$. The maximum cutoff radii $r_{c,\max}$ considered here depend on the order of accuracy r and on the window function ϕ_j, as listed in table 4.1.

	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}
$r = 2$	5ε	10ε	10ε	10ε	10ε	10ε	5ε	3ε	3ε	3ε
$r = 4$	7ε	12ε	12ε	12ε	12ε	12ε	7ε	3ε	3ε	3ε

Table 4.1: The maximum cutoff radii $r_{c,\max}$ considered for the different window functions in the present parameter study.

The Ratio $c = h/\varepsilon$ A small ratio $c = h/\varepsilon$ reduces the difference between the discrete moments (2.3) and the continuous moments (1.4). It therefore typically improves the accuracy of an UC operator. The cost of operator evaluation, however, is large for small values of c since more particles are located in the closed ball $B_{r_c}[\boldsymbol{x}]$ defining the neighborhood of a particle at position $\boldsymbol{x}$. The accuracy of a DC operator is expected to be higher for larger c: the influence of the direct neighbors relative to the influence of more distant particles is typically higher because the window function

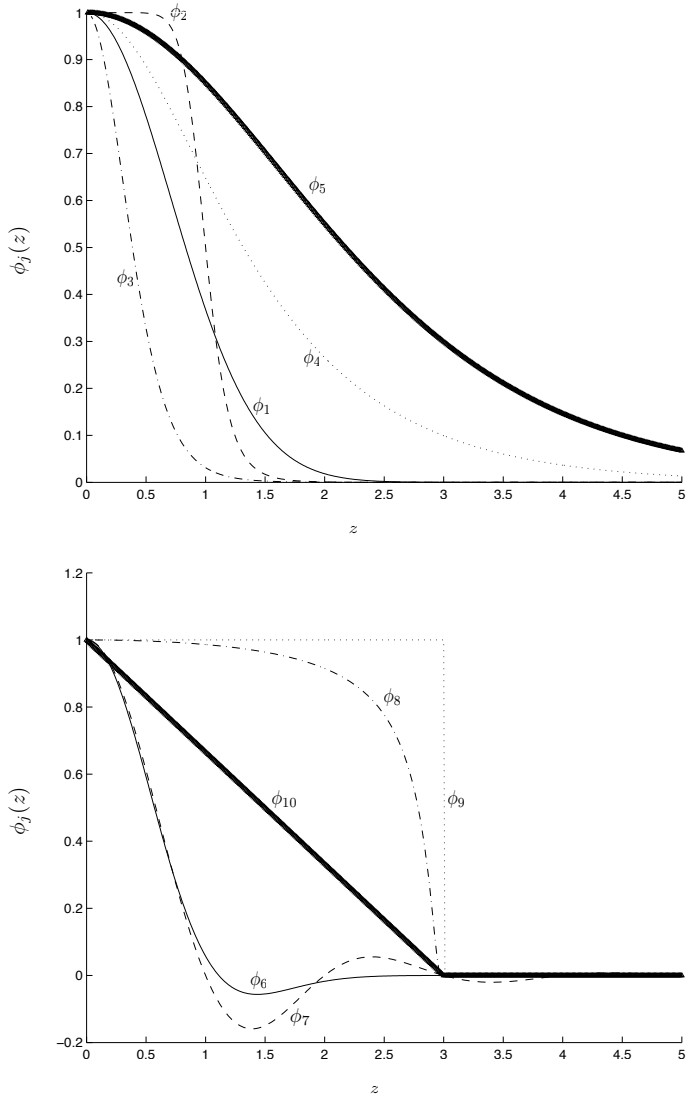

Figure 4.1: The ten window functions of equations (4.6).

is more peaked. Less smoothing occurs. The values of c tested here are $\{0.25, 0.3, 0.35, \ldots, 1.4\}$.

4.3.1.2 Results

In our parameter study, we evaluated the presented quality measures for both DC and UC PSE diffusion operators on uniform Cartesian particle distributions. We visited 443 956 points in parameter space, on the grid described in the previous section. We summarize the results schematically and point out potential conflicts of interest in the parameter choice.
The cost measure C is by definition independent of K_i and ϕ_j and hence depends only on the cutoff radius r_c relative to the interparticle spacing h. A sketch of this dependence is shown in figure 4.2. Since it does not

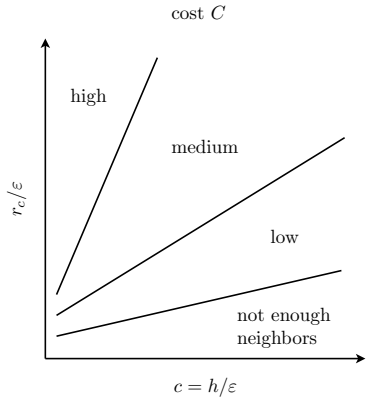

Figure 4.2: Cost measure C depending on the cutoff radius r_c and the ratio c.

depend on K_i, this sketch is identical for both UC and DC operators. In the region where $r_c/\varepsilon < c$ or $r_c/\varepsilon < 2c$ for $r = 2$ or $r = 4$, respectively, there are not enough particles in the closed ball $B_{r_c}[\boldsymbol{x}]$. The number of neighbors around $\boldsymbol{x}$ is then too small and there is not enough information to approximate $Lf(\boldsymbol{x})$ with the desired order of accuracy. The minimum possible cost is $C = V_3$ for $r = 2$ and $C = 8V_3$ for $r = 4$ on the lines $r_c/\varepsilon = c$ and $r_c/\varepsilon = 2c$, respectively.

For the other quality measures, typical landscapes over r_c/ε and c for UC and for DC operators are sketched in figures 4.3 and 4.4. In practice, UC operators can only be used in the parameter ranges for r_c and c where the accuracy $A_s, s < r$, is medium to high, because convergence is otherwise not observed. This can be understood from the error bound (4.2). The threshold for $A_s, s < r$, depends on the accuracy requirements. Given this restriction on the choice of r_c and c, the user can still choose an operator with a reasonable time-step limit (stability). However, the cost of operator evaluation is high and the accuracy A_r low.
For DC operators, the accuracy measures $A_s, s < r$, are by definition infinitely large whenever enough neighbor particles are present. In practice, where numerical errors are introduced when constructing the operators, they are finite but still very high compared to A_r, such that they usually do not influence the choice of r_c and c. If accuracy is the only criterion for operator design, the choice of r_c and c is simple: for decreasing $r_c/(c\varepsilon)$, the accuracy A_r typically increases and the cost C decreases such that r_c and c should be chosen on the line $r_c/\varepsilon = c$ or $r_c/\varepsilon = 2c$ for $r = 2$ or $r = 4$, respectively. For second-order accurate operators, the choice of c is arbitrary, because all operators on $r_c/\varepsilon = c$ (for any window function ϕ_j) reduce to the classical FD stencil

$$Q_h f(\boldsymbol{x}) \;=\; \frac{\nu}{h^2} \sum_{i=1}^{n} [f(\boldsymbol{x} + \boldsymbol{h}_i) - 2f(\boldsymbol{x}) + f(\boldsymbol{x} - \boldsymbol{h}_i)]$$

if ν is constant, or

$$\begin{aligned} Q_h f(\boldsymbol{x}) = \frac{1}{2h^2} \sum_{i=1}^{n} \{ & [f(\boldsymbol{x} + \boldsymbol{h}_i) - f(\boldsymbol{x})][\nu(\boldsymbol{x}) + \nu(\boldsymbol{x} + \boldsymbol{h}_i)] \\ & + [f(\boldsymbol{x} - \boldsymbol{h}_i) - f(\boldsymbol{x})][\nu(\boldsymbol{x}) + \nu(\boldsymbol{x} - \boldsymbol{h}_i)] \} \end{aligned} \tag{4.8}$$

else, where $\boldsymbol{h}_i$ is a vector of length h pointing along dimension i. The fourth-order accurate operators on the line $r_c/\varepsilon = 2c$ generally vary with c and are different for different choices of the window function ϕ_i and the correction function K_j. Therefore, the choices of ϕ_j, K_i, and c influence A_r. If accuracy is not the only important criterion, the choices of ϕ_j, K_i, r_c, and c are more difficult. As can be seen from figure 4.4, highly accurate operators typically require small time-step limits for stable explicit time

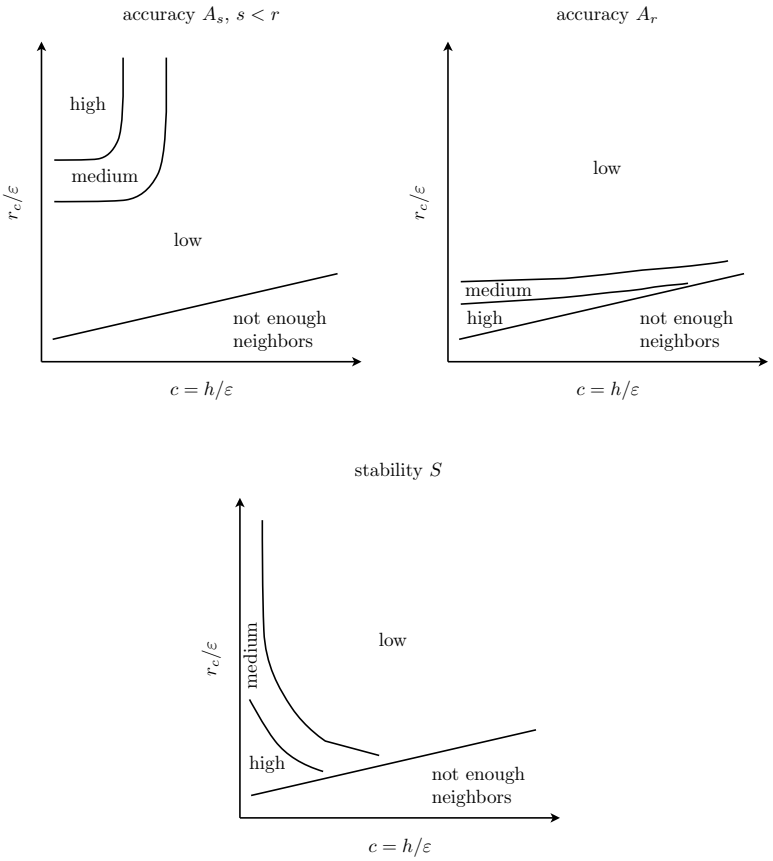

Figure 4.3: Typical quality measure landscapes for UC diffusion operators.

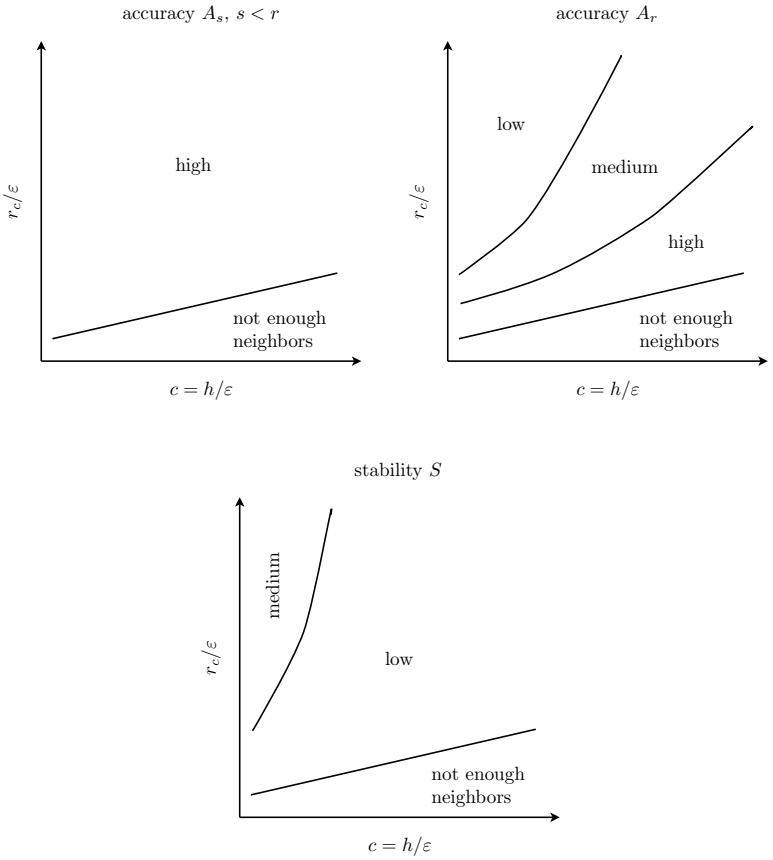

Figure 4.4: Typical quality measure landscapes for DC diffusion operators.

stepping. On the other hand, a larger stability measure S requires a more costly operator of comparably low accuracy. A trade-off must hence be found.

4.3.2 Real-World Applicability and Interplay of the Quality Measures: a Case Study

In practice, the most interesting property of an operator is probably the computational time needed in order to solve a given problem with a certain maximum error. Unfortunately, this property may depend on all three quality measures introduced above and their influence depends on the problem to be solved.
In this section, we show the utility of the introduced quality measures for choosing an operator and comparing its performance to the performance of operators that are typically used in the cases considered. In order to do so, we develop a rough estimate of the operator performance based on the quality measures such that their interplay and individual importance become apparent. A suitable operator is then chosen from all operators analyzed in the parameter study presented in §4.3.1.
As a test case we use the three-dimensional advection-diffusion problem described in appendix A.2. First, we consider simulations where the particles are remeshed at every time step, such that the particle distribution is always uniform Cartesian when the diffusion operator is applied. Second, we consider simulations where the intervals between the remeshing processes are large.

4.3.2.1 Advection-diffusion simulation with remeshing at every time step

We consider an advection-diffusion problem that we solve with remeshing at every time step. For remeshing, we use the third-order accurate M_4'-kernel given in equation (A.2). From a few simple timing experiments shown in figure 4.5, we estimate that the CPU time needed for one time step of the simulation is proportional to

$$C_{\mathrm{UC}} = C_{\mathrm{DC}} \sim 57 + C$$

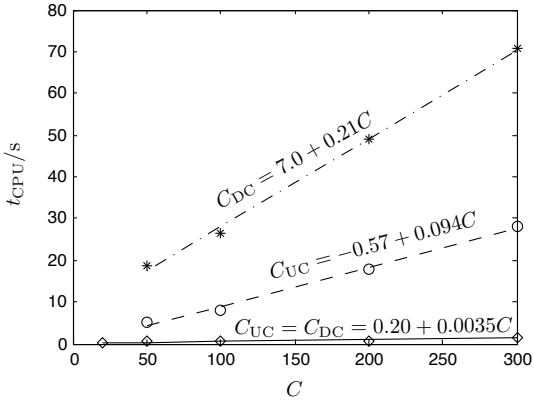

Figure 4.5: Timing experiments. Relationship between the cost measure C and the CPU time t_{CPU} for $h = 0.217$ and $\Delta t = 0.0125$ when remeshing at every time step (UC/DC: $\diamond$) or remeshing at large time intervals (UC: $\circ$, DC: $*$). We approximate $t_{\mathrm{CPU}}/\mathrm{s}$ with the functions $C_{\mathrm{UC}}(C)$ and $C_{\mathrm{DC}}(C)$, respectively, using linear regression.

where C is the cost measure of the diffusion operator used. Assuming that we use the largest time step possible (for efficiency), the CPU time needed for the complete simulation is proportional to $C_{\mathrm{UC/DC}}/S$. Thus, an efficient operator for this simulation should have a low ratio

$$F = \frac{57 + C}{S}$$

under the constraint that it has a zeroth-order accuracy A_0 of more than 10^7 and a positive stability measure S.[1] We do not account for the accuracy measures A_s, $s > 0$, because we expect that the remeshing error dominates the error introduced by the diffusion operator.
The operator in our parameter study that meets the constraints with the lowest ratio F is the DC PSE operator with $r = 2$, ϕ_7, $c = 0.6$, and $r_c/\varepsilon = 1.47$. We will call it the *appointed* operator in the following. We compare its performance to six other operators with typical parameter configurations. Two of these operators are the FD stencils of equation (4.8) and

$$\begin{aligned} Q_h f(\boldsymbol{x}) \quad = \quad & \frac{1}{24h^2} \sum_{i=1}^{n} \{ [\nu(\boldsymbol{x} - 2\boldsymbol{h}_i) + \nu(\boldsymbol{x})][f(\boldsymbol{x}) - f(\boldsymbol{x} - 2\boldsymbol{h}_i)] \\ & +16[\nu(\boldsymbol{x} - \boldsymbol{h}_i) + \nu(\boldsymbol{x})][f(\boldsymbol{x} - \boldsymbol{h}_i) - f(\boldsymbol{x})] \\ & +16[\nu(\boldsymbol{x} + \boldsymbol{h}_i) + \nu(\boldsymbol{x})][f(\boldsymbol{x} + \boldsymbol{h}_i) - f(\boldsymbol{x})] \\ & +[\nu(\boldsymbol{x} + 2\boldsymbol{h}_i) + \nu(\boldsymbol{x})][f(\boldsymbol{x}) - f(\boldsymbol{x} + 2\boldsymbol{h}_i)] \} \end{aligned} \tag{4.9}$$

of orders $r = 2$ and $r = 4$, respectively. The other four operators are second- and fourth-order DC and UC PSE operators with large[2] cutoff radii and medium c. For each operator, we adjust h and Δt semi-automatically such that the CPU time is low and the maximum error does

[1] The value of 10^7 is chosen arbitrarily and any value in $[0.6, 10^{15}]$ leads to the same result. This can be explained by the typical quality measure landscapes depicted in figures 4.2, 4.3, and 4.4: medium values of A_0 are associated with large costs C and low or medium values of S, such that operators with $A_0 \in [0.6, 10^{15}]$ correspond to large F.
The stability measure S is required to be positive such that the time integration is stable.

[2] The cutoff radii of $r_c = 3.5$ and $r_c = 5.5$ for the second- and fourth-order accurate operators, respectively, are large compared to those of the FD stencils and the appointed operator. They correspond to values recommended as best practice for UC PSE operators.

not exceed a certain target error level $E^*_{\max}$ at simulation time $T = 0.25$. We remark that all simulations take advantage of the consistently uniform Cartesian distribution of particles: particles are identified by mesh indices and no neighbor lists are used. The results are given in table 4.2. As expected, the appointed operator performs best. In fact, the simulation completes more than twice as fast as the fastest simulation using any of the other operators. Note that the DC and UC operators of the same parameters lead to the same results here. This is because r_c/ε is large enough and c is small enough for the UC operators to be able to reach the target error level.
The fourth-order accurate operators perform worst. This can be explained by the fact that the remeshing procedure is of lower order and the advantage of the high-order accuracy is not exploited in these simulations. Yet, the operator evaluation is more costly due to the larger cutoff radii required.
For the target error level $E^*_{\max} = 0.05$, the second-order FD operator outperforms the two PSE operators listed first. This is not surprising since its cost C is much lower (4.19 vs. 246.).[3] The cost C of the appointed operator, however, is higher than the cost of the FD operator (61.6 vs. 4.19). The stability S of the appointed operator is much higher, though, such that a larger time step can be chosen. This is enough to compensate for the higher cost of the operator evaluation.
For the target error level $E^*_{\max} = 0.01$, the second-order FD operator is outperformed by all listed second-order PSE operators. The reason is the low time stepping stability of the FD operator, which is more apparent at higher resolutions. Again, the high stability of the appointed operator (combined with its lower cost) renders it more efficient than the other operators.

4.3.2.2 Advection-diffusion simulation with remeshing at large intervals

If remeshing is performed at large time intervals only, we expect the diffusion error introduced at each time step to be dominant. We therefore modify the objective function F to represent the influence of the oper-

[3] Also, from the parameter study presented in §4.3.1 we see that its accuracy is the highest of all operators tested. We do, however, consider the effect to be marginal due to the dominance of the remeshing error.

Table 4.2: Advection-diffusion simulations with remeshing at every time step. We report the shortest[a] CPU times for solving the test case to a target error level $E^*_{\max}$. We show the results for six typical diffusion operators and for the appointed operator (last entry).

$E^*_{\max} = 0.05$

	r	c	r_c/ε	h	Δt	t_{CPU} in s
UC, ϕ_1	2	0.9	3.5	0.331	0.0625	0.0421
DC, ϕ_1	2	0.9	3.5	0.331	0.0625	0.0442
2^{nd}-order FD (4.8)				0.349	0.0417	0.0223
UC, ϕ_1, K_2	4	0.9	5.5	0.370	0.0417	0.207
DC, ϕ_1, K_2	4	0.9	5.5	0.370	0.0417	0.208
4^{th}-order FD (4.9)				0.349	0.0278	0.0555
DC, ϕ_7	2	0.6	1.47	0.370	0.125	0.00950

$E^*_{\max} = 0.01$

	r	c	r_c/ε	h	Δt	t_{CPU}/s
UC, ϕ_1	2	0.9	3.5	0.273	0.0208	0.257
DC, ϕ_1	2	0.9	3.5	0.273	0.0208	0.264
2^{nd}-order FD (4.8)				0.242	0.00417	0.637
UC, ϕ_1, K_2	4	0.9	5.5	0.185	0.00556	13.2
DC, ϕ_1, K_2	4	0.9	5.5	0.185	0.00556	13.0
4^{th}-order FD (4.9)				0.185	0.00287	3.55
DC, ϕ_7	2	0.6	1.47	0.251	0.0417	0.113

[a] Considering the CPU time as a function of h and Δt, we report the minimum CPU time and the corresponding values of h and Δt under the constraint of staying below the target error level.

ator accuracy A_r on the CPU time required to solve the given problem to the target error level $E^*_{\max}$: we assume a maximum simulation error $E_{\max} \sim h^r/A_r$. In this case, an interparticle spacing h proportional to $A_r^{1/r}$ is required. Since the number of particles is proportional to h^{-n}, $n = 3$, and the number of time steps to h^{-2}, the factor $A_r^{-5/r}$ is introduced into F. Thus, including the results of the timing experiments shown in figure 4.5,

$$F = \frac{C - 6.1 + \mathbb{I}_{\mathrm{DC}}(81 + 1.2C)}{SA_r^{5/r}},$$

where $\mathbb{I}_{\mathrm{DC}} = 1$ for DC operators and $\mathbb{I}_{\mathrm{DC}} = 0$ for UC ones. Note that the influence of the discretization correction on the simulation cost is significant when remeshing is done at large intervals only since the kernels have to be recomputed at every time step due to particle movement.
We use the same constraints on A_s, $s < r$, and S as in §4.3.2.1. Additionally, we impose a lower bound on the cutoff radii such that all operator neighborhoods contain enough particles at all times: $r_c/h > 2.14$ for $r = 2$ and $r_c/h > 2.52$ for $r = 4$. The cutoff radius $r_c = h$ or $r_c = 2h$, leading to compact FD stencils, is thus not included. From the parameter study presented in §4.3.1, the second-order DC operator with window function ϕ_2, ratio $c = 1.4$, and cutoff radius $r_c/\varepsilon = 3.2$ is then expected to perform best. We remark, however, that the parameter study was based on uniform Cartesian particle distributions, we deal with irregular distributions since remeshing is not done at every time step. It is therefore interesting to see how the performance measures generalize to this case.
Table 4.3 lists the resolutions and CPU times required to solve the given problem using the different operators.We do not use fourth-order operators in this test case because the simulation costs are expected to be much higher than those for the second-order operators: the timing experiments shown in figure 4.5 are done using second-order accurate operators; for fourth-order operators the runtime t_{CPU} is already 696s for $C = 100$, due to the higher costs in setting up and solving the linear systems of equations. We thus show the results for two typical second-order accurate PSE operators and the appointed PSE operator.
The appointed operator outperforms the other two operators tested. Thus, the prediction of the operator efficiencies based on the measures for uniform Cartesian particle distributions is valuable also for irregularly distributed particles. Interestingly, the DC operator is less accurate than the UC

Table 4.3: Advection-diffusion simulation with remeshing at large intervals. We report the shortest[a] CPU times for solving the test case to a target error level $E^*_{\max}$. We show the results for two typical PSE diffusion operators and for the appointed operator (last entry).

$E^*_{\max} = 0.05$

	r	c	r_c/ε	h	Δt	t_{CPU} in s
UC, ϕ_1	2	0.9	3.5	0.628	0.125	0.104
DC, ϕ_1	2	0.9	3.5	0.571	0.125	0.340
DC, ϕ_2	2	1.4	3.2	1.05	0.125	0.0196

$E^*_{\max} = 0.01$

	r	c	r_c/ε	h	Δt	t_{CPU} in s
UC, ϕ_1	2	0.9	3.5	0.349	0.050	1.32
DC, ϕ_1	2	0.9	3.5	0.331	0.0625	3.44
DC, ϕ_2	2	1.4	3.2	0.628	0.0625	0.157

[a] Considering the CPU time as a function of h and Δt, we report the minimum CPU time and the corresponding values of h and Δt under the constraint of staying below the target error level.

operator with the same parameters; a slightly higher resolution is required to reach the target error level. The simulation runtime using these DC operators is almost three times as long due to the computational cost of the discretization correction. With the appointed operator, however, this extra cost is amortized by the gain in accuracy and the cost reduction due to the smaller neighborhood. In our example, the gain in speed is five to eightfold.
Note that the runtimes of the simulations with remeshing at large time intervals (table 4.3) are longer than those of the simulations where remeshing is done at every time step (table 4.2). For the simulations using the UC operators, the main reason for this is the different implementation: while Verlet lists [83] are used when particles are irregularly distributed, the particles' mesh-like arrangement is exploited for looking up neighbors in the remeshed case. When using the DC operators, the discretization correction further doubles the computational cost per particle and time step, see figure 4.5. Rare remeshing is, however, expected to be faster for larger Péclet numbers, see §2.4.2.2.

4.3.3 Conclusions

We proposed three problem-independent quality measures for diffusion operators in particle methods. The measures model the respective operator's accuracy, computational cost, and stability in explicit time stepping. A parameter study has shown how the free operator parameters that have to be chosen by the user influence the operator properties as quantified by the introduced measures. For many tasks there are trade-offs when choosing a "good" operator. The problem-dependent interplay of the operator properties is then important.
Using the example of a three-dimensional Lagrangian advection-diffusion simulation, we demonstrated how simple objective functions of the quality measures can help in choosing efficient operators. Both frequent and occasional remeshing was tested. We were able to demonstrate that an operator's computational efficiency might depend on all three measures. The specific dependence, however, depends on the problem to be solved: for problems that are limited in the time step size by advection rather than by diffusion, the stability measure of the diffusion operator loses its importance. Similarly, the accuracy measure of the diffusion operator might lose importance if frequent remeshing is performed because the remeshing

error dominates the error introduced by the operator. If remeshing is a comparably rare event, the operator's accuracy is not negligible. At the same time, however, the most accurate operators typically require small time-step limits and choosing a higher spatial resolution and an operator of lower accuracy might be beneficial.
The performance model introduced allowed to select simulation-specific operators that were two to eight times faster than the next best of the tested diffusion operators.

CHAPTER

FIVE

Adaptive Multiresolution Simulations using Self-Organizing Lagrangian Particles

5.1 Introduction

Lagrangian particle methods are inherently adaptive with respect to the flow map. Advection, however, may lead to the formation of holes or clusters in the distribution of particles, jeopardizing the accuracy of the method. This issue is commonly addressed by periodical remeshing onto a Cartesian mesh [47]. Lagrangian particle methods are usually straightforward to implement and to parallelize. As such, they have been successfully used for large-scale simulations on supercomputers [13] and an efficient middleware for their parallel implementation exists [4, 77]. Yet, for systems with large spatial inhomogeneities, they have to be combined with multiresolution refinement techniques in order to remain computationally efficient. These techniques invariably introduce extra layers of complexity that may detract from the elegant simplicity of Lagrangian particle methods and complicate their implementation on (parallel) computers. A number of multiresolution frameworks have been proposed for particle methods; we briefly summarize them below. An in-depth review in the context of flow simulations has been presented by Koumoutsakos [48]. Smooth particle methods with spatially varying kernel widths were first

introduced by Cottet and coworkers [25] for vortex methods. This has enabled simulations of incompressible flows with vortex particles whose sizes are locally adapted to the scales of the flow field. The physical space, where particle sizes are locally adapted, is mapped to a reference space with uniform resolution. All operators, including remeshing, are applied in the uniform reference space. Particles living in low-resolution areas are therefore not allowed travel too far into any high-resolution area between two remeshing steps. Also, the mapping function from the physical to the reference space needs to be explicitly known. This limits the method to problems with static geometries or resolution requirements that are known beforehand. Bergdorf and coworkers later introduced the Adaptive Global Mapping (AGM) [9] by numerically approximating the mapping function on the particles, and numerically evaluating the Jacobian of the mapping in order to map differential operators between the physical and the reference space. This concept is related to r-adaptive finite element methods or moving meshes, in which the computational elements are dynamically moved to areas where increased resolution is needed. The number of computational elements and their topology, however, are preserved. The problem is typically formulated as equidistribution of the error onto the computational elements by means of a *monitor function* [12]. The monitor function may also carry other information: in AGM, for example, it depends on the spatial scales in the solution. AGM evaluates this monitor function at each particle location. The global mapping from the physical solution space to the reference space is then computed by inverting a large matrix with the number of rows and columns equal to the total number of particles in the simulation. This matrix inversion, or the corresponding FFTs if done spectrally, limits the scalability of the method on distributed-memory computers. Moreover, it does not allow for creation or removal of particles during adaptation. Dynamic adaptation of the number of particles, however, can be valuable when flow structures (interfaces, shocks, singularities, changes of phase, or regions of high vorticity) significantly evolve over time.

Another class of adaptive refinement schemes for particle methods are multilevel methods, such as wavelet particle methods (WPM) [8], adaptive mesh refinement (AMR) as applied to particle methods [9], and adaptive tree codes [66]. Wavelet particle methods provide a very sparse representation of multiscale data and a Lagrangian adaptation mechanism [8]. Due to the global nature of the involved data structures and transforms,

however, they are hard to parallelize on distributed-memory computers. AMR-type particle methods employ hierarchies of partially (or completely) overlapping mesh patches onto which the particles are remeshed at every time step. This also requires additional data structures, whose distribution onto a parallel machine is not trivial. Adaptive tree codes have for example been used for node refinement in 2D using radial basis functions [66]. The global tree data structure, however, again hampers efficient implementation on distributed-memory computers.
We propose an adaptive multiresolution Lagrangian particle method that does not require hierarchical data structures or mapping functions. This allows easy and efficient implementation on distributed-memory computers. The method relies on moving the particles to areas where high resolution is needed and on dynamic insertion and removal of particles in under- and over-resolved regions. The movement of the particles is guided by the flow and by pseudo forces that lead to the desired resolution and guarantee a well-organized particle distribution. This amounts to a self-organizing configuration of particles representing the solution fields in an adaptive manner. Pseudo forces have previously been used in moving-mesh methods [2]. There, the mesh nodes interact with each other through pseudo forces that depend on a measure of the local truncation error. Dynamic insertion and removal of particles was previously considered in the context of hybrid particle-mesh methods for convection-reaction-diffusion problems [81].
Approximate equidistribution of the error by means of a monitor function defines the desired resolution everywhere. DC PSE operators as described in chapter 2 are used to compute field derivatives. We take advantage of their flexibiliy (see §2.7) by reusing the discretization correction for evaluating derivatives of different degrees in order to reduce the computational cost. The same discretization correction is also reused for consistent interpolation.

5.2 Method

We introduce the concept of the presented method in §5.2.1 and discuss its details in §5.2.2.

5.2.1 Method Concept

We start from the concept of Lagrangian particle methods for transport problems in the strong formulation and introduce self-organizing particles as described in §5.2.1.2. The resulting method is outlined in §5.2.1.3. For details, we refer to §5.2.2.

5.2.1.1 Lagrangian Particle Methods for Transport Problems – Strong Formulation

We focus on transport problems of the form

$$\frac{\partial f}{\partial t} + \nabla \cdot (\boldsymbol{u} f) = \mathcal{L}(f), \tag{5.1}$$

where $\mathcal{L}$ is an elliptic differential operator, $\boldsymbol{u}$ an advection velocity field (i.e., the flow map), and $f\colon \mathbb{R}^d \to \mathbb{R}$ a continuous scalar field representing the concentration of the transported quantity. We disregard particle volumes such that the function approximation, as in generalized finite-difference methods, is given by

$$f(\boldsymbol{x}) \approx f^h(\boldsymbol{x}) = \sum_{p=1}^{N} f_p \zeta_\varepsilon(\boldsymbol{x} - \boldsymbol{x_p}, \boldsymbol{x}). \tag{5.2}$$

The kernel functions ζ are assumed to have compact support of radius $r_c(\boldsymbol{x})$ and are rescaled to the characteristic width $\varepsilon(\boldsymbol{x})$. The kernel widths $\varepsilon_p = \varepsilon(\boldsymbol{x}_p)$ and cutoff radii $r_{c,p} = r_c(\boldsymbol{x}_p)$ are scaling parameters that define the local spatial resolution of the method in the neighborhood of particle p. Note that they can be different on each particle, depending on the local resolution required. However, ε_p and $r_{c,p}$ are of the same order of magnitude.
The evolution of the particles follows the system of ordinary differential equations

$$\frac{\mathrm{d}\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{x}_p, t) = \boldsymbol{u}_p(t), \tag{5.3a}$$

$$\frac{\mathrm{d}f_p}{\mathrm{d}t} = \mathcal{L}^h(f_p, t) - f_p \nabla^h \cdot \boldsymbol{u}, \tag{5.3b}$$

where $\mathcal{L}^h$ and ∇^h are discrete approximations of the differential operators $\mathcal{L}$ and ∇.
During a simulation, the initially well-ordered particle distribution is distorted by the advection of the flow. Therefore, particles are commonly remeshed [47], that is, they are periodically reinitialized onto a mesh.

5.2.1.2 Self-Organizing Particles

The *self-organization* of particles in the present method comprises (i) the *self-assembly* of the particles due to pairwise particle interactions as in atomic, molecular, and supramolecular systems [86] and (ii) the dynamic insertion and removal of particles.
Figure 5.1 shows a two-dimensional example of self-assembling particles interacting via the Lennard-Jones potential. In this example, the particles equilibrate to form the nodes of a triangular or isometric grid. Other types of equilibrium configurations are observed for other interaction potentials. Finding an interaction potential that leads to self-assembly in specific target configurations is an inverse problem tackled both experimentally and theoretically [30, 74]. The most prominent target structures that could be valuable for numerical particle methods are isometric grids (triangular lattices), hexagonal grids (honeycomb lattices), and Cartesian grids (square lattices). For the method presented here, we concentrate on triangular lattices, which can usually be robustly obtained from self-assembly. We are, however, interested in multiresolution configurations. A straightforward way to realize multiresolution structures is to let the pairwise interaction potential depend on the position of the particles involved. Where high resolutions are required, the preferred interparticle distance (that is, the position of the well in the example of the Lennard-Jones potential) is set to a small value. Where coarse resolutions are sufficient, the preferred interparticle distance is increased. The equilibrium configuration shown in figure 5.2, for example, is obtained by letting the position of the well of the Lennard-Jones potential vary with the mean y-position of the two interacting particles. The triangular arrangement of the particles is still apparent, but there are more lattice defects resulting from the varying resolution.
The concept of the Lagrangian particle method presented in this chapter is to first let the particles move with the flow and then "correct" their positions by letting them self-assemble to a multiresolution configuration

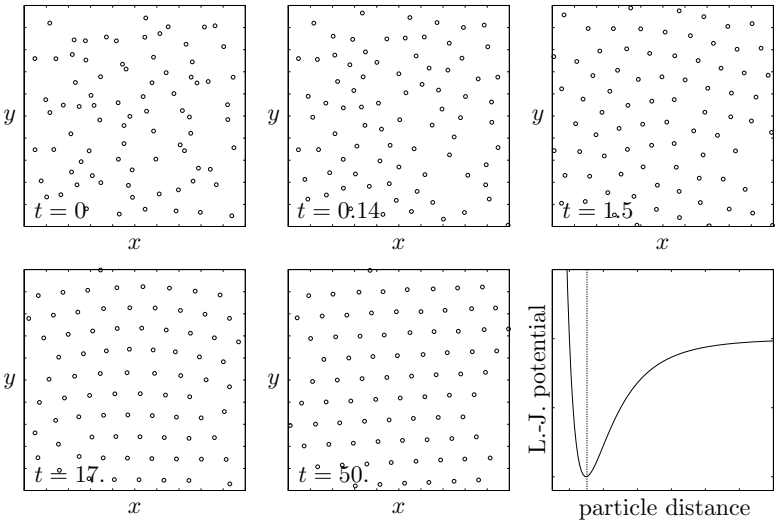

Figure 5.1: Self-assembly starting from an initially unstructured set of particles to a regular triangular lattice. We show a time series of self-assembling particles interacting via the pairwise Lennard-Jones potential sketched in the bottom right panel. The dotted line indicates the position of the potential well, the preferred distance between two interacting particles.

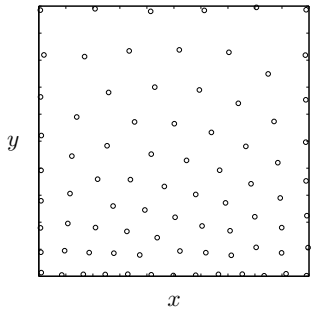

Figure 5.2: Self-assembled multiresolution configuration resulting from a position-dependent Lennard-Jones potential.

with resolutions defined by a monitor function. Typically, the monitor function is also carried by the flow, as it depends on the current solution of the problem. The corrections are therefore usually small.
In addition to the process of self-assembly, the self-organization of the particles includes dynamic insertion and removal of particles. This is beneficial when certain target error levels have to be met and structures like shocks or regions of high vorticity emerge or dissolve during a simulation. With constant numbers of particles, the user accepts the loss of small scales or additional computational cost. Another reason for insertion and removal of particles is the gain in computational efficiency in cases where the monitor functions varies rapidly over time. If, for example, the necessity of a very fine resolution vanishes in one region and manifests itself in another region, correction of the particle positions via self-assembly alone would need many iterations. The insertion/removal strategy offers an effective solution to this problem. A positive side-effect of the subsequent self-assembly is that the exact position of a newly inserted particle is of minor importance. In the present method, particles are inserted if not enough neighbors are found within the cutoff radii defined. Since the cutoff radii are linked to the preferred interparticle distance, a lack of neighbors usually corresponds to underresolution. A particle is removed if it gets too close to another one – they naturally merge if the interaction potential is chosen to be attractive for very short interparticle distances.
The solution carried by the particles then needs to be interpolated then from the particle positions before self-assembly to the positions after self-

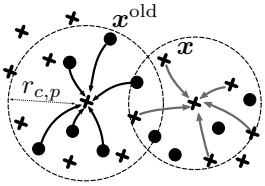

Figure 5.3: Interpolation of function values from the old set of particles (circles) to the current set of particles (crosses). After interpolation, the differential operators can be approximated either by using the values on the old particles (left circle, black arrows), or the values on the new particles (right circle, gray arrows). While the two ways are algebraically equivalent, they differ in computational cost when using DC PSE operators.

assembly. This corresponds to a remeshing procedure with an unstructured multiresolution target mesh. We use DC (PSE) operators both for particle-particle interpolation and for derivative approximation. We therefore do not need to evolve or determine the particle volumes because they are not required for consistent results. This also simplifies particle insertion and removal as the redistribution of the particle volumes does not need to be considered when particles vanish or appear.

5.2.1.3 A Self-Organizing Lagrangian Particle Method

We substitute the remeshing step by self-organization of the particles, as described in §5.2.1.2, followed by interpolation of the field values from the original to the reorganized particle set. We thus temporarily use two sets of particles, as depicted in figure 5.3. For the interpolation from the old to the new set of particles, we use DC interpolation kernels with the Kronecker delta property as described in §2.8. The particles self-organize at every time step. This ensures that the average nearest neighbor distance between particles of the original and the reorganized set is smaller than the average distance between particles and mesh nodes in the case of remeshing (see figure 5.4), thus reducing interpolation errors. Particles further self-organize into configurations that are nondegenerate with respect to the interpolation scheme used, ensuring that the field is well sampled everywhere.
After particle self-organization, we evolve the field values according to

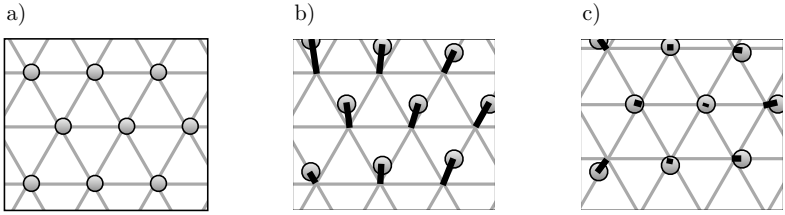

Figure 5.4: Remeshing onto a fixed mesh versus interpolation onto self-organized particles. a) Initial particle distribution. b) Particles have moved. The distances of the particles to the mesh nodes are displayed as thick black lines. c) Particles have moved. The distances between the old set of particles and the self-organized one (displayed as mesh nodes) are smaller than for case b). The interpolation is comparable to remeshing onto a moving mesh.

equation (5.3b). We use DC PSE operators for approximating the spatial derivatives. There are two ways the DC PSE operators can be used to compute the right-hand side of equation (5.3b) in a collocation setting:

(i) using the new set of particles as both source and collocation points (gray arrows in figure 5.3), or

(ii) using the old set of particles as source points, but the new set as collocation points (black arrows in figure 5.3).[1]

The two ways are algebraically equivalent for corresponding kernel choices. In fact, in both ways the two kernels (interpolation kernel and derivative approximation kernel) could be combined into one, as has been done for the case where $\mathcal{L}$ is the Laplacian and the particles are remeshed onto a uniform Cartesian mesh [85] . The two ways, however, differ with respect to their computational cost when using DC operators: variant (i) uses two kernels with different sets of source particles, hence requiring two different linear systems of equations to be solved on each particle. In variant (ii) both kernels use the same set of source particles and their weights can hence be determined from the same linear system of equations. Since solving the linear systems of equations at each particle accounts for the greatest part of the computational cost of a simulation the second variant is about twice as fast as the first variant. We therefore choose the variant (ii). We do, however, choose not to combine the two kernels into one for the sake of clarity of the presentation. Whenever possible we reuse the matrix inverse of the linear system of equations that needs to solved for the discretization correction, as described in §2.7.

5.2.2 Method Details

We consider the key elements of the present method in greater detail. We discuss the monitor function in §5.2.2.1, the choice of the cutoff radii in §5.2.2.2, the properties of the neighbor lists in §5.2.2.3, the particle-particle interaction potential in §5.2.2.4, the particle insertion and removal strategy in §5.2.2.5, the minimization of the potential energy in §5.2.2.6, the initialization procedure in §5.2.2.7, and boundary conditions in §5.2.2.8.

[1] In this case, the DC operator needs to be defined as in equation (2.29), that is, the zeroth moment must be set to zero.

5.2.2.1 Resolution Field (Monitor Function)

The pointwise error bounds for the approximation of the derivatives and the interpolation motivate a spatially adapted resolution. Ideally, the error would be equidistributed across all particles. This would then result in the minimum number of particles needed for the approximations to reach below a certain error level everywhere in the domain. The self-organization of the particles thus follows a target resolution that is determined by a monitor function that is based on properties of the current solution.
We denote by $\tilde{D}(\boldsymbol{x})$ the desired local target resolution of the spatial discretization. It defines the smallest scales that ought to be resolved by the numerical approximation in the neighborhood of $\boldsymbol{x}$.
In order to be able to determine the locally required resolution at every point in the computational domain, $\tilde{D}(\boldsymbol{x})$ needs to be expressed as a function of known or computable properties of f. The choice of the target resolution field "appropriate to the accurate solution of a PDE is difficult, problem-dependent, and the subject of much research" [12]. Among many possible choices, we choose the simple arclength-based[2] form

$$\tilde{D}(\boldsymbol{x}) = \frac{D_0}{\sqrt{1 + |\nabla f(\boldsymbol{x})|^2}}, \tag{5.4}$$

which is often used in multiresolution methods, including moving-mesh methods. D_0 is a user-defined parameter that sets the coarsest resolution in the computational domain and hence an upper bound on the interparticle spacing h. We refer to §2.5 for a discussion of how the resolution influences the accuracy of DC PSE operators.
Each particle p is assigned the minimum value of $\tilde{D}$ over its neighbors within a certain cutoff radius:

$$D_p = \min_{|\boldsymbol{x}_q - \boldsymbol{x}_p| \leq D_p r^*} \tilde{D}(\boldsymbol{x}_q). \tag{5.5}$$

This smoothes the resolution field and assures that the norms of the field f in the pointwise error bounds are comparable to the corresponding derivatives of the field f at that point. Without this smoothing procedure, an

[2] D_0 is then the uniform target spacing of the points $(\boldsymbol{x}_p, f_p)$ in $(n+1)$-dimensional space.

equidistribution of the error could not be obtained if the monitor function only included point information. The values of D_p can be determined as outlined in algorithm 1. The particle kernel widths then adapt to the new cutoff radii as $\varepsilon_p = r_{c,p} = D_p r^*$.

$D_p \leftarrow \tilde{D}(\boldsymbol{x}_p)$
Create (transient) neighbor lists.
foreach particle p **do**
  foreach particle $q \in \mathcal{N}_p$ **do**
    $D_q \leftarrow \min(D_q, \tilde{D}(\boldsymbol{x}_p))$

Algorithm 1: Computing D_p as defined in equation (5.5).

5.2.2.2 Choice of the Cutoff Radii

The particle-specific cutoff radii $r_{c,p}$ are determined by the target resolutions D_p and a global dimensionless cutoff radius r^* as $r_{c,p} = r^* D_p$. The parameter r^* is chosen such that $|\mathcal{N}_p| \approx \mathcal{N}^*$ if the particles were arranged on a regular triangular lattice with spacing D_p. Here, $\mathcal{N}_p$ is the index set of neighbors of particle p, $|\mathcal{N}_p|$ the actual number of neighbors of particle p, and $\mathcal{N}^*$ is the minimum number of neighbors required for a consistent discretization. For a two-dimensional triangular lattice, for example, $r^* = \{1, \sqrt{3}, 2, \sqrt{7}, 3\}$ leads to $|\mathcal{N}_p| = \{6, 12, 18, 30, 36\}$, respectively. In three dimensions, $r^* = \{1, \sqrt{2}, \sqrt{3}, 2\}$ leads to $|\mathcal{N}_p| = \{12, 18, 42, 54\}$. While this provides a good value for r^*, the particles generally do not form a regular triangular lattice and the particle insertion/removal strategy outlined in §5.2.2.5 becomes important.

5.2.2.3 Neighbor Lists

In the present method the cutoff radius r_c is a function of space. Defining the neighbors of particle p, denoted by the index set $\mathcal{N}_p$, as those particles within a ball of radius $r_{c,p}$ around $\boldsymbol{x}_p$ could hence lead to the situation where particle p is a neighbor of particle q, but not vice versa. This could lead to prohibitively high computational costs because particles in coarsely resolved regions (r_c large) would interact with potentially large clusters of

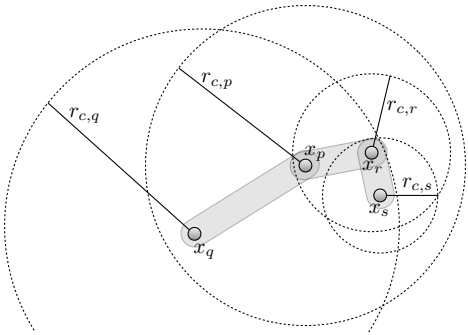

Figure 5.5: Illustration of the present definition of neighborhood. Particles that are neighbors are grouped together by shaded links.

particles in nearby finely resolved regions (r_c small). As illustrated in figure 5.5, we therefore consider as neighbors of particle p all particles q at a distance less than $\min(r_{c,p}, r_{c,q})$ from particle p. This ensures that $q \in \mathcal{N}_p \Leftrightarrow p \in \mathcal{N}_q$. Using such symmetric neighbor lists enables the method to handle arbitrarily large ranges of resolutions. Moreover, ghost layers in parallel implementations using symmetric neighbor lists are generally smaller leading to a better scalability.

5.2.2.4 Interaction Potential

Particle self-organize to the target resolution by pseudo forces resulting from pairwise interaction potentials between the particles. The interaction potential between particles p and q is of the form

$$V_{pq} = D_{pq}^2 V(|\boldsymbol{x}_p - \boldsymbol{x}_q| / D_{pq}), \tag{5.6}$$

where $D_{pq} = \min(D_p, D_q)$ and $V(r)$ is a user-defined pair potential. This form ensures that the adaptation pseudo forces scale with D_{pq} and that the length scale of the potential corresponds to the local resolution requirement. Two possible choices for $V(r)$ are shown in figure 5.6. The potential in figure 5.6a is attractive/repulsive and well suited for problems with free-space boundary conditions. It leads to a particle distribution with a characteristic local spacing of D_{pq}. The potential in figure 5.6b is

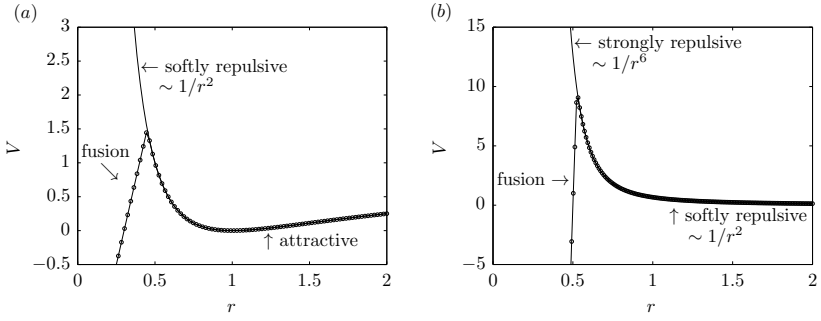

Figure 5.6: Examples of normalized pairwise interaction potentials. (a) $V(r) = (1 - 1/r)^2$, (b) $V(r) = r^{-2}/2 + r^{-6}/6$. The lines with symbols denote the modified potentials with strong short-range attraction to induce particle fusion in over-resolved regions.

purely repulsive, which is advantageous on finite domains. In this case, the potential ensures that the ratio of the distance between two particles and their kernel widths reflects the local resolution requirement. The absolute value of the interparticle distance is adjusted by changing the total number of particles in the computational domain. In practice, both potentials are modified to linearly decay below $D_{pq}/2$ (circles in figure 5.6), which ensures that particles that are too close to each other fuse, see also §5.2.2.5.
Locally minimizing the total potential energy

$$W(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \quad \equiv \quad \sum_p \sum_q V_{pq} \tag{5.7}$$

with respect to the particle positions $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ leads to a distribution of particles that provides a spatial resolution close to the target resolution $\tilde{D}(\boldsymbol{x})$, such that the characteristic interparticle spacing h_p near a particle at $\boldsymbol{x}_p$ is smaller than or equal to $\tilde{D}(\boldsymbol{x}_p)$. Finding the global minimum of the potential energy of a large collection of interacting particles is rarely feasible and always computationally expensive. Finding a particle distribution of "reasonably low" energy is comparatively much easier. For sufficiently smooth pairwise potentials, simple gradient descent algorithms are able to robustly find local minima of the potential energy, which turns out to be sufficient for the purpose of the present method. We find a "reaonsably low" W using the method of steepest descent as described in §5.2.2.6. Figure 5.7 shows a two-dimensional example of a resulting particle distribution.

5.2.2.5 Insertion/Removal of Particles

The number of iterations required by the gradient descent algorithm for minimizing the potential energy W closely depends on how far the initial condition is from the local minimum where the algorithm terminates. If the initial particle distribution is very different from the adapted one, the number of iterations required can be prohibitively large. This may happen, for example, when an initially uniform solution field develops steep gradients in a small area of the computational domain. In order to resolve these gradients, many particles distributed across the entire computational domain need to move to the region where refinement is

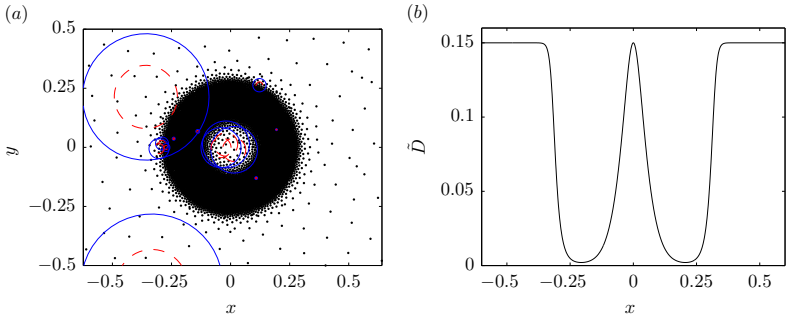

Figure 5.7: (a) Self-organized particles for the resolution field (5.4) and $f = \tanh\left(\left(x^2+y^2-0.4\right)/0.02\right)$; circles represent D_p (dashed) and $r_{c,p}$ (solid) for a sample set of particles. (b) Corresponding profile of $\tilde{D}$ along $y = 0$.

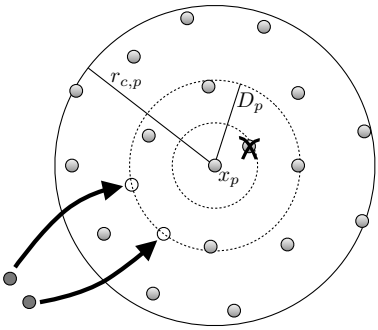

Figure 5.8: Insertion and removal of particles. Particles that are too close to others are removed and new particles are inserted at distance D_p if there are too few particles in the neighborhood.

needed. This major particle rearrangement would quickly become the bottleneck in large simulations. Moreover, if fine scales in the solution develop or disappear over time, not only the distribution of particles but also their total number has to adapt.

We achieve high adaptation efficiency and adaptive particle numbers by dynamically removing particles from overresolved regions and inserting new ones in underresolved regions. We do this by fusing particles that are too close to each other and by generating new particles in regions where the already existing ones have fewer neighbors than a critical number, see figure 5.8.

We want that near any particle p the neighboring particles locally adapt to a distribution with characteristic spacing D_p. The first requirement that needs to be fulfilled is that each particle must have a minimum number of $\mathcal{N}^*$ neighbors within its cutoff radius $r_{c,p}$ in order for the discretization to be consistent. This number is equal to the number of moment conditions that need to be fulfilled by the discretized operators and depends on the order of accuracy of the spatial discretization of the elliptic operator $\mathcal{L}$ in equation (5.1), see §2.2. For fourth-order interpolation and second-order approximation of the Laplacian, for example, $\mathcal{N}^* = 10$ in 2D and $\mathcal{N}^* = 20$ in 3D. The actual number of neighbors of any particle is ideally identical to $\mathcal{N}^*$ as any additional neighbors increase the computational cost. We thus take $\mathcal{N}^*$ as the critical number of neighbors for the particle

insertion/removal strategy.
As described in §5.2.2.2, the cutoff radius $r_{c,p}$ is fixed by the target spacing D_p and the parameter r^*. In order to assure that the actual number of neighbors is at least $\mathcal{N}^*$ and that the actual particle spacing is close to the target spacing, the local particle density is adapted during gradient descent by inserting particles in regions where $\mathcal{N}_p < \mathcal{N}^*$ and fusing particles that are closer to each other than $D_p/2$. The inserted particles are placed randomly at distance D_p from particle position $\boldsymbol{x}_p$. The total number of particles N in a simulation is hence not a free parameter of the method, but is determined adaptively by the algorithm at runtime.
We find that this insertion/removal strategy is effective in dealing with global changes in the required resolution, allowing the gradient descent minimizer to reach a local minimum within few iterations (typically less than ten).
If necessary, N can be bounded from above by imposing a minimum threshold value on the resolution field $\tilde{D}(\boldsymbol{x})$. This is necessary, for example, when the solution develops infinitely steep gradients. We therefore impose a minimum threshold on the resolution field:

$$\tilde{D}(\boldsymbol{x}) \leftarrow \max\left(\tilde{D}(\boldsymbol{x}), D_{\min}\right).$$

This guarantees that N does not exceed $N_{\max} \approx |\Omega| / D_{\min}^n$, where Ω is the n-dimensional computational domain. It also means, however, that the field f will be underresolved in regions where $\tilde{D} < D_{\min}$ and the desired accuracy level cannot be reached.

5.2.2.6 Steepest Descent on the Potential Energy

The self-organization of the particle positions and kernel widths is driven by pseudo forces such as to locally minimize the total potential energy W given in equation (5.7). After fusing particles that are too close to each other and inserting new particles where needed, a single step of a gradient descent is performed. This displaces each particle a step $\boldsymbol{w}_p$ in

the direction of the gradient of the interaction potential W:

$$\begin{aligned}\boldsymbol{w}_p &= -\alpha \partial W\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\right) / \partial \boldsymbol{x}_p, \\ &= -\alpha \left[\sum_{q \in \mathcal{N}_p} \partial V_{pq} / \partial \boldsymbol{x}_p + \sum_{q=1 \ldots N, \text{ s.t. } p \in \mathcal{N}_q} \partial V_{qp} / \partial \boldsymbol{x}_p \right],\end{aligned}$$

where the step size α is determined by a line search.[3] Since both the pairwise interaction potential V and the neighborhood relations are symmetric, see equation (5.6) and §5.2.2.3, the gradient descent flow simplifies to

$$\boldsymbol{w}_p = -2\alpha \sum_{q \in \mathcal{N}_p} \partial V_{pq} / \partial \boldsymbol{x}_p. \tag{5.8}$$

The chain rule leads to

$$\boldsymbol{w}_p = -2\alpha \sum_{q \in \mathcal{N}_p} D_{pq} \left[V'(r) \boldsymbol{e}_{pq} + \left(2V\left(r\right) - rV'(r)\right) \nabla_{\boldsymbol{x}_p} D_{pq} \right]_{r = \frac{|\boldsymbol{x}_p - \boldsymbol{x}_q|}{D_{pq}}}, \tag{5.9}$$

where $\boldsymbol{e}_{pq}$ is the unit vector pointing from particle p to particle q. Note that once the field f is properly resolved (that is, the total number of particles N is sufficiently large) then $\nabla_{\boldsymbol{x}_p} D_{pq} \ll 1$ and the second term in equation (5.9) can be neglected.

During particle self-organization, the insertion/removal and steepest descent algorithms are iterated until all particles have at least $\mathcal{N}^*$ neighbors and the stopping criterion $\min_p \min_{q \in \mathcal{N}_p} \left(|\boldsymbol{x}_p - \boldsymbol{x}_q| / D_{pq} \right) \geq d_c$ is met. This criterion ensures that no two particles are too close to each other. We find that a value of $d_c = 0.4$ leads to a small number of iterations (typically less than ten) while ensuring a suitable distribution of particles where the local density of particles matches the desired resolution field. Choosing a larger value for d_c leads to more regular particle distributions at the expense of a larger number of iterations.

[3] We use the quadratic fit line search method: for each step, we find a three-point bracket $0 = \alpha_1 < \alpha_2 < \alpha_3$ such that $W(\boldsymbol{X}_2) < \min[W(\boldsymbol{X}_1), W(\boldsymbol{X}_3)]$, $\boldsymbol{X}_i = \boldsymbol{X} - \alpha_i \nabla W(\boldsymbol{X})$, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{nN}$. We fit a (one-dimensional) quadratic polynomial to W using the values at $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$. We set α to the step size going to the minimum of the quadratic fit.

We set $\boldsymbol{w}_p = \mathbf{0}$ in the special case where $\tilde{D}(\boldsymbol{x}_p) > 2D_p$. This ensures that the self-organization remains stable and accurate even when the gradient of the resolution field $\tilde{D}$ is very large, in other words, when fine scales and coarse scales in the solution are not separated by a layer of intermediate scales. In this case, particles p that are in a coarse region where $\tilde{D}(\boldsymbol{x}_p)$ is large, but are neighbors of particles located in a fine region, have a small D_p, see equation (5.5). If they were allowed to move, the discrepancy between $\tilde{D}$ and D_p would cause them to be pushed away from the fine region and deteriorate the overall accuracy.

5.2.2.7 Initialization

There are several possibilities of how to place the particles at the beginning of a simulation: Particles can be initialized on a Cartesian mesh, placed uniformly random in the computational domain, or sampled from a probability density function that depends on the initial condition $f(\boldsymbol{x}, t = 0)$. In practice, we find that uniformly random placement of a fixed number of particles and subsequent adaptation to the initial condition is sufficient. This leads to a very simple and robust initialization strategy as the number of particles required to represent the initial condition to below a certain error bound is determined by the adaptation algorithm and does not need to be known beforehand. We illustrate this initialization procedure for the example of a Gaussian pulse (see §5.3.3 below). We initially randomly place a fixed number of 800 particles with uniform kernel widths as shown in figure 5.9. We then iterate the adaptation algorithm (loop in line 10 of algorithm 2 below) until it terminates. Figure 5.10 shows the two different resulting particle distributions for $D_0 = 0.2$ and $D_0 = 0.05$, respectively. The total resulting number of particles is 3027 in the first case and 27631 in the second. Note that this is found by the adaptation algorithm itself and does not need to be imposed by the user.
This resulting particle distribution is then used to represent the initial condition of the problem before entering time stepping.

5.2.2.8 Boundary Conditions

In the numerical experiments and benchmarks presented below, all boundaries are periodic. Other types of boundary conditions, however, can be treated in the standard ways.

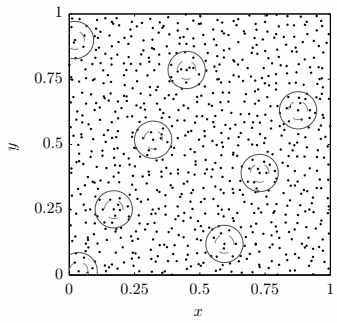

Figure 5.9: Initial set of 800 uniformly randomly placed particles. Circles represent D_p (dashed) and $r_{c,p}$ (solid) for a sample set of particles.

Homogeneous Neumann or Dirichlet boundary conditions can be imposed using mirror particles in a small neighborhood outside the computational domain (method of images). These mirror particles are only used to evaluate the right-hand side of equation (5.3b). They are not considered for the self-organization, nor for particle-particle interpolation. Instead, they are regenerated after self-organization of the particles is complete. During self-organization, the boundaries of the domain (if not periodic) are treated as rigid walls that confine the particles to the computational domain.
Inhomogeneous or mixed boundary conditions can be enforced by locally modifying the intensities of the particles in the neighborhood of the boundary [49].

5.3 Numerical experiments and benchmarks

We present an array of numerical experiments and benchmarks that are designed to demonstrate the capabilities and limitations of the present method. The first benchmark in §5.3.1 demonstrates the consistency of the operator approximation and particle-particle interpolation schemes introduced in §2.5, §2.7, and §2.8. The second benchmark considers a pure advection problem with a passive scalar. This is the same test case as was considered by Bergdorf and Koumoutsakos [8], which allows comparing the present method to the multiresolution wavelet particle method presented there. The third test case, in §5.3.3, adds diffusion and consid-

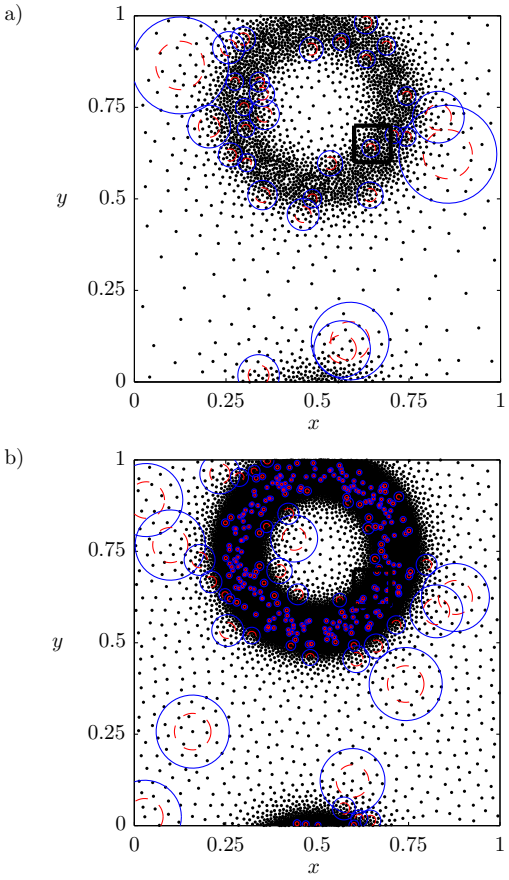

Figure 5.10: Particle distribution after adaptation to the initial condition (5.14). We use the repulsive interaction potential shown in figure 5.6b). a) $D_0 = 0.2$ ($N = 3027$ particles, 7 gradient-descent iterations); b) $D_0 = 0.05$ ($N = 27631$ particles, 21 gradient-descent iterations). Circles represent D_p (dashed) and $r_{c,p}$ (solid) for a sample set of particles.

ers an advection-diffusion problem with known analytical solution. It also compares the behavior of the present method in two and three dimensions. The fourth test case, in §5.3.4, is the two-dimensional unsteady Burgers equation, which serves as a benchmark for nonlinear transport problems. The fifth test case demonstrates the efficiency and accuracy of the present method for the real-world application of the nonlinear two-dimensional Buckeley-Leverett problem.
All benchmarks are done in finite domains with periodic boundaries using the purely repulsive self-organization potential shown in figure 5.6b. Any time-stepping scheme can be used to solve the system of ordinary differential equations (5.3) until final time $t = T$. The time-stepping schemes used for the numerical experiments and benchmarks presented below are indicated for each test case separately.
We implemented the method as summarized in algorithm 2 in Fortran 90 using the PPM library [4, 77].

5.3.1 Consistency of Interpolation and Derivative Approximation

We assess the convergence of the interpolation kernels and of the discretized Laplacian operator on the test function

$$f(x, y) = \tanh\left(\frac{x^2 + y^2 - 0.2^2}{0.01}\right) \tag{5.10}$$

in the domain $\Omega = [-1, 1]^2$. This test function has a steep sigmoidal transition of tunable slope. Successively reducing the global scale parameter D_0, we measure the errors as follows: For each value of $D_0^{(n)}$, particles are assumed to be adapted to the field f with this $D_0^{(n)}$ and the function values on the particles are set to the exact values:

$$f_p^{(n)} = f\left(\boldsymbol{x}_p^{(n)}\right).$$

The particles then self-organize to a finer resolution $D_0^{(n+1)} = 0.95 D_0^{(n)}$ as described in §5.2.1.2 and the new function values $f_p^{(n+1)}$ are interpolated from the old values $f_p^{(n)}$. The matrices that have to be inverted for each particle to compute the interpolation kernels are reused to compute an

input : field $f_0(\boldsymbol{x}) = f(\boldsymbol{x}, t = 0)$
output: field $f^h(\boldsymbol{x}, t = T)$

Initialize $\boldsymbol{x}_p$ and f_p.
while $t < T$ **do**
  Choose Δt based on the globally smallest value of the interparticle distance.
  Advect particles with velocity $\boldsymbol{u}$ between t and $t + \Delta t$.
  Create neighbor lists.
  Solve equations (2.27) for DC PSE operators (source points = collocation points).
  Compute the field derivatives for $\tilde{D}(\boldsymbol{x}_p)$.
  Compute $D(\boldsymbol{x}_p)$ using equation (5.5).
  $\boldsymbol{x}_p^{\text{old}} \leftarrow \boldsymbol{x}_p$, $f_p^{\text{old}} \leftarrow f_p$
  repeat
    Fuse particles where $|\boldsymbol{x}_q - \boldsymbol{x}_p| < D_{pq}/2$.
    Insert particles in the neighborhood of particles that have fewer than $\mathcal{N}^*$ neighbors.
    Update the neighbor lists (both within $\boldsymbol{x}_p^{\text{new}}$ and between $\boldsymbol{x}_p$ and $\boldsymbol{x}_p^{\text{old}}$).
    Compute $D(\boldsymbol{x}_p)$ by 1^{st}-order interpolation from $D(\boldsymbol{x}_p^{\text{old}})$.
    Adapt the cutoff radii $r_{c,p}$ and kernel widths ε_p.
    Compute the total energy of the adaptation potential and its gradient.
    Perform a line search for the gradient descent step size and move the particles by one step along the energy gradient. Do not move particles for which $\tilde{D}(\boldsymbol{x}_p) > 2D_p$.
  until stopping criterion of the gradient descent is met **and** every particle has more than $\mathcal{N}^*$ neighbors.
  Create cross-neighbor lists: find neighbors of the collocation points in the set of source points.
  Solve equations (2.27) for the interpolation kernels and the kernels for evaluating the RHS of equation (5.3b). Use $\{\boldsymbol{x}_p^{\text{old}}\}$ as source points and $\{\boldsymbol{x}_p\}$ as collocation points.
  Interpolate the field values f_p^{old} from $\boldsymbol{x}_p^{\text{old}}$ to $\boldsymbol{x}_p$, obtaining f_p.
  Evaluate the RHS of equation (5.3b) and update f_p.
  $t \leftarrow t + \Delta t$

Algorithm 2: Adaptive multiresolution method using self-organizing Lagrangian particles

approximation $(\nabla^2)^h$ of the Laplacian operator ∇^2 using the old particles $\{\boldsymbol{x}_p^{(n)}, f_p^{(n)}\}$ as source points, as described in §5.2.1.3. The pointwise errors are defined as the differences

$$E_p^{\mathbf{0}} = \left| f_p^{(n+1)} - f\left(\boldsymbol{x}_p^{(n+1)}\right)\right|, \quad E_p^{\nabla^2} = \left|(\nabla^2)^h f\left(\boldsymbol{x}_p\right) - \nabla^2 f\left(\boldsymbol{x}_p\right)\right|.$$

for the interpolation and the approximation of the Laplacian, respectively. All kernels satisfy the corresponding moment conditions up to (and including) the fourth moments, leading to fourth-order convergence of the interpolation functions and second-order convergence of the Laplacian operator, as verified in figure 5.11. As D_0 decreases from 0.4 to 0.006, the number of particles increases from 10^2 to 2×10^5, approximately. Due to the self-organizaton, each value $D_0^{(n)}$ corresponds to a different set of particles to represent the test function f. The ruggedness of the convergence plot in figure 5.11 for low resolutions can thus be interpreted as the sensitivity of the error norm to the underlying particle distribution. This is not specific to the present method. The same effect in the L_∞-norm of the error also occurs, for instance, in Cartesian finite-difference schemes when rotating the mesh.

5.3.2 Advection

In order to illustrate the adaptivity of the present method, we consider two-dimensional advection of a passive scalar f by a given velocity field $\boldsymbol{w}$:

$$\frac{\partial f}{\partial t} + \boldsymbol{w} \cdot \nabla f = 0.$$

For direct comparison with the Lagrangian wavelet-particle method of Bergdorf and Koumoutsakos [8], we consider the same test problem. This comprises the advection of an initial "blob"

$$f\left(x, y, 0\right) = \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} \frac{\sqrt{2}}{2} \operatorname{erf}\left(c_1 \left(c_2 - \sqrt{\left(x - x_0 + i\right)^2 + \left(y - y_0 + j\right)^2}\right) + 1\right), \tag{5.11}$$

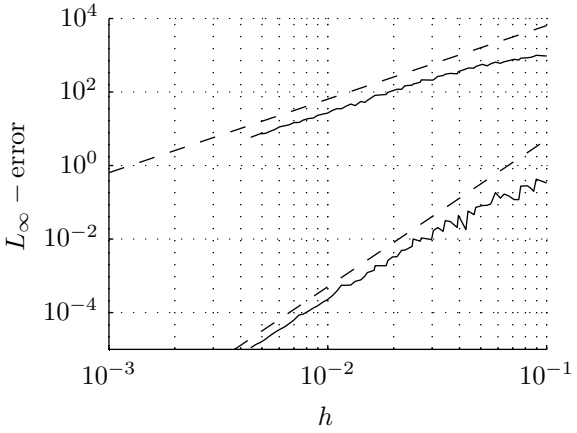

Figure 5.11: Maximum errors in function approximation, $\max_p E_p^{\mathbf{0}}$ (lower curve), and in the approximation of the Laplacian, $\max_p E_p^{\nabla^2}$ (upper curve), for the test function (5.10), plotted against the average interparticle spacing $h = \sqrt{|\Omega| / N}$. Dashed lines indicate second- and fourth-order convergence, respectively.

with $(x_0, y_0) = (0.5, 0, 75)$, $c_1 = 21.269446$, and $c_2 = 0.16811704$ by the divergence-free velocity field

$$\boldsymbol{w} = 2\cos(\pi t/T)\begin{pmatrix} -\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ \sin^2(\pi y)\sin(\pi x)\cos(\pi x) \end{pmatrix} \tag{5.12}$$

in the computational domain $\Omega = [0,1]^2$ with doubly periodic boundary conditions. We simulate the time evolution of the advected field f up to final time $T = 2.5$. The analytical solution at this final time is identical to the initial condition (5.11). The maximum distortion of the field f occurs at $t = T/2$ and is shown in figure 5.12.
As Bergdorf and Koumoutsakos [8], we use a fourth-order Runge-Kutta time-stepping scheme with a time step of $\Delta t = 0.025$. Interpolation is performed using fourth-order kernels and the error is defined as

$$E(\boldsymbol{x}) = \left(f^h(\boldsymbol{x}, T) - f(\boldsymbol{x}, T)\right) \|f(\boldsymbol{x}, T)\|_\infty^{-1} . \tag{5.13}$$

Figure 5.13 shows the L_∞ norm of $E(\boldsymbol{x})$ as a function of the the average interparticle spacing $h = 1/\sqrt{N}$. The convergence is fourth-order, as expected. Quantitatively, the errors are lower than or comparable to those published for the wavelet-particle method [8] for the same test case.

5.3.3 Solid-Body Rotation with Diffusion in 2D and 3D

As an advection-diffusion problem with known analytical solutions in both 2D and 3D, we consider the n-dimensional Gaussian pulse

$$f(\boldsymbol{x}, 0) = \exp\left(-\mathrm{Pe}\frac{|\boldsymbol{x} - \mathbf{x}_0|^2}{4}\right), \tag{5.14}$$

initially centered at $\mathbf{x}_0$, diffusing and being advected by solid-body rotation about the center of the computational domain $\Omega = [-L, L]^n$. This is described by the equation

$$\frac{\partial f}{\partial t} + \boldsymbol{w} \cdot \nabla f = \frac{1}{\mathrm{Pe}} \nabla^2 f, \tag{5.15}$$

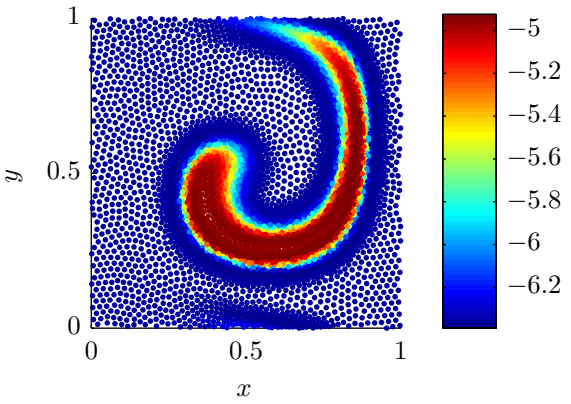

Figure 5.12: Passive advection of a scalar by the velocity field (5.12). The figure shows the particles at the time of maximum distortion, $t = T/2$. Color codes the function values f_p.

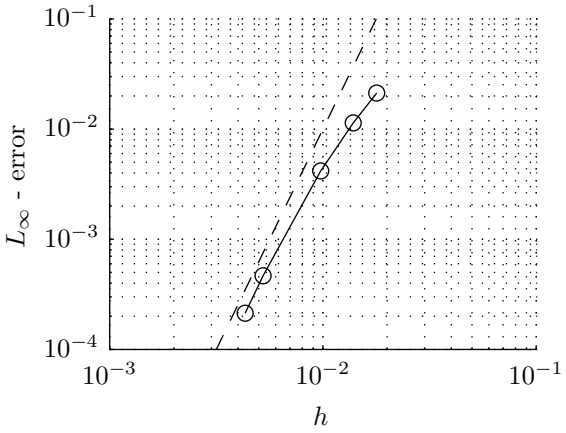

Figure 5.13: L_∞-error in the passive advection of the function (5.11) versus the average interparticle spacing $h = 1/\sqrt{N}$. The dashed line indicates fourth-order convergence.

where Pe is the dimensionless Péclet number and $\boldsymbol{w}$ is the velocity field of the solid-body rotation. After one revolution, at $t = 1$, the exact solution is given by

$$f(\boldsymbol{x}, 1) = 2^{-n/2} \exp\left(-\mathrm{Pe}\frac{|\boldsymbol{x} - \mathbf{x}_0|^2}{8}\right). \tag{5.16}$$

All kernels satisfy the corresponding moment conditions up to (and including) the fourth moments. The characteristic width of the initial pulse, $(2/\mathrm{Pe})^{1/2}$, and the halfwidth L of the computational domain are such that f is negligible near the boundaries of Ω at all times $t \leq 1$. The method converges with the expected second-order accuracy, the error being dominated by the diffusion term (the advection term is, in this case, computed exactly). The results are compared with those obtained using a remeshed Lagrangian particle method. There, advection is also computed exactly and diffusion is performed using second-order centered finite differences after particles have been interpolated onto a uniform Cartesian mesh of resolution h using the third-order M_4' interpolation kernel [22] first introduced by Monaghan [64]. Time-stepping is done in both cases using a forward Euler scheme with step size $\Delta t = h_{\min}^2/4$, where $h_{\min}$ is the smallest distance between any two particles. Trivially, $h_{\min} = h$ for the remeshed Lagrangian particle method.

We first consider the two-dimensional case with the domain $\Omega = [-1, 1]^2$, the center of the pulse $\mathbf{x}_0 = (0.58, 0.02)$, and the velocity field $\boldsymbol{w} = 2\pi(y, -x)$. An example particle distribution after adaptation to the initial condition (5.14) is shown in figure 5.14 for $\mathrm{Pe} = 10^4$. The steep gradients of the Gaussian pulse are well resolved by the self-organizing particles. Figure 5.15a shows the maximum pointwise error (5.13) at $t = 1$ for different resolutions and for $\mathrm{Pe} = 1000$. The method converges with the expected second-order accuracy of the diffusion operator.

We illustrate the behavior of the present method in three dimensions by considering the case with $\Omega = [-1.5, 1.5]^3$, $\mathbf{x}_0 = (0.58, 0.02, 0.02)$, and $\boldsymbol{w} = 2\pi\,(y, -x, 0)$. Also for three dimensions, the method converges with the expected accuracy, as shown in figure 5.15b for $\mathrm{Pe} = 100$. We do not show the maximum pointwise errors for the remeshed Lagrangian particle method because the computations performed in the range of h shown in the plot are not yet in the region of convergence. More particles would be necessary.

For solutions with large gradients, the finest scales that are resolved by

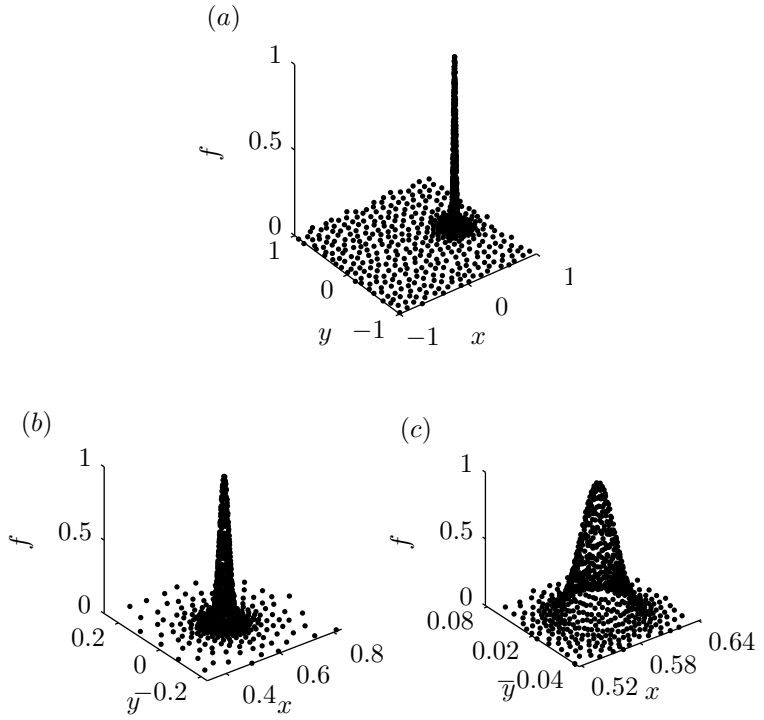

Figure 5.14: Example particle distribution $\{x_p, y_p, f(x_p, y_p, 0)\}$ after adaptation to the initial condition of the advected Gaussian pulse problem (Eq. (5.14)) for $\mathrm{Pe} = 10^4$. (a) entire computational domain Ω; (b)/(c) successive close-ups on the Gaussian pulse.

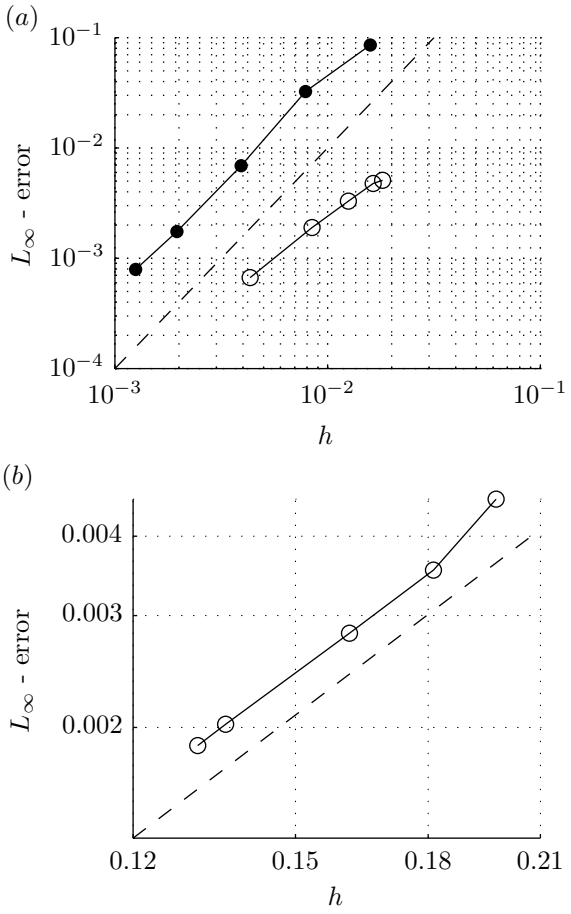

Figure 5.15: Maximum pointwise error versus average interparticle spacing h in the advection-diffusion problem (5.15) after one revolution of the Gaussian pulse at $t = 1$. (a) Two-dimensional case with Pe = 1000 and $h = 2/\sqrt{N}$; filled circles: remeshed Lagrangian particle method, open circles: present method. (b) Three-dimensional case with Pe = 100 and $h = 3/N^{1/3}$. Both dashed lines indicate second-order convergence.

the present method are of length $D_0/\max|\nabla f|$, which in this test case decreases as $\mathrm{Pe}^{-1/2}$. This length is approximately $0.23D_0$ for $\mathrm{Pe} = 100$. In the three-dimensional case we find that the corresponding number of particles required to achieve the same resolution on a uniform Cartesian mesh is about 50 times larger than when using the present multiresolution method.

5.3.4 2D Burgers equation

In order to demonstrate the performance of the present method on a non-linear problem, we consider the two-dimensional unsteady Burgers equation

$$\frac{\partial u}{\partial t} + \mathrm{Re}\,\boldsymbol{u}\cdot\nabla u = \nabla^2 u, \tag{5.17}$$

where Re is the Reynolds number and $\boldsymbol{u} = (u, u)$. We solve equation (5.17) subject to the initial condition $u(x, y, t = 0) = \sin(2\pi x)\cos(2\pi y)$ and doubly periodic boundary conditions in the computational domain $\Omega = [0, 1]^2$. For large Re, the Burgers equation behaves as a hyperbolic partial differential equation and the solution develops steep wave fronts over time, requiring an increasingly high resolution.
The solution as computed by the present method is shown in figure 5.16 at $t = 0.17/\mathrm{Re}$ for $\mathrm{Re} = 1000$ and $D_0 = 0.15$. The steep gradients in figure 5.16*a* correspond to the dense regions in figure 5.16*b*. The ratio of scales between fine and coarse regions is approximately 12 in this case and depends mostly on the gradient of the solution u and not on the user-defined resolution limit D_0. This indicates that all scales in the solution are properly resolved.
For comparison, we also solve the Burgers equation (5.17) using a remeshed Lagrangian particle method in the weak formulation where particles have a volume and carry an extensive strength. We use the fact that equation (5.17) can be rewritten in conservative form as a transport equation for the quantity u with a flow of velocity $\mathrm{Re}\,\boldsymbol{u}/2$:

$$\frac{\partial u}{\partial t} + \nabla\cdot\left(\frac{\mathrm{Re}\,\boldsymbol{u}}{2}u\right) = \nabla^2 u. \tag{5.18}$$

The particles are initialized on the nodes of a uniform Cartesian mesh

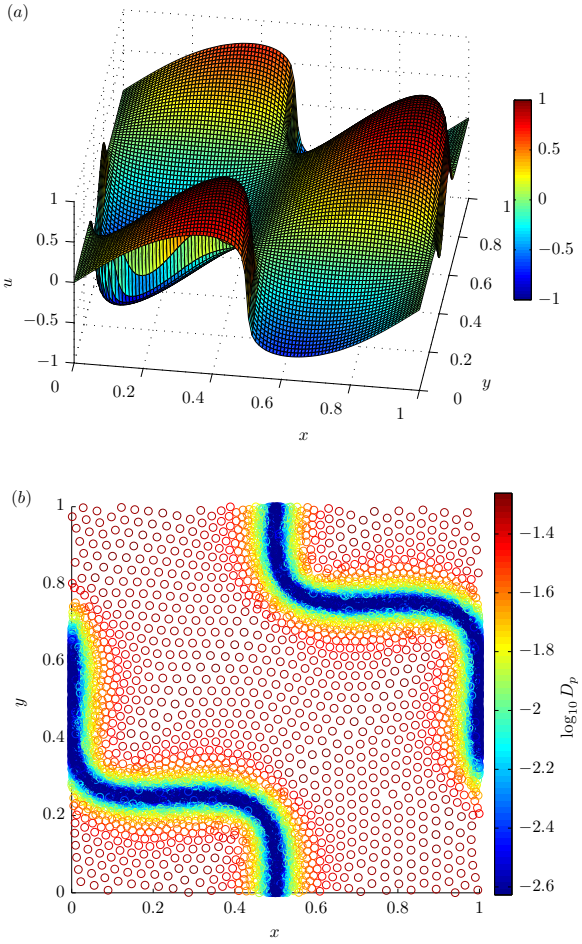

Figure 5.16: Numerical solution of the 2D Burgers equation for $\mathrm{Re} = 1000$ at $t = 0.17/\mathrm{Re}$ using the present method. (a) Values of $u(x, y)$ interpolated onto a Cartesian mesh for visualization purposes; color codes function value u. (b) Particle positions and sizes; color codes $\log_{10}(D_p)$.

covering Ω with a resolution of h. Advection with the velocity $\mathrm{Re}\,\boldsymbol{u}/2$ is performed using forward Euler with a time step of $\Delta t = h^2/4$. After each time step, the particles are remeshed using the M_4' interpolation kernel with width $\varepsilon = h$ [64]. The diffusion term is computed on the mesh using centered second-order finite differences.
Since no analytical solution is available for this problem, we use a numerical reference solution computed on a 2048×2048 mesh using the remeshed Lagrangian particle method. A second-order interpolation of this reference solution is used to compute pointwise errors at all particle locations. The maximum of these pointwise errors as a function of $h = 1/\sqrt{N}$ is shown in figure 5.17 at $t = 0.1/\mathrm{Re}$ for $\mathrm{Re} = 100$. Convergence with the average interparticle spacing h is second order in both cases. We use fourth-order interpolation kernels and second-order kernels for the approximation of the Laplacian. Time integration is done using forward Euler with a time step $\Delta t = h_{\min}^2/4$, where $h_{\min}$ is the smallest distance between any two particles in the domain. We do not show a convergence plot for the simulation at $\mathrm{Re} = 1000$, the solution shown in figure 5.16 because our numerical reference solution that we obtained with the remeshed Lagrangian particle method is not accurate enough.
The maximum number of particles used by the present method to achieve any given error level in this case ($\mathrm{Re} = 100$) is about eight times smaller than that of the remeshed Lagrangian particle method, independent of the target error level. This ratio, however, depends on the solution itself and increases as finer scales and steeper gradients develop.

5.3.5 The five-spot problem: 2D Buckeley-Leverett equation

As a real-world application we consider another two-dimensional nonlinear problem, known as the five-spot problem or the waterflooding problem. This popular test case for oil reservoir modeling describes the injection of a wetting fluid (water) at the center of a porous medium initially saturated with a nonwetting fluid (oil). The oil is flushed away by the pressurized water and sucked out from the four corners of the reservoir. Details of this test case are described by Iske and Käser [45]. When neglecting gravity and capillary effects, the problem reduces to the viscous Buckeley-Leverett

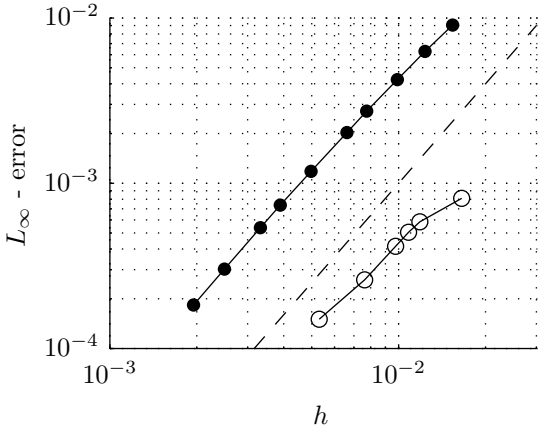

Figure 5.17: Maximum pointwise error versus average interparticle spacing $h = 1/\sqrt{N}$ of the 2D Burgers equation at $t = 0.1/\mathrm{Re}$ for $\mathrm{Re} = 100$; filled circles: remeshed Lagrangian particle method, open circles: present method. The dashed line indicates second-order convergence.

equation

$$\frac{\partial u}{\partial t} + \boldsymbol{v} \cdot \nabla f(u) = \nu \nabla^2 u \tag{5.19}$$

for the water saturation u. Here, the flux function f is of the form

$$f(u) = \frac{u^2}{u^2 + \mu (1-u)^2},$$

where $\mu > 0$ is the ratio of the two fluids' viscosities. The artificial diffusion term on the right-hand side of equation (5.19), with $\nu > 0$, is a standard regularization technique to render the equation parabolic and guarantee the existence of smooth solutions.
The oil is pumped out at the four corners of the domain $\Omega = [-0.5, 0.5]^2$ and water is injected at the origin $\boldsymbol{x} = \boldsymbol{0}$. Using the same simplifications as Iske and Käser [45], we assume that the velocity field is stationary and given by $\boldsymbol{v} = -\boldsymbol{\nabla} p$ with

$$p = \sum_{i=1}^{4} \log(\|\boldsymbol{x} - \boldsymbol{c}_i\|) - \log(\|\boldsymbol{x}\|),$$

where $\boldsymbol{c}_i$ are the positions of the four corners, as plotted in figure 5.18a. At $t = 0$, $u \equiv 1$ inside a disk of radius 0.02 centered at the injection well $\boldsymbol{x} = \boldsymbol{0}$. Equation (5.19) is then solved using the present method. Time integration is done using forward Euler with a step size of $\Delta t = h_{\min}^2/4$. Particle distributions and saturation fields at different times are shown in figure 5.19. The particles mostly adapt to concentrate near the steep water/oil front. The number of particles in the simulation grows from 373 to 7300 as the front elongates and propagates across the reservoir, see figure 5.18b.

5.4 Conclusions and Discussion

We presented an adaptive multiresolution Lagrangian particle method where the particles self-organize according to adaptation pseudo forces such as to smooth the numerical approximation error. This reduces the total number of particles required to represent the solution field with a given accuracy everywhere. In contrast to previous multiresolution parti-

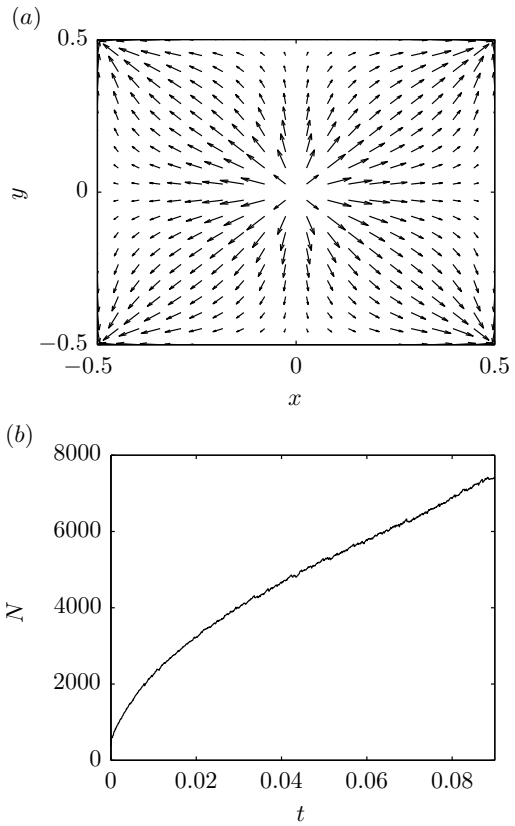

Figure 5.18: (a) Velocity field for the five-spot problem. The injection well is in the center of the domain and the fluid is pumped out from the four corners. (b) Adaptive evolution of the number of particles N during a simulation using the present method.

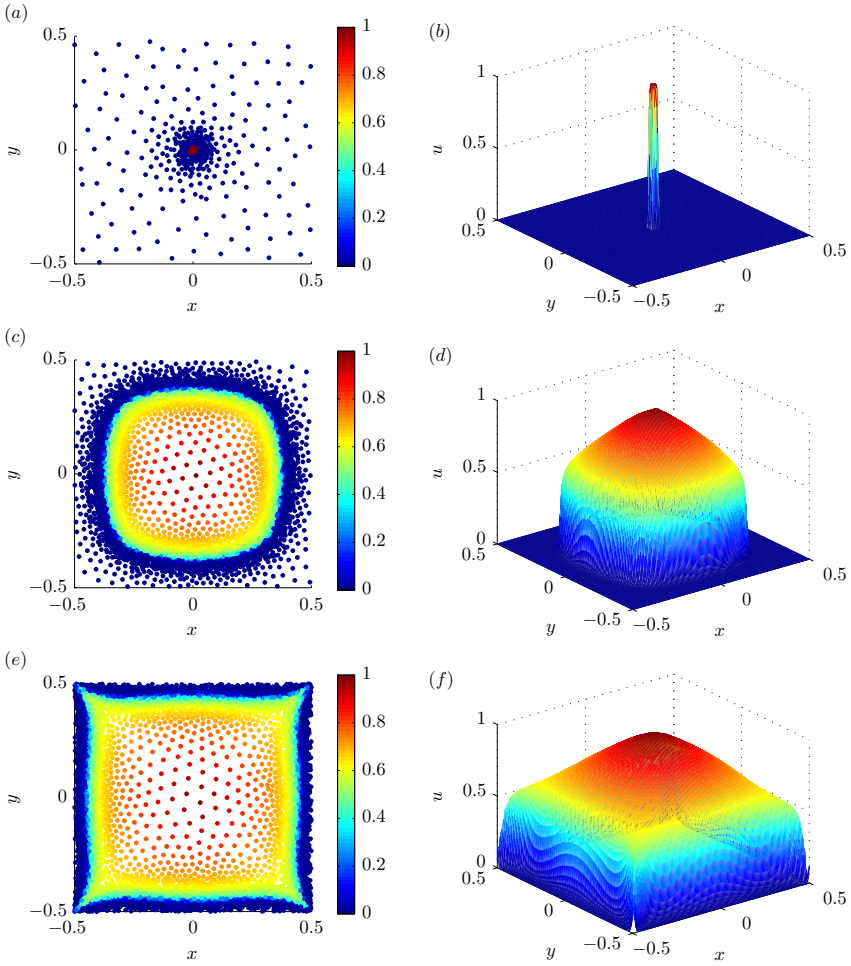

Figure 5.19: Visualization of the particle distribution (left column) and water saturation field (right column) for the five-spot problem at different times: $(a)/(b)$ $t = 0$; $(c)/(d)$ $t = 0.056$; $(e)/(f)$ $t = 0.09$.

cle methods, the present approach does not require any implicit or explicit mapping functions into a reference space of uniform resolution, nor does it require global transforms or data structures. This makes it particularly well suited for the implementation on large-scale distributed-memory computers and for problems in complex geometries or where the resolution requirements are not priorly known.

The present method relies on pairwise interaction potentials according to which the particles self-organize in an energy minimization process. Together with dynamic insertion and removal of particles, this leads to a robust and efficient adaptation of the particle density to the features of the field functions to be represented. Remeshing is replaced by interpolation from the old set of particles before self-organization to the new, adapted set of particles. The adaptation potential hereby guarantees that the solution field is always well sampled and that no holes or clusters develop in the particle distribution. The method relies on DC approximations of differential operators on scattered sets of particles with varying kernel widths, as well as on DC particle-particle interpolation schemes with the Kronecker delta property. Computing the DC operators and the interpolation kernels requires inverting a small linear system of equations for each particle. These systems, however, only need to be solved once and all operator and interpolation kernels can be computed from the same inverse.

Compared to state-of-the-art multiresolution methods, the errors of the present method are competitive, as demonstrated by comparing them to those of the wavelet-particle method [8]. Compared to nonadaptive methods, the present method is beneficial whenever the solution has multi-scale features. In these cases, the present method requires much fewer particles. We have shown that the truncation errors of the discretization schemes correspond to those predicted by theory, both for particle-particle interpolation and for the DC PSE operators. We have validated the present method in two- and three-dimensional advection-diffusion problems where analytical solutions are available and have shown that the method can be used to address also more complex, nonlinear problems.

In its current form, however, the present method has a couple of limitations. A main limitation is that it is not conservative. Exact conservation of mass could be enforced by symmetric DC PSE operators. Constructing such operators, however, is an open problem for convergence orders larger than one, see also §2.2. Another limitation of the present formulation is that in global time-stepping schemes, such as Euler or Runge-Kutta, the

time step size is dictated by the CFL condition in the highest-resolution region. In applications where most of the particles are located in coarsely resolved regions, this is clearly inefficient. Multiresolution time-stepping schemes, such as the multirate Runge-Kutta scheme [78], are available to alleviate this. In most applications of multiresolution methods, however, the majority of the particles is located in high-resolution regions.
Current and future work is concerned with generalizing the presented method to also discretize the weak form of the governing equations. This requires particles with a nonzero physical volume that carry the extensive quantity associated with the field f. Weak formulations are favorable if f is discontinuous. It should then also be interesting to compare the present DC method to its UC counterpart. This will be possible once the particle volumes are evolved during the simulation. Since the particle distribution is well-ordered due to the self-organization, UC operators may be compatible for reasonable target errors, even though the neighbor lists will need to be larger. Another topic of future work is benchmarking the parallel scalability and efficiency of the present method as implemented based on the PPM library [4, 77], and coupling of the present method with multiresolution Poisson solvers for far-field interactions.

CHAPTER

SIX

Conclusions and Outlook

This thesis examined and discussed discretization correction of particle strength exchange (PSE) operators and introduced a novel multiresolution particle method using self-organizing particles. In particular, we studied the operators' numerical properties, their relations to other operator types, and their performance in practice. We summarize the key contributions of this thesis below and discuss opportunities for future work.

Discretization Correction of PSE operators We have presented a general framework for discretization correction of PSE operators. We have studied and discussed the properties of DC PSE operators both theoretically and empirically. As expected, we found significant differences to the properties of uncorrected (UC) PSE operators.
DC PSE operators were found more accurate than their UC counterparts. Particularly near boundaries and on irregular particle distributions, the discretization correction was found to be crucial for converging approximation errors. The higher accuracy of the DC PSE operators goes along with lower numerical diffusion and dispersion, which results in more restrictive time-step limits. One of the main advantages of the DC PSE operators is that they allow to significantly reduce the number of particles needed to evaluate the operator while maintaining or even increasing its accuracy. However, discretization correction is computationally expen-

sive because a small[1] linear system of equations has to be solved for each particle. The overall benefit of the discretization correction was therefore assessed in numerical experiments.
As expected, the discretization correction accounts for the largest part of the computational cost of the operator evaluation. This is particularly pronounced for moving particles because the operators need to be recomputed whenever the particle distribution changes. For fixed particle distributions the correction has to be performed only once, at the beginning of a simulation, and the use of DC operators is always worthwhile. This is true also when remeshing is done at every time step. The computational efficiency of the DC PSE operators thus depends on the type of simulation. We therefore tested it in different benchmarks and compared it to standard finite difference (FD) stencils and to UC PSE operators. We found that, if the particles are remeshed at every time step, the remeshing error dominates the overall error. Lagrangian FD schemes are then the most efficient choice because the standard finite-difference stencils are the most compact operators on uniform Cartesian meshes. In problems where frequent remeshing is not required, DC PSE operators can outperform both UC PSE operators and FD stencils. While UC PSE operators require a finer resolution and more neighbors to reach the same accuracy, FD stencils can only be employed if remeshing is done at every time step, in which case the remeshing error also demands a finer resolution. We therefore conclude that the computational cost of discretization correction is amortized by the gain in accuracy for advection-dominated problems with remeshing at large time intervals. If advection plays a minor role, Eulerian methods are more efficient.
We showed how DC PSE operators are formulated consistently for multiresolution particle distributions. We also highlighted how a single discretization correction can be (re)used to approximate derivatives of different degrees as well as to interpolate a field between two sets of scattered particles. This greatly improves the computational efficiency of simulations where different types of differential operators or interpolations are required.

[1] The number of equations and unknowns depends on the order of accuracy, the order of the approximated derivative, and the number of spatial dimensions. For a two-dimensional second-order operator approximating a first derivative, for example, there are six equations and unknowns; in three dimensions, there are ten.

A main limitation of discretization correction as presented in this thesis is that the exact conservativeness of the PSE operators is lost. More precisely, the symmetry in the particle-specific contributions to the derivative approximations is broken. A possible solution to this problem could involve approximating derivatives at off-particle locations for determining the pairwise strength exchange between two particles. This, however, is costly because $kN/2$ instead of N discretization corrections are required, where N is the number of particles and k is the average number of neighbors per particle. Future work dedicated to the restoration of conservativeness would be a valuable contribution.

Relatives of DC PSE Operators Several types of operators are available for approximating field derivatives on discrete (scattered) data points. The discretization correction framework presented in this thesis allowed us to investigate the relation between DC PSE operators and four other operator types. This helped to position DC PSE operators in the family of differential operators used in numerical methods and allowed a unifying view on method descriptions.
We have shown that DC PSE operators are directly related to FD stencils. For uniform Cartesian particle distributions and an infinitely peaked window function, DC PSE operators reduce to standard, classical FD stencils. Furthermore, we highlighted the relation to between DC PSE operators and the operators used in corrected SPH, RKPM, MLS, DRK, and redistribution schemes. Unlike all these operators, DC PSE operators for approximating derivatives of even order do not impose any condition on the zeroth moment of the kernel function. We found that this freedom to adjust the zeroth moment can lead to better stability in explicit time stepping. DC PSE operators thus provide more flexibility than the other operators. Nevertheless, DC PSE operators become equivalent to other operators in special cases: if

(i) the zeroth moment of a DC PSE operator is set to zero,

the DC PSE operator is equivalent to the operators used in corrected smoothed particle dynamics or reproducing kernel particle methods. If additionally

(ii) the volumes of the particles are all set to one,

it is equivalent to the operators used in moving least squares methods or

differential reproducing kernel methods. If in addition to (i) also

(iii) the number of neighbors of each particle is the identical to the number of moment conditions and

(iv) the explicit Euler time stepping scheme or the explicit second-order Runge-Kutta time stepping scheme is used

the DC PSE operator for diffusion is equivalent to the diffusion operator of the redistribution methods by Shankar and van Dommelen or Gharakhani, respectively. If (iii) is not true, the diffusion operator of the redistribution method can be seen as a DC PSE operator based on a specific window function. In summary, the operator types considered here can be interpreted as special cases of DC PSE operators.
Future research could study how the value of the zeroth moment influences the properties of DC PSE operators. The operators' flexibility could then be exploited for improved operator designs. We are, however, not sure that general correlations can be found.

Objective Rating of Operator Properties There are many possible choices for differential operators in particle methods. The repertoire of diffusion operators is particularly large due to their widespread occurance. Usually, comparing different operators is either done on a conceptual level or based specific benchmark problems. While the former approach is usually not helpful in practice, the latter requires transferring the benchmark results to a possibly different type of application. We therefore introduced three problem-independent measures that quantify the accuracy, the stability in explicit time-stepping schemes, and the computational cost of discretized diffusion operators. These measures are designed for an objective comparison of different operators. We have shown in two test problems that they can be used to select efficient operators from a pool of given operators. In fact, we could demonstrate that the simulations using the predicted best[2] operators are more than twice as fast as the ones using a compact FD stencil. While the compact FD stencil was the most accurate operator in our pool, and the one with the lowest computational cost, the appointed operator still "won" because it allowed larger time steps. As this outcome is not obvious, we conclude that the presented quality measures were successful in the two cases considered.

[2] appointed by the presented objective rating framework.

Furthermore, we used the three quality measures in a parameter study for UC and DC PSE operators. We found the operator properties to form characteristic landscapes in parameter space. These results help develop a deeper understanding of the correlations between the free operator parameters and the operator properties. They are expected to be useful for operator tuning and operator design.
Even though we demonstrated that the three quality measures can rate operators of different types, we only considered PSE operators in the parameter study. It should be interesting to evaluate the measures also for collections of different operators in order to enable comparisons across numerical methods.

Adaptive Multiresolution Simulations using Self-Organizing Lagrangian Particles We introduced a novel numerical method for the solution of multiresolution transport problems. It uses Lagrangian particles that self-organize into well-ordered multiresolution configurations according to particle-particle interaction potentials. The self-organization includes a particle insertion and removal strategy, leading to dynamic adaptation of the number of particles to the resolution requirements. The method neither involves a mapping function, nor multiresolution meshes, but is inherently adaptive to the current solution. This simplifies its implementation and parallelization and renders it well suited for problems involving complex geometries or resolution requirements that are not known in advance. As the method is based on the strong formulation of the equations to be solved, it requires discretization correction both for differential operators and interpolation. We demonstrated that DC PSE operators are a suitable choice. We further demonstrated that the method outperforms nonadaptive methods if the solution has multiple scales, and that it is competitive with the particle-wavelet method, a state-of-the-art multiresolution particle method.
Future work is concerned with generalizing the method to the weak formulation, which is beneficial for discontinuous solutions. The main issue there is the evolution of the particle volumes in combination with dynamic particle insertion and removal. The next step is to couple the method with multiresolution Poisson solvers for far-field interactions.
In the weak formulation it could also be interesting to use UC PSE operators. Since the particle configuration is well ordered due to self-organization, there is a good chance that UC operators will perform reasonably

well. Their reduced computational cost would then render the method more attractive whenever intermediate accuracy is sufficient.
The self-organization of the particles relies on a robust ground state of the interaction potential. As potentials can be designed to have specific ground states, a thorough comparison of different particle configurations could be interesting. We suggest comparing different interaction potentials for self-organization using the three quality measures introduced in this thesis. As the operator properties depend on the particle distribution, their rating could provide a means of selecting good interaction potentials.
Finally, future work should include benchmarking the parallel scalability of the method.

APPENDIX

A

Test Cases

A.1 A Two-Dimensional Advection-Diffusion Test Case

We describe the two-dimensional advection-diffusion problem that is used as a test case in §2.4.2.2 and provide details about the numerical methods used.

A.1.1 Test Case Description

We solve the dimensionless advection-diffusion equation

$$\frac{\partial f(\boldsymbol{x},t)}{\partial t} + \nabla \cdot (f(\boldsymbol{x},t)\boldsymbol{u}(\boldsymbol{x})) = \frac{\nabla^2 f(\boldsymbol{x},t)}{\mathrm{Pe}}\,,\ \boldsymbol{x} \in \Omega,\, t \in [0,T],$$

for the unknown field $f(\boldsymbol{x}) = f(x,y)$ in the unit square $\Omega = [0,1]^2$. The Péclet number Pe represents the ratio between advection and diffusion. Higher Péclet numbers thus characterize more advection-dominated problems. The advection velocity field $\boldsymbol{u}(\boldsymbol{x}) = (y,0)$ is that of a Couette flow between two parallel flat walls (neglecting pressure gradients). The wall at $y = 0$ is fixed whereas the one at $y = 1$ moves with unit speed. Both walls are subject to homogeneous Neumann boundary conditions, and we

assume periodicity along x. The initial condition is given by

$$f_0(\boldsymbol{x}) = f(\boldsymbol{x}, t=0) = K + \sum_{k=0}^{3} (-1)^k \frac{\cos(2(2k+1)\pi x)}{2k+1},\ K = \text{const.},$$

which defines a smooth (regularized) approximation to a rectangular pulse. We consider four different Péclet numbers: $\mathrm{Pe} \in \{1, 10, 100, 1000\}$.

A.1.2 Reference Solution

We compare the numerical results obtained with UC and DC PSE operators to a reference solution at final time

$$T = \frac{\mathrm{Pe} + 2 - 2\sqrt{\mathrm{Pe}+1}}{\mathrm{Pe}}.$$

This is the time at which the mass diffusion length $\sqrt{4T/\mathrm{Pe}}$, added to the maximum distance traveled by a particle due to advection, is one. The reference solution is computed using a Lagrangian finite difference method (see §A.1.3.3) on a high-resolution grid of 1000×1000 nodes for $\mathrm{Pe} = \{1, 10, 100\}$ and 1600×1600 for $\mathrm{Pe} = 1000$. The L_∞ norm of the relative error — normalized by the concentration range

$$\max_{i,j} f_{i,j}^{T/\Delta t} - \min_{i,j} f_{i,j}^{T/\Delta t}$$

of the reference solution — is computed after interpolating the reference solution from the grid to the particle locations. The interpolation is linear. The interpolation error is considered insignificant compared to the errors from the comparably low resolved simulations.

A.1.3 Method Details

We implement the numerical methods described in the following in Fortran 90, compiled with the Intel Fortran Compiler v11 with optimization flag `-O3`, and run the simulations on Intel Xeon QuadCore 2.8GHz processors with 2GB RAM per core (4 simulations per processor, no multithreading). The linear systems of equations of the discretization correction are solved using the LU factorization provided by LAPACK [3].

A.1.3.1 PSE Operators

All simulations start from a Cartesian particle distribution of resolution h and use explicit Euler time stepping with $\Delta t \lesssim 0.15h^2\mathrm{Pe}$. Since the advection velocity field is divergence-free, the particles retain equal and constant volumes $v = h^2$. Their strengths are initialized to $F_p(t=0) = vf_0(\boldsymbol{x}_p(t=0))$. The time evolution of the particle positions and strengths is given by:

$$\begin{aligned}\boldsymbol{x}_p(t+\Delta t) &= \boldsymbol{x}_p(t) + \Delta t\, \boldsymbol{u}(\boldsymbol{x}_p(t))\,,\\ F_p(t+\Delta t) &= F_p(t) + \frac{\Delta t}{\mathrm{Pe}}\left[Q_h^{(2,0)} + Q_h^{(0,2)}\right] F_p(t)\,,\end{aligned}$$

where the subscript $p = 1, \ldots, N$ is the particle index. The operators $Q_h^{\boldsymbol{\beta}}$ are chosen as either DC or UC PSE operators with varying c and r_c (see §2.4.2.2). The discretization correction is recomputed at each time step unless the particles are remeshed. For irregular particle distributions, the zeroth-order moment of the DC kernels is fixed to $Z_h^{\mathbf{0}} = 5$. The homogeneous Neumann boundary conditions are imposed using mirror particles in a band of width r_c outside the domain (method of images).
Remeshing is done by interpolating the particle strengths to a new set of particles with uniform Cartesian positions $\boldsymbol{x}_p$ as

$$F(\boldsymbol{x}_p) = \sum_{q=1}^{N} \tilde{F}(\tilde{\boldsymbol{x}}_q) M_4'\left(\frac{x_p - \tilde{x}_q}{h}\right) M_4'\left(\frac{y_p - \tilde{y}_q}{h}\right)\,, \tag{A.1}$$

with the third-order accurate M_4' interpolation kernel, introduced as the W_4-function by Monaghan [64]:

$$M_4'(z) = \begin{cases} 0, & |z| > 2,\\ \frac{1}{2}(2-|z|)^2(1-|z|), & 1 \le |z| \le 2,\\ 1 - \frac{5}{2}z^2 + \frac{3}{2}|z|^3, & |z| \le 1. \end{cases} \tag{A.2}$$

A.1.3.2 Eulerian Finite Differences

The solution $f(\boldsymbol{x}, t)$ is approximated on a uniform Cartesian grid of resolution h. The values $f^n_{i,j}$ at all grid nodes $\{i, j\}$ are evolved in time ($t = n\Delta t$) as

$$f^{n+1}_{i,j} = f^n_{i,j} + \frac{\Delta t}{h^2}\left[\frac{1}{\mathrm{Pe}'_j}(f^n_{i+1,j} - 2f^n_{i,j} + f^n_{i-1,j}) + \frac{1}{\mathrm{Pe}}(f^n_{i,j+1} - 2f^n_{i,j} + f^n_{i,j-1}) - y_j h(f^n_{i,j} - f^n_{i-1,j})\right],$$

where $\mathrm{Pe}'_j = 2\mathrm{Pe}/(2 - y_j h\mathrm{Pe})$ if $\mathrm{Pe} \leq 2/h$, and $\mathrm{Pe}'_j = \infty$ else. We use a time step of $\Delta t \lesssim \min[0.15h^2\mathrm{Pe}, 0.30h]$. The scheme is second-order accurate for $\mathrm{Pe} \leq 2/h$ and first order otherwise. The Neumann boundary conditions are again imposed using the method of images.

A.1.3.3 Lagrangian Finite Differences

The solution is approximated on a uniform Cartesian grid, but advection and diffusion are treated using a time-splitting scheme. In each time step, the grid nodes are advected with the flow and remeshed using the M'_4 kernel. Afterward, diffusion is computed using the FD scheme

$$f^{n+1}_{i,j} = f^n_{i,j} + \frac{\Delta t}{h^2\mathrm{Pe}}\left[f^n_{i+1,j} + f^n_{i-1,j} + f^n_{i,j+1} + f^n_{i,j-1} - 4f^n_{i,j}\right]$$

with a time step of $\Delta t \lesssim 0.15h^2\mathrm{Pe}$. The method of images is applied at the Neumann boundaries, and periodicity is used along x.

A.2 A Three-Dimensional Advection-Diffusion Test Case

The test case consists of solving the advection-diffusion equation

$$\frac{\partial f}{\partial t} + \nabla \cdot (f\boldsymbol{v}) \;=\; \nabla \cdot (\nu \nabla f) \text{ in } \mathbb{R}^3 \times [0, T]$$

with the flow field, the viscosity field, and the initial concentration given by

$$\begin{aligned} \boldsymbol{v}(\boldsymbol{x}) &= [\cos x \sin y \sin z,\ \sin x \cos y \sin z,\ \sin x \sin y \cos z]^{\mathrm{T}}, \\ \nu(\boldsymbol{x}) &= \sin x \sin y \sin z + 1, \text{ and} \\ f_0(\boldsymbol{x}) &= f(\boldsymbol{x}, t=0) \ = \ \sin x \sin y \sin z, \end{aligned}$$

respectively. The analytical solution is

$$f(\boldsymbol{x}, t) \ = \ \sin x \sin y \sin z \exp(-3t)\,.$$

The problem is computed in a cube of side length $L = 2\pi$ with periodic boundary conditions, thus no boundary effects affect the numerical solution.
Explicit Euler time stepping is used for time integration, hence,

$$\begin{aligned} f_p(t+\Delta t) &= \frac{V_p(t)}{V_p(t+\Delta t)}[f_p(t) + \Delta t Q_h f_p(t)], \\ \boldsymbol{x}_p(t+\Delta t) &= \boldsymbol{x}_p(t) + \Delta t \boldsymbol{v}(\boldsymbol{x}_p(t)), \\ V_p(t+\Delta t) &= V_p(t)[1 + \Delta t \nabla \cdot \boldsymbol{v}(\boldsymbol{x}_p(t))], \end{aligned}$$

where the subscript p indicates properties of particle p. For remeshing, the concentration field f is interpolated from the current particle positions $\tilde{\boldsymbol{x}}_p$ to the positions $\boldsymbol{x}_p$ of the remeshed particles using the M_4'-kernel given in equation (A.2):

$$f_p(t) \ = \ \frac{1}{h^3} \sum_q \tilde{v}_q(t) \tilde{f}_q(t) \prod_{i=1}^{3} M_4' \left[\frac{(\boldsymbol{x}_p - \tilde{\boldsymbol{x}}_q(t))_i}{h} \right].$$

APPENDIX

B

Limiting FD Stencil of the Second-Order n-Dimensional Anisotropic Diffusion Operator

The anisotropic diffusion operator

$$Df(\boldsymbol{x}) = \nabla \cdot (\boldsymbol{L}(\boldsymbol{x}) \nabla f(\boldsymbol{x})) \;=\; \sum_{i,j=1}^{n} \frac{\partial}{\partial x_i} \left(L_{ij}(\boldsymbol{x}) \frac{\partial f(\boldsymbol{x})}{\partial x_j} \right)$$

can be approximated with second-order accuracy by the PSE operator [26]

$$Qf(\boldsymbol{x}) = \frac{1}{\varepsilon^2} \int_{\mathbb{R}^n} \sum_{i,j=1}^{n} M_{ij}(\boldsymbol{x}, \boldsymbol{y}) \psi_{ij}^{\varepsilon}(\boldsymbol{y} - \boldsymbol{x}) (f(\boldsymbol{y}) - f(\boldsymbol{x})) d\boldsymbol{y} \,. \tag{B.1}$$

In order for the operator to be symmetric (and hence conservative), the matrix $\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{y})$ is chosen as [26]

$$\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{m}(\boldsymbol{x}) + \boldsymbol{m}(\boldsymbol{y})}{2} \,, \quad \text{with } \boldsymbol{m}(\boldsymbol{x}) \;=\; \boldsymbol{L} - \frac{1}{n+2} \mathrm{Tr}(\boldsymbol{L}) \boldsymbol{I} \,. \tag{B.2}$$

The continuous moment conditions for the kernel function $\psi_{ij}(\boldsymbol{z})$ are

$$Z_{ij}^{\boldsymbol{\alpha}} = 0 \text{ for } 1 \leq |\boldsymbol{\alpha}| \leq 3 \text{ and } |\alpha| \neq 2, \text{ and}$$

$$\sum_{i,j=1}^{n} m_{ij}(\boldsymbol{x}) Z_{ij}^{\boldsymbol{e}_k+\boldsymbol{e}_l} = 2L_{kl}(\boldsymbol{x}) \quad \forall k,l \in [1,n], \text{ with } Z_{ij}^{\boldsymbol{\alpha}} = \int_{\mathbb{R}^n} \psi_{ij}(\boldsymbol{z}) \boldsymbol{z}^{\boldsymbol{\alpha}} d\boldsymbol{z}.$$

With a kernel function of the form $\psi_{ij}(\boldsymbol{z}) = z_i z_j (a_0 + a_1|\boldsymbol{z}|) \exp\left(-\boldsymbol{z}^2\right)$, $i,j = 1,\ldots,n$, and equation (B.2), this translates into the moment conditions

$$\begin{aligned} Z_{ij}^{\boldsymbol{e}_i+\boldsymbol{e}_j} &= 1 \text{ if } i \neq j, \text{ and} \\ Z_{ii}^{2\boldsymbol{e}_i} &= 3\,. \end{aligned} \tag{B.3}$$

The integral operator (B.1) is discretized as

$$Q_h f(\boldsymbol{x}) = \frac{1}{\varepsilon^2} \sum_p v_p \sum_{i,j=1}^{n} M_{ij}(\boldsymbol{x}, \boldsymbol{x}_p) \psi_{ij}^{\varepsilon}(\boldsymbol{x}_p - \boldsymbol{x})(f(\boldsymbol{x}_p) - f(\boldsymbol{x})) \tag{B.4}$$

and for symmetric particle distributions the discrete moment conditions are analogous to the continuous conditions (B.3). The operator for a uniform Cartesian particle distribution is hence given by

$$\begin{aligned} Q_h f(\boldsymbol{x}) = \frac{c^{n+2}}{2\varepsilon^2} \sum_{|\boldsymbol{k}|^2=0}^{l} \sum_{i,j=1}^{n} &(m_{ij}(\boldsymbol{x}) + m_{ij}(\boldsymbol{x} + \boldsymbol{k}c\varepsilon))(f(\boldsymbol{x} + \boldsymbol{k}c\varepsilon) - f(\boldsymbol{x})) \\ &\times (a_0 + a_1 c|\boldsymbol{k}|) k_i k_j \exp\left(-c^2|\boldsymbol{k}|^2\right), \boldsymbol{k} \in \mathbb{Z}^n, \end{aligned} \tag{B.5}$$

where $\boldsymbol{k}$ is a vector of integers, $l = \lfloor r_c/h \rfloor^2$, and r_c is arbitrary, but large enough to contain the resulting FD stencil. The coefficients a_0 and a_1 are obtained by solving the system of equations that results from substituting the kernel into conditions (B.3), thus:

$$\begin{aligned} c^{n+4} \sum_{|\boldsymbol{k}|^2=0}^{l} (a_0 + a_1 c|\boldsymbol{k}|) k_p^2 k_q^2 \exp\left(-c^2|\boldsymbol{k}|^2\right) &= 1\,,\ p \neq q, \\ c^{n+4} \sum_{|\boldsymbol{k}|^2=0}^{l} (a_0 + a_1 c|\boldsymbol{k}|) k_p^4 \exp\left(-c^2|\boldsymbol{k}|^2\right) &= 3. \end{aligned}$$

This system can be solved analytically and the coefficients a_0 and a_1 are

found as

$$a_0 = \frac{3B - D}{c^{n+4}(BC - AD)}, \qquad a_1 = \frac{C - 3A}{c^{n+5}(BC - AD)}, \text{ where} \tag{B.6}$$

$$A = \sum_{|\boldsymbol{k}|^2=0}^{l} k_p^2 k_q^2 \exp\left(-c^2|\boldsymbol{k}|^2\right), \quad B = \sum_{|\boldsymbol{k}|^2=0}^{l} |\boldsymbol{k}| k_p^2 k_q^2 \exp\left(-c^2|\boldsymbol{k}|^2\right), \; p \neq q,$$

$$C = \sum_{|\boldsymbol{k}|^2=0}^{l} k_p^4 \exp\left(-c^2|\boldsymbol{k}|^2\right), \qquad D = \sum_{|\boldsymbol{k}|^2=0}^{l} |\boldsymbol{k}| k_p^4 \exp\left(-c^2|\boldsymbol{k}|^2\right).$$

Substituting these coefficients into equation (B.5) yields

$$Q_h f(\boldsymbol{x}) = \frac{1}{2h^2 U} \sum_{|\boldsymbol{k}|^2=0}^{l} \sum_{i,j=1}^{n} (m_{ij}(\boldsymbol{x}) + m_{ij}(\boldsymbol{x} + \boldsymbol{k}c\varepsilon))(f(\boldsymbol{x} + \boldsymbol{k}c\varepsilon) - f(\boldsymbol{x})),$$
$$\times (V + W|\boldsymbol{k}|) k_i k_j \exp\left(-c^2|\boldsymbol{k}|^2\right)$$

where

$$U = BC - AD, V = 3B - D, W = C - 3A.$$

This can be rewritten as

$$\begin{aligned} Q_h f(\boldsymbol{x}) = & \sum_{|\boldsymbol{k}|^2=1} \sum_{i,j=1}^{n} (m_{ij}(\boldsymbol{x}) + m_{ij}(\boldsymbol{x} + \boldsymbol{k}c\varepsilon))(f(\boldsymbol{x} + \boldsymbol{k}c\varepsilon) - f(\boldsymbol{x})) \\ & \times \frac{V_2 + W_2}{2h^2 U_3} k_i k_j \exp\left(-c^2(|\boldsymbol{k}|^2 - 1)\right) \\ & + \sum_{|\boldsymbol{k}|^2=2} \sum_{i,j=1}^{n} (m_{ij}(\boldsymbol{x}) + m_{ij}(\boldsymbol{x} + \boldsymbol{k}c\varepsilon))(f(\boldsymbol{x} + \boldsymbol{k}c\varepsilon) - f(\boldsymbol{x})) \\ & \times \frac{V_1 + \sqrt{2}W_1}{2h^2 U_3} k_i k_j \exp\left(-c^2(|\boldsymbol{k}|^2 - 2)\right) \end{aligned}$$

$$+\sum_{|\boldsymbol{k}|^2=3}^{l}\sum_{i,j=1}^{n}(m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}+\boldsymbol{k}c\varepsilon))(f(\boldsymbol{x}+\boldsymbol{k}c\varepsilon)-f(\boldsymbol{x}))$$
$$\times\frac{V_1+W_1|\boldsymbol{k}|}{2h^2U_3}k_ik_j\exp\left(-c^2(|\boldsymbol{k}|^2-2)\right),$$

where

$$U_3=U\exp\left(3c^2\right),\qquad V_1=V\exp\left(c^2\right),\qquad V_2=V\exp\left(2c^2\right),$$
$$W_1=W\exp\left(c^2\right),\qquad W_2=W\exp\left(2c^2\right).$$

In the limit $c\to\infty$,

$$\lim_{c\to\infty}U_3=8(\sqrt{2}-1),\ \lim_{c\to\infty}V_1=-2,\ \lim_{c\to\infty}W_1=2,\ \text{and}$$
$$\lim_{c\to\infty}(V_2+W_2)=4(\sqrt{2}-1)(4-n).$$

Hence, we find the operator

$$\begin{aligned}\lim_{c\to\infty}Q_hf(\boldsymbol{x})=\frac{4-n}{4h^2}&\sum_{|\boldsymbol{k}|^2=1}\sum_{i,j=1}^{n}(m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}+\boldsymbol{k}c\varepsilon))\\&\times(f(\boldsymbol{x}+\boldsymbol{k}c\varepsilon)-f(\boldsymbol{x}))k_ik_j\\&+\frac{1}{8h^2}\sum_{|\boldsymbol{k}|^2=2}\sum_{i,j=1}^{n}(m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}+\boldsymbol{k}c\varepsilon))\\&\times(f(\boldsymbol{x}+\boldsymbol{k}c\varepsilon)-f(\boldsymbol{x}))k_ik_j.\end{aligned}\tag{B.7}$$

Using the vectors $\boldsymbol{h}_i=h\boldsymbol{e}_i$, $i=1,\dots,n$, where $\boldsymbol{e}_i$ is the unit vector along dimension i, this operator can be rewritten as

$$\begin{aligned}\lim_{c\to\infty}Q_hf(\boldsymbol{x})=\frac{4-n}{4h^2}\sum_{i=1}^{n}\big\{&\\&(m_{ii}(\boldsymbol{x})+m_{ii}(\boldsymbol{x}+\boldsymbol{h}_i))(f(\boldsymbol{x}+\boldsymbol{h}_i)-f(\boldsymbol{x}))\\&+(m_{ii}(\boldsymbol{x})+m_{ii}(\boldsymbol{x}-\boldsymbol{h}_i))(f(\boldsymbol{x}-\boldsymbol{h}_i)-f(\boldsymbol{x}))\big\}\end{aligned}$$

$$
\begin{aligned}
&+\frac{1}{8h^2}\sum_{\substack{i,k=1\\ i\neq k}}^{n}\Big\{\\
&\quad (m_{ii}(\boldsymbol{x})+m_{ii}(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_k))(f(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_k)-f(\boldsymbol{x}))\\
&+(m_{ii}(\boldsymbol{x})+m_{ii}(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_k))(f(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_k)-f(\boldsymbol{x}))\\
&+(m_{ii}(\boldsymbol{x})+m_{ii}(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_k))(f(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_k)-f(\boldsymbol{x}))\\
&+(m_{ii}(\boldsymbol{x})+m_{ii}(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_k))(f(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_k)-f(\boldsymbol{x}))\Big\}\\
&+\frac{1}{8h^2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\Big\{\\
&\quad (m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j)-f(\boldsymbol{x}))\\
&-(m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j)-f(\boldsymbol{x}))\\
&-(m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j)-f(\boldsymbol{x}))\\
&+(m_{ij}(\boldsymbol{x})+m_{ij}(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j)-f(\boldsymbol{x}))\Big\}\\
&=\frac{4-n}{2}\sum_{i=1}^{n}\left[m_{ii}(\boldsymbol{x})\frac{\partial^2 f}{\partial x_i^2}+\frac{\partial m_{ii}}{\partial x_i}\cdot\frac{\partial f}{\partial x_i}\right]\\
&+\frac{1}{2}\sum_{\substack{i,k=1\\ i\neq k}}^{n}\left[m_{ii}(\boldsymbol{x})\left(\frac{\partial^2 f}{\partial x_i^2}+\frac{\partial^2 f}{\partial x_k^2}\right)+\frac{\partial m_{ii}}{\partial x_i}\cdot\frac{\partial f}{\partial x_i}+\frac{\partial m_{ii}}{\partial x_k}\cdot\frac{\partial f}{\partial x_k}\right]\\
&+\sum_{\substack{i,j=1\\ i\neq j}}^{n}\left[m_{ij}(\boldsymbol{x})\frac{\partial^2 f}{\partial x_i x_j}+\frac{1}{2}\frac{\partial m_{ij}}{\partial x_i}\frac{\partial f}{\partial x_j}+\frac{1}{2}\frac{\partial m_{ij}}{\partial x_j}\frac{\partial f}{\partial x_i}\right]+O(h^2)\\
&=\sum_{i,j=1}^{n}\frac{\partial}{\partial x_i}\left[L_{ij}(\boldsymbol{x})\frac{\partial f}{\partial x_j}\right]+O(h^2)\;=\;Df(\boldsymbol{x})+O(h^2)\,.
\end{aligned}
$$

Expressing this in terms of the matrix $\boldsymbol{L}$, instead of the matrix $\boldsymbol{m}$, we find

the compact second-order FD stencil for anisotropic diffusion

$$
\begin{aligned}
\lim_{c\to\infty} Q_h f(\boldsymbol{x}) = \frac{4-n}{4h^2}\sum_{i=1}^{n}\Big\{ & \\
& (L_{ii}(\boldsymbol{x}) + L_{ii}(\boldsymbol{x}+\boldsymbol{h}_i))(f(\boldsymbol{x}+\boldsymbol{h}_i) - f(\boldsymbol{x})) \\
& +(L_{ii}(\boldsymbol{x}) + L_{ii}(\boldsymbol{x}-\boldsymbol{h}_i))(f(\boldsymbol{x}-\boldsymbol{h}_i) - f(\boldsymbol{x})) \\
& -\frac{1}{n+2}\sum_{k=1}^{n}\big[\\
& (L_{kk}(\boldsymbol{x}) + L_{kk}(\boldsymbol{x}+\boldsymbol{h}_i))(f(\boldsymbol{x}+\boldsymbol{h}_i) - f(\boldsymbol{x}))+ \\
& (L_{kk}(\boldsymbol{x}) + L_{kk}(\boldsymbol{x}-\boldsymbol{h}_i))(f(\boldsymbol{x}-\boldsymbol{h}_i) - f(\boldsymbol{x}))\big]\Big\} \\
& +\frac{1}{8h^2}\sum_{\substack{i,j=1\\ i\neq j}}^{n}\Big\{ \\
& (L_{ii}(\boldsymbol{x}) + L_{ii}(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{ii}(\boldsymbol{x}) + L_{ii}(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{ii}(\boldsymbol{x}) + L_{ii}(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{ii}(\boldsymbol{x}) + L_{ii}(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{ij}(\boldsymbol{x}) + L_{ij}(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& -(L_{ij}(\boldsymbol{x}) + L_{ij}(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& -(L_{ij}(\boldsymbol{x}) + L_{ij}(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{ij}(\boldsymbol{x}) + L_{ij}(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& -\frac{1}{n+2}\sum_{k=1}^{n}\big[\\
& (L_{kk}(\boldsymbol{x}) + L_{kk}(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i+\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{kk}(\boldsymbol{x}) + L_{kk}(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}+\boldsymbol{h}_i-\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{kk}(\boldsymbol{x}) + L_{kk}(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i+\boldsymbol{h}_j) - f(\boldsymbol{x})) \\
& +(L_{kk}(\boldsymbol{x}) + L_{kk}(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j))(f(\boldsymbol{x}-\boldsymbol{h}_i-\boldsymbol{h}_j) - f(\boldsymbol{x}))\big]\Big\}.
\end{aligned}
$$

APPENDIX

C

Evaluating the Quality Measures for Different Diffusion Operators

Closed-form expressions for the quality measures introduced in §4.2 can be derived for different types of diffusion operators. We exemplify this by deriving the measures A_s and S for four different operators: PSE diffusion operators, Fishelov diffusion operators, the diffusion operator of a resampling method, and a diffusion operator used in smoothed particle hydrodynamics. The cost measure C is universal as it simply approximates the number of particles used for evaluating $Q_h f(\boldsymbol{x})$.

C.1 PSE Diffusion Operators

Particle strength exchange (PSE) diffusion operators, briefly described in §1.3, were introduced by Degond and Mas-Gallic for the cases of isotropic and anisotropic viscosity, [26, 27]. In the following, we give their accuracy measures A_s and their stability measure S.

C.1.1 The Case of Isotropic Viscosity

For the PSE operators defined in equation (1.3), the accuracy measures are $A_{-2}(\boldsymbol{x}) = \infty$ and

$$A_s(\boldsymbol{x}) = \left[\sum_{|\boldsymbol{\alpha}|=1}^{s+2} \sum_{|\boldsymbol{\beta}|=s+2-|\boldsymbol{\alpha}|} \frac{1+\delta_{0,|\boldsymbol{\beta}|}}{2c^s \boldsymbol{\alpha}!\boldsymbol{\beta}!} \left| Z_h^{\boldsymbol{\alpha}+\boldsymbol{\beta}}(\boldsymbol{x}) - Y^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \right| \right]^{-1}, -1 \leq s < r,$$

$$A_r(\boldsymbol{x}) = \left[\sum_{|\boldsymbol{\alpha}|=1}^{r+2} \sum_{|\boldsymbol{\beta}|=r+2-|\boldsymbol{\alpha}|} \frac{1+\delta_{0,|\boldsymbol{\beta}|}}{2c^r \boldsymbol{\alpha}!\boldsymbol{\beta}!} \times \sum_{p \in \mathcal{N}(\boldsymbol{x})} \left| v_p \left(\frac{\boldsymbol{x}-\boldsymbol{x}_p}{\varepsilon} \right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p) \right| \right]^{-1}. \tag{C.1}$$

These expressions are derived by Taylor-expanding f and ν in equation (1.3) around $\boldsymbol{x}$ and finding a bound for the overall error (4.1), separating problem-dependent and problem-independent terms.
For DC operators, by construction $A_s(\boldsymbol{x}) = \infty$, $s < r$. In this case the only relevant accuracy measure is $A_r(\boldsymbol{x})$.
In order to express the stability measure S of a PSE diffusion operator, such that the time step limit in explicit Euler time stepping is given by equation (4.4), we consider the semi-discrete equation (4.3). Discretizing the time derivative by a forward difference results in the explicit Euler scheme

$$f(\boldsymbol{x}, t+\Delta t) = f(\boldsymbol{x}, t) + \Delta t\, Q_h f(\boldsymbol{x}, t).$$

A von Neumann stability analysis reveals that the limit on the time step guaranteeing exponential stability is

$$\Delta t < \frac{-2\mathrm{Im}(\omega(h\boldsymbol{k}))}{|\omega(h\boldsymbol{k})|^2} \quad \forall\, h\boldsymbol{k} \text{ s.t. } |h\boldsymbol{k}| \in]0, \sqrt{n}\pi],$$

where

$$\omega(h\boldsymbol{k}) = \frac{\hat{\imath}\nu}{\varepsilon^2} \sum_{p \in \mathcal{N}(\boldsymbol{x})} v_p \left[\exp\left(\hat{\imath} h\boldsymbol{k} \cdot \frac{\boldsymbol{x}_p - \boldsymbol{x}}{c\varepsilon} \right) - 1 \right] \eta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p)$$

is obtained as the dispersion relation for a wave $g(\boldsymbol{x},t)=\exp\left(\hat{\imath}(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t)\right)$ that satisfies the original equation (4.3). The problem-independent stability measure S then is

$$S=\min_{|h\boldsymbol{k}|\in]0,\sqrt{n\pi^2}]}\left\{\frac{-2\nu\mathrm{Im}(\omega_{\mathrm{mod}}(h\boldsymbol{k}))}{c^2\varepsilon^2|\omega_{\mathrm{mod}}(h\boldsymbol{k})|^2}\right\}, \tag{C.2}$$

thus a quantitative measure for the admissible time-step size.

C.1.2 The Case of Anisotropic Viscosity

The accuracy measures for the case of anisotropic viscosity are $A_{-2}(\boldsymbol{x})=\infty$ and

$$A_s(\boldsymbol{x})=\left[\sum_{|\boldsymbol{\alpha}|=1}^{s+2}\sum_{|\boldsymbol{\beta}|=s+2-|\boldsymbol{\alpha}|}\frac{\left(1+3\delta_{0,|\boldsymbol{\beta}|}\right)n}{2\boldsymbol{\alpha}!\boldsymbol{\beta}!}\left(\frac{|\mathrm{tr}(U)|}{n}+\frac{\lambda^U_{\max}-\lambda^U_{\min}}{2}\right.\right.$$
$$\left.\left.+\left|\frac{\mathrm{tr}(U)}{n}-\frac{\lambda^U_{\max}+\lambda^U_{\min}}{2}\right|\right)\right]^{-1},\ -1\leq s<r,$$
$$A_r(\boldsymbol{x})=\left[\sum_{|\boldsymbol{\alpha}|=1}^{r+2}\sum_{|\boldsymbol{\beta}|=r+2-|\boldsymbol{\alpha}|}\frac{1+\delta_{|\boldsymbol{\beta}|,0}}{4\boldsymbol{\alpha}!\boldsymbol{\beta}!}\right.$$
$$\left.\times\sum_{p\in\mathcal{N}}\mathrm{tr}(X_p)\left|v_p\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p)\left(\frac{\boldsymbol{x}-\boldsymbol{x}_p}{\varepsilon}\right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}}\right|\right]^{-1}.$$

The elements of the matrices X_p, $U\in\mathbb{R}^{n\times n}$ are

$$(U)_{ij}=Z_h^{\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{e}_i+\boldsymbol{e}_j}-Y^{\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{e}_i+\boldsymbol{e}_j},\quad (X_p)_{ij}=\left(\frac{\boldsymbol{x}-\boldsymbol{x}_p}{\varepsilon}\right)^{\boldsymbol{e}_i+\boldsymbol{e}_j},$$

and $\lambda^U_{\min}$ and $\lambda^U_{\max}$ denote the minimum and maximum eigenvalues of U.

The stability measure S is derived as in the isotropic case, resulting in

$$S = \min_{|h\boldsymbol{k}|\in]0,\sqrt{n\pi^2}]} \left[\frac{-2\alpha_{\min}^{h^2\mathrm{Im}(\Omega)}}{\max(\alpha_{\min}^{h^2\mathrm{Re}(\Omega)^2}, \alpha_{\max}^{h^2\mathrm{Re}(\Omega)^2}) + \max(\alpha_{\min}^{h^2\mathrm{Im}(\Omega)^2}, \alpha_{\max}^{h^2\mathrm{Im}(\Omega)^2})} \right],$$

$$\alpha_{\min}^X = \lambda_{\min}^X - \frac{\mathrm{tr}(X)}{n+2}, \quad \alpha_{\max}^X = \lambda_{\max}^X - \frac{\mathrm{tr}(X)}{n+2},$$

where the elements of the matrix Ω are defined as

$$\Omega_{ij} = \frac{\hat{\imath}}{\varepsilon^4} \sum_{p\in\mathcal{N}(\boldsymbol{x})} v_p \left(\exp(\hat{\imath}\boldsymbol{k}\cdot(\boldsymbol{x}_p - \boldsymbol{x})) - 1\right) \eta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p)(\boldsymbol{x}-\boldsymbol{x}_p)^{\boldsymbol{e}_i+\boldsymbol{e}_j}.$$

The viscosity ν in condition (4.4) has to be replaced by $\mathrm{tr}(\boldsymbol{\nu})$ in the anisotropic case. We remark that condition (4.4) is then sufficient, but not necessary.

C.2 A Fishelov Diffusion Operator

We consider a diffusion operator following the Fishelov scheme [33],

$$Q_h f(\boldsymbol{x}) = \nu h^3 \sum_p f(\boldsymbol{x}_p)\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{x}_p),$$

where

$$\eta_\varepsilon(\boldsymbol{z}) = \nabla^2 W(\boldsymbol{z},\varepsilon), \quad W(\boldsymbol{z},\varepsilon) = \frac{1}{\pi\sqrt{\pi}\varepsilon^3}\exp(-z^2/\varepsilon^2),\ z = |\boldsymbol{z}|.$$

Thus,

$$\eta_\varepsilon(\boldsymbol{z}) \ = \ \frac{1}{\pi\sqrt{\pi}\varepsilon^5}\left(4\frac{z^2}{\varepsilon^2} - 6\right)\exp(-z^2/\varepsilon^2)$$

for constant isotropic viscosity ν.
A Taylor expansion of f around $\boldsymbol{x}$ enables bounding the overall error $\epsilon(\boldsymbol{x})$,

as in equation (4.1), with the accuracy measures

$$A_s = \left[\sum_{|\boldsymbol{\alpha}|=s+2} \frac{c^{3-s}}{\pi\sqrt{\pi}\boldsymbol{\alpha}!} \left| \sum_p \left(4 \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 - 6 \right) \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}} \right.\right.$$
$$\left.\left. \times \exp\left(- \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 \right) - Y^{\boldsymbol{\alpha}} \right| \right]^{-1}, \ -2 \le s \le 1,$$
$$A_2 = \left[\sum_{|\boldsymbol{\alpha}|=4} \frac{c}{\pi\sqrt{\pi}\boldsymbol{\alpha}!} \sum_p \left| \left(4 \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 - 6 \right) \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}} \right.\right.$$
$$\left.\left. \times \exp\left(- \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 \right) \right| \right]^{-1},$$

where $Y^{\boldsymbol{\alpha}} = 1$ if $\boldsymbol{\alpha} = 2\boldsymbol{e}_i$ and $Y^{\boldsymbol{\alpha}} = 0$ else. The stability measure S is obtained by substituting the dispersion relation

$$\omega_{\mathrm{mod}}(h\boldsymbol{k}) = \frac{\hat{\imath}\nu c^3}{\pi\sqrt{\pi}\varepsilon^2} \sum_p e^{\hat{\imath}h\boldsymbol{k}\cdot\frac{\boldsymbol{x}_p - \boldsymbol{x}}{c\varepsilon}} \left(4 \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 - 6 \right)$$
$$\times \exp\left(- \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 \right)$$

into equation (C.2).

C.3 The Diffusion Operator of a Resampling Method

We derive the accuracy and stability measures for a resampling method [24] with constant viscosity ν. From the evolution equation

$$f(\boldsymbol{x}_q, t + \Delta t) = \sum_p v_p(t) f(\boldsymbol{x}_p, t) \mathcal{G}(\boldsymbol{x}_q - \boldsymbol{x}_p, \nu\Delta t),$$
$$\mathcal{G}(\boldsymbol{z}, \nu\Delta t) = \frac{1}{(4\pi\nu\Delta t)^{3/2}} \exp\left(-\frac{z^2}{4\nu\Delta t} \right),$$

one can construct the equivalent diffusion operator for explicit Euler time stepping as

$$Q_h f(\boldsymbol{x}) \;=\; \sum_p v_p(t) f(\boldsymbol{x}_p, t) \frac{\mathcal{G}(\boldsymbol{x}-\boldsymbol{x}_p, \nu\Delta t)}{\Delta t} - \frac{f(\boldsymbol{x},t)}{\Delta t}. \tag{C.3}$$

Fixing the CFL number $\alpha_{\mathrm{CFL}} = \Delta t \nu / h^2$, the accuracy measures are

$$\begin{aligned} A_s = \Bigg[\sum_{|\boldsymbol{\alpha}|=s+2} \Bigg| \frac{1}{8\pi\sqrt{\alpha_{\mathrm{CFL}}^5 \pi} c^2 \boldsymbol{\alpha}!} \sum_p \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}} \\ \times \exp\left(-\frac{1}{4\alpha_{\mathrm{CFL}} c^2} \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 \right) - Y^{\boldsymbol{\alpha}} \Bigg| \Bigg]^{-1}, \; -2 \le s \le 1, \\ A_2 = 8\pi\sqrt{\alpha_{\mathrm{CFL}}^5 \pi} c^2 \Bigg[\sum_{|\boldsymbol{\alpha}|=4} \frac{1}{\boldsymbol{\alpha}!} \sum_p \Bigg| \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}} \\ \times \exp\left(-\frac{1}{4\alpha_{\mathrm{CFL}} c^2} \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right|^2 \right) \Bigg| \Bigg]^{-1}, \end{aligned} \tag{C.4}$$

where $Y^{\boldsymbol{\alpha}} = 1$ if $\boldsymbol{\alpha} = \mathbf{0}$ or $\boldsymbol{\alpha} = 2\boldsymbol{e}_i$ and $Y^{\boldsymbol{\alpha}} = 0$ else. The stability measure S can not be given in closed form, but should be defined as the maximum CFL number α_{CFL} such that

$$\left| \sum_p \frac{\exp\left(\hat{\imath} h \boldsymbol{k} \cdot \frac{\boldsymbol{x}_p - \boldsymbol{x}}{c\varepsilon} - \frac{1}{4\alpha_{\mathrm{CFL}}} \left| \frac{\boldsymbol{x}_p - \boldsymbol{x}}{c\varepsilon} \right|^2 \right)}{\sqrt{4\pi\alpha_{\mathrm{CFL}}}^3} \right| < 1 \; \forall \, |h\boldsymbol{k}| \in]0, \sqrt{n}\pi]. \tag{C.5}$$

Numerical exploration of equation (C.5), however, reveals that there is a lower bound rather than an upper bound on the CFL number. The left panel of figure C.1 shows the dependence of the numerical amplification on the CFL number for uniform Cartesian particle distributions and $r_c/h = 6$. The dashed line marks the CFL number where the amplification is one. The CFL number hence has to be chosen larger than this minimum value $(\alpha_{\mathrm{CFL}})_{\min}$. The minimum value $(\alpha_{\mathrm{CFL}})_{\min}$ depends on the cutoff radius. This dependence is shown in the right panel of figure C.1. Note that in this

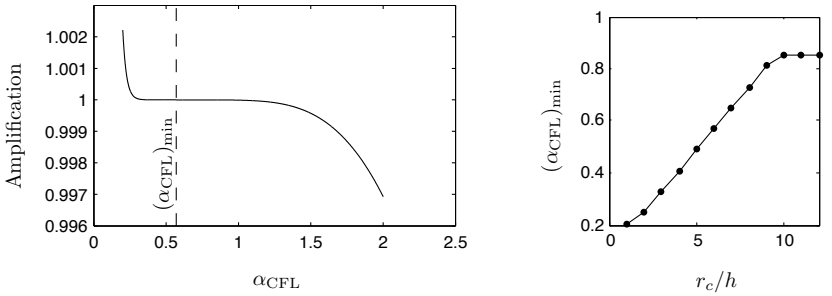

Figure C.1: Numerical stability of the resampling method. Left panel: Amplification versus CFL number α_{CFL} for $r_c/h = 6$. The dashed line marks the CFL number $(\alpha_{\mathrm{CFL}})_{\mathrm{min}}$ where the amplification is one. Right panel: Dependence of the minimum CFL number on the cutoff radius.

method the highest amplifications occur for the smallest wavenumbers. This is because the amplification is simply the numerical quadrature of the heat kernel $\mathcal{G}$ using the rectangular rule. Hence, it is an artifact of the discretization error of the method.
As we did not find an upper bound for the CFL number, a stability measure as it exists for the other diffusion operators cannot be defined. Instabilities here thus occur for small time steps rather than for large ones.

C.4 A Robust Diffusion Operator in Smoothed Particle Hydrodynamics

Cleary and Monaghan [18] introduced a robust (towards particle disorder and jumps in the viscosity) diffusion operator for smoothed particle hydrodynamics:

$$Q_h f(\boldsymbol{x}) \;=\; \sum_p \frac{m_p}{\rho_p}(f(\boldsymbol{x}_p) - f(\boldsymbol{x}))\mu(\boldsymbol{x}, \boldsymbol{x}_p)\eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p), \tag{C.6}$$

where the ratio of mass m_p and density ρ_p can be replaced by the particle volume v_p,

$$\mu(\boldsymbol{x}, \boldsymbol{x}_p) = \frac{4\nu(\boldsymbol{x})\nu(\boldsymbol{x}_p)}{\nu(\boldsymbol{x}) + \nu(\boldsymbol{x}_p)}, \quad \boldsymbol{z}\eta_\varepsilon(\boldsymbol{z}) = -\nabla W(\boldsymbol{z}, \varepsilon)\,,$$

and $W(\boldsymbol{z}, \varepsilon)$ is as in equation (C.2). Thus,

$$\eta_\varepsilon(\boldsymbol{z}) = \frac{2}{\pi\sqrt{\pi}\varepsilon^5} \exp\left(-\frac{z^2}{\varepsilon^2}\right).$$

Due to the nonlinear mean $\mu(\boldsymbol{x}, \boldsymbol{x}_p)$, we did not find a bound for the overall error of the form given in equation (4.2). However, using the multivariate version of Faá di Bruno's formula [19, 42], one can evaluate $D^{\boldsymbol{\beta}}\mu(\boldsymbol{x}, \boldsymbol{x})$,

where the derivative is applied with respect to the second argument, as

$$\begin{aligned} D^{\boldsymbol{\beta}}\mu(\boldsymbol{x},\boldsymbol{y}) &= \frac{\partial^{|\boldsymbol{\beta}|}}{\partial^{\beta_1} y_1 \dots \partial^{\beta_n} y_n}\mu(\boldsymbol{x},\boldsymbol{y}) \\ &= \sum_{1\le\lambda\le|\boldsymbol{\beta}|} \mu_\nu^{(\lambda)}(\nu_p) \sum_{s=1}^{\lambda} \sum_{p_s(\boldsymbol{\beta},\lambda)} (\boldsymbol{\beta}!) \prod_{j=1}^{s} \frac{[D^{\boldsymbol{l}_j}\nu_p]^{k_j}}{(k_j!)[\boldsymbol{l}_j!]^{k_j}}, \end{aligned}$$

where $\mu_\nu(\nu_p) = \frac{4\nu\nu_p}{\nu+\nu_p}$, $\nu = \nu(\boldsymbol{x})$, $\nu_p = \nu(\boldsymbol{y})$, $\boldsymbol{l}_i \in \mathbb{N}^n$,

$$\begin{aligned} p_s(\boldsymbol{\beta},\lambda) =& \{(k_1,\dots,k_s;\boldsymbol{l}_1,\dots,\boldsymbol{l}_s) : k_i > 0, \\ & \boldsymbol{0} \prec \boldsymbol{l}_1 \prec \dots \prec \boldsymbol{l}_s, \sum_{i=1}^{s} k_i = \lambda \text{ and } \sum_{i=1}^{s} k_i \boldsymbol{l}_i = \boldsymbol{\beta}\}. \end{aligned}$$

For $\boldsymbol{a},\boldsymbol{b} \in \mathbb{N}^n$, $\boldsymbol{a} \prec \boldsymbol{b}$ means that one of the following holds:

(i) $|\boldsymbol{a}| < |\boldsymbol{b}|$,

(ii) $|\boldsymbol{a}| = |\boldsymbol{b}|$ and $a_1 < b_1$, or

(iii) $|\boldsymbol{a}| = |\boldsymbol{b}|$, $a_1 = b_1, \dots, a_k = b_k$ and $a_{k+1} < b_{k+1}$ for some k, $1 \le k \le n$.

In words: the vectors $\boldsymbol{l}_i$ are the partitions of $\boldsymbol{\beta}$, the scalars k_i their multiplicities. The variable λ counts the number of partitions while the variable s counts the number of different partitions. For example,

$$\boldsymbol{\beta} = \underbrace{(3,0,0)}_{\substack{\boldsymbol{l}_1=(3,0,0),s=1,\\ \lambda=1,k_1=1}} = \underbrace{(2,0,0)+(1,0,0)}_{\substack{\boldsymbol{l}_1=(1,0,0),\boldsymbol{l}_2=(2,0,0),\\ s=2,\lambda=2,k_1=1,k_2=1}} = \underbrace{(1,0,0)+(1,0,0)+(1,0,0)}_{\substack{\boldsymbol{l}_1=(1,0,0),s=1,\\ \lambda=3,k_1=3}}.$$

Please note that $p_s(\boldsymbol{\beta},\lambda)$ may be empty. With

$$\mu_\nu^{(\lambda)}(\nu_p) = \frac{4(-1)^{\lambda+1}(\lambda!)\nu^2}{(\nu+\nu_p)^{\lambda+1}},\ \lambda > 0,$$

one obtains

$$D^{\boldsymbol{\beta}}\mu(\boldsymbol{x},\boldsymbol{y}) = \sum_{1\le\lambda\le|\boldsymbol{\beta}|} \frac{4(-1)^{\lambda+1}(\lambda!)\nu(\boldsymbol{x})^2}{(\nu(\boldsymbol{x})+\nu(\boldsymbol{y}))^{\lambda+1}} \sum_{s=1}^{\lambda} \sum_{p_s(\boldsymbol{\beta},\lambda)} (\boldsymbol{\beta}!) \prod_{j=1}^{s} \frac{[D^{\boldsymbol{l}_j}\nu(\boldsymbol{y})]^{k_j}}{(k_j!)[\boldsymbol{l}_j!]^{k_j}}$$

for $|\boldsymbol{\beta}| > 0$. Taylor-expanding $\mu(\boldsymbol{x}, \boldsymbol{x}_p)$ around $\boldsymbol{x}$ and computing the remainder term according to Taylor's theorem, we can thus write

$$\mu(\boldsymbol{x}, \boldsymbol{x}_p) = \sum_{|\boldsymbol{\beta}|=0}^{n} \frac{(\boldsymbol{x}_p - \boldsymbol{x})^{\boldsymbol{\beta}}}{\boldsymbol{\beta}!} D^{\boldsymbol{\beta}} \mu(\boldsymbol{x}, \boldsymbol{x}) + \sum_{|\boldsymbol{\beta}|=n+1} \frac{(\boldsymbol{x}_p - \boldsymbol{x})^{\boldsymbol{\beta}}}{\boldsymbol{\beta}!} R^{\mu}_{\boldsymbol{\beta}}(\boldsymbol{x}_p),$$

where

$$R^{\nu}_{\boldsymbol{\beta}}(\boldsymbol{x}_p) = |\boldsymbol{\beta}| \int_0^1 (1-t)^{|\boldsymbol{\beta}|-1} D^{\boldsymbol{\beta}} \mu(\boldsymbol{x}, \boldsymbol{x} + t(\boldsymbol{x}_p - \boldsymbol{x})) \mathrm{d}t.$$

This means that

$$\begin{aligned}
|\epsilon(\boldsymbol{x})| = \Bigg| & \sum_{|\boldsymbol{\alpha}|=1}^{r+1} \sum_{|\boldsymbol{\beta}|=0}^{r+1-|\boldsymbol{\alpha}|} D^{\boldsymbol{\alpha}} f(\boldsymbol{x}) D^{\boldsymbol{\beta}} \mu(\boldsymbol{x}, \boldsymbol{x}) \\
& \times \left[\frac{1}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \sum_p v_p (\boldsymbol{x}_p - \boldsymbol{x})^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) - Y^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \right] \\
& + \sum_{|\boldsymbol{\alpha}|=1}^{r+1} \sum_{|\boldsymbol{\beta}|=r+2-|\boldsymbol{\alpha}|} D^{\boldsymbol{\alpha}} f(\boldsymbol{x}) \\
& \times \frac{1}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \sum_p v_p R^{\mu}_{\boldsymbol{\beta}}(\boldsymbol{x}_p) (\boldsymbol{x}_p - \boldsymbol{x})^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \\
& + \sum_{|\boldsymbol{\alpha}|=r+2} \frac{1}{\boldsymbol{\alpha}!} \sum_p v_p R^{f}_{\boldsymbol{\alpha}}(\boldsymbol{x}_p) \mu(\boldsymbol{x}, \boldsymbol{x}_p) (\boldsymbol{x}_p - \boldsymbol{x})^{\boldsymbol{\alpha}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \Bigg| \\
\leq & \sum_{s=-1}^{0} \varepsilon^s \|f\|_{s+2,\infty,B_{r_c}[\boldsymbol{x}_p]} \|\nu\|_{s+1,\infty,B_{r_c}[\boldsymbol{x}_p]} \sum_{|\boldsymbol{\alpha}|=1}^{s+2} \sum_{|\boldsymbol{\beta}|=s+2-|\boldsymbol{\alpha}|} \\
& \times \left| \frac{\varepsilon^2}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \sum_p v_p \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) - Y^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \right|
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{s=1}^{r-1} \varepsilon^s \|f\|_{s+2,\infty,B_{r_c}[\boldsymbol{x}_p]} \|\mu\|_{s+1,\infty,B_{r_c}[\boldsymbol{x}_p]} \sum_{|\boldsymbol{\alpha}|=1}^{s+2} \sum_{|\boldsymbol{\beta}|=s+2-|\boldsymbol{\alpha}|} \\
&\times \left| \frac{\varepsilon^2}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \sum_p v_p \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \right| \\
&+ \varepsilon^r \|f\|_{r+2,\infty,B_{r_c}[\boldsymbol{x}_p]} \|\mu\|_{r+1,\infty,B_{r_c}[\boldsymbol{x}_p]} \sum_{|\boldsymbol{\alpha}|=1}^{r+2} \sum_{|\boldsymbol{\beta}|=r+2-|\boldsymbol{\alpha}|} \\
&\times \frac{\varepsilon^2}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \sum_p \left| v_p \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \right|, \qquad \text{(C.7)}
\end{aligned}
$$

where $R^f_{\boldsymbol{\alpha}}(\boldsymbol{x}_p) = |\boldsymbol{\alpha}| \int_0^1 (1-t)^{|\boldsymbol{\alpha}|-1} D^{\boldsymbol{\alpha}} f(\boldsymbol{x}+t(\boldsymbol{x}_p-\boldsymbol{x}))\mathrm{d}t$, $Y^{\boldsymbol{\gamma}} = 1$ if $\boldsymbol{\gamma} = 2\boldsymbol{e}_i$, and $Y^{\boldsymbol{\gamma}} = 0$ else. We are thus able to define an upper bound on the overall error $\epsilon(\boldsymbol{x})$ as

$$
\begin{aligned}
|\epsilon(\boldsymbol{x})| \leq \sum_{s=-2}^{0} \frac{h^s}{A_s(\boldsymbol{x})} \|f\|_{s+2,\infty,B_{r_c}[\boldsymbol{x}]} \|\nu\|_{s+1,\infty,B_{r_c}[\boldsymbol{x}]} \\
+ \sum_{s=1}^{2} \frac{h^s}{A^*_s(\boldsymbol{x})} \|f\|_{s+2,\infty,B_{r_c}[\boldsymbol{x}]} \|\mu\|_{s+1,\infty,B_{r_c}[\boldsymbol{x}]},
\end{aligned}
$$

where

$$
\begin{aligned}
&A_s(\boldsymbol{x}) = \infty,\ s = -2, \\
&A_s(\boldsymbol{x}) = \Bigg[\sum_{|\boldsymbol{\alpha}|=1}^{s+2} \sum_{|\boldsymbol{\beta}|=s+2-|\boldsymbol{\alpha}|} \frac{\varepsilon^2}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \\
&\qquad \times \left| \sum_p v_p \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) - Y^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \right| \Bigg]^{-1},\ -1 \leq s \leq 1,
\end{aligned}
$$

$$A_s^*(\boldsymbol{x}) = \left[\sum_{|\boldsymbol{\alpha}|=1}^{4} \sum_{|\boldsymbol{\beta}|=4-|\boldsymbol{\alpha}|} \frac{\varepsilon^2}{\boldsymbol{\alpha}!\boldsymbol{\beta}!} \times \sum_p \left| v_p \left(\frac{\boldsymbol{x}_p - \boldsymbol{x}}{\varepsilon} \right)^{\boldsymbol{\alpha}+\boldsymbol{\beta}} \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p) \right| \right]^{-1}, \; s = 2.$$

The stability measure S is obtained by substituting the dispersion relation

$$\omega_{\mathrm{mod}}(h\boldsymbol{k}) = 2\hat{\imath}\nu \sum_p v_p \left[\exp\left(\hat{\imath} h\boldsymbol{k} \cdot \frac{\boldsymbol{x}_p - \boldsymbol{x}}{c\varepsilon} \right) - 1 \right] \eta_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_p)$$

into equation (C.2).

Bibliography

[1] C. R. Anderson and C. Greengard, editors. *Vortex dynamics and vortex methods*, volume 28 of *Lectures in Applied Mathematics*, chapter A Lagrangian Vortex Method for the Incompressible Navier-Stokes Equations by G. Russo, pages 585–596. American Mathematical Society, 1991.

[2] D. A. Anderson and M. M. Rai. The use of solution adaptive grids in solving partial differential equations. *Appl. Math. Comput.*, 10: 317–338, 1982.

[3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).

[4] O. Awile, O. Demirel, and I. F. Sbalzarini. Toward an object-oriented core of the PPM library. In *Proc. ICNAAM, Numerical Analysis and Applied Mathematics, International Conference*, pages 1313–1316. AIP, 2010.

[5] J. T. Beale and A. Majda. Rates of convergence for visvous splitting of the Navier-Stokes equations. *Math. Comput.*, 37(156):243–259, 1981.

[6] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *Int. J. Numer. Meth. Eng.*, 37:229–256, 1994.

[7] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: An overview and recent developments. *Comput. Method. Appl. Mech. Engrg.*, 139:3–47, 1996.

[8] M. Bergdorf and P. Koumoutsakos. A Lagrangian particle-wavelet method. *Multiscale Model. Simul.*, 5(3):980–995, 2006.

[9] M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. Multilevel adaptive particle methods for convection-diffusion equations. *Multiscale Model. Simul.*, 4(1):328–357, 2005.

[10] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *Int. J. Numer. Meth. Engng.*, 47:1189–1214, 2000.

[11] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Courier Dover Publications, second revised edition, 2001.

[12] C. J. Budd, W. Huang, and R. D. Russell. Adaptivity with moving grids. *Acta Numerica*, pages 1–131, 2009.

[13] P. Chatelain, A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, and P. Koumoutsakos. Billion vortex particle direct numerical simulations of aircraft wakes. *Comput. Method. Appl. Mech. Engrg.*, 197: 1296–1304, 2008.

[14] J.-S. Chen, W. Han, Y. You, and X. Meng. A reproducing kernel method with nodal interpolation property. *Int. J. Numer. Meth. Engng.*, 56:935–960, 2003.

[15] J. P. Choquin and S. Huberson. Particles simulation of viscous flow. *Computers & Fluids*, 17(2):397–410, 1989.

[16] J. P. Choquin and B. Lucquin-Desreux. Accuracy of a deterministic particle method for Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, 8:1439–1458, 1988.

[17] A. J. Chorin. Numerical study of slightly viscous flow. *J. Fluid Mech.*, 57(4):785–796, 1973.

[18] P. W. Cleary and J. J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *J. Comp. Phys.*, 148:227–264, 1999.

[19] G. M. Constantine and T. H. Savits. A multivariate Faá di Bruno formula with applications. *Trans. Amer. Math. Soc.*, 348(2):503–520, 1996.

[20] R. Cortez. Convergence of high-order deterministic particle methods for the convection-diffusion equation. *Commun. Pure Appl. Math.*, L: 1235–1260, 1997.

[21] G. H. Cottet. A particle-grid superposition method for the Navier-Stokes equations. *J. Comput. Phys.*, 89:301–318, 1990.

[22] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods – Theory and Practice*. Cambridge University Press, New York, 2000.

[23] G.-H. Cottet and S. Mas-Gallic. Une méthode de décomposition pour une équation de type convection-diffusion combinant résolution explicite et méthode particulaire (a splitting algorithm for a convection-diffusion equation based on combining a particle method with an explicit solution). *C. R. Acad. Sci. I-Math.*, 297(2):133–136, 1983.

[24] G. H. Cottet and S. Mas-Gallic. A particle method to solve the navier-stokes system. *Numer. Math.*, 57:805–827, 1990.

[25] G.-H. Cottet, P. Koumoutsakos, and M. L. Ould Salihi. Vortex methods with spatially varying cores. *J. Comput. Phys.*, 162:164–185, 2000.

[26] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. Part 2: The anisotropic case. *Math. Comput.*, 53(188):509–525, 1989.

[27] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. Part 1: The case of an isotropic viscosity. *Math. Comput.*, 53(188):485–507, 1989.

[28] P. Degond and F.-J. Mustieles. A deterministic approximation of diffusion equations using particles. *SIAM J. Sci. Stat. Comput.*, 11 (2):293–310, 1990.

[29] L. Demkowicz, A. Karafiat, and T. Liszka. On some convergence results for FDM with irregular mesh. *Comput. Method. Appl. Mech. Engrg.*, 42:343–355, 1984.

[30] P. du Toit, K. Grubits, J. Marsden, and S. Costiner. Fast generation of potentials for self assembly of particles. Technical report, Control and Dynamical Systems, California Institute of Technology, Pasadena, 2009.

[31] J. D. Eldredge, A. Leonard, and T. Colonius. A general deterministic treatment of derivatives in particle methods. *J. Comput. Phys.*, 180: 686–709, 2002.

[32] G. E. Fasshauer. *Meshfree approximation methods with MATLAB.* World Scientific, 2007.

[33] D. Fishelov. A new vortex scheme for viscous flows. *J. Comput. Phys.*, 86:211–224, 1990.

[34] A. L. Fogelson and R. H. Dillon. Optimal smoothing in function-transport particle methods for diffusion problems. *J. Comp. Phys.*, 109:155–163, 1993.

[35] T.-P. Fries and H.-G. Matthies. Classification and overview of meshfree methods. Informatikbericht 2003-03, Technical University Braunschweig, 2004.

[36] M. J. Fritts, W. P. Crowley, and H. Trease, editors. *The Free-Lagrange method; Proceedings of the First International Conference, Hilton Head Island, SC, March 4-6, 1985*, volume 238 of *Lecture Notes in Physics*, 1985.

[37] T. Gasser, H.-G. Müller, and V. Mammitzsch. Kernels for nonparametric curve estimation. *J. R. Statist. Soc. B*, 47(2):238–252, 1985.

[38] A. Gharakhani. A higher order vorticity redistribution method for 3-D diffusion in free space. Technical report, Sandia National Laboratories, 2000.

[39] C. Golia, B. Buonomo, and A. Viviani. A corrected vortex blob method for 3D thermal buoyant flows. *Energ. Convers. Manage.*, 49:3243–3252, 2008.

[40] J. Goodman. Convergence of the random vortex method. *Commun. Pure Appl. Math.*, XL:189–220, 1987.

[41] C. Greengard. The core spreading vortex method approximates the wrong equation. *J. Comput. Phys.*, 61(2):345 – 348, 1985.

[42] M. Hardy. Combinatorics of partial derivatives. *Electron. J. Combin.*, 13(#R1):1–13, 2006.

[43] S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. *J. Comput. Phys.*, 210:342–367, 2005.

[44] T. Y. Hou. Convergence of a variable blob vortex method for the Euler and Navier-Stokes equations. *SIAM J. Numer. Anal.*, 27(6): 1387–1404, 1990.

[45] A. Iske and M. Käser. Two-phase flow simulation by AMMoC, an adaptive meshfree method of characteristics. *Comput. Model. Eng. & Sci.*, 7(2):133–148, 2005.

[46] G. R. Johnson and S. R. Beissel. Normalized smoothing functions for SPH impact computations. *Int. J. Numer. Meth. Engng.*, 39:2725–2741, 1996.

[47] P. Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex. *J. Comput. Phys.*, 138:821–857, 1997.

[48] P. Koumoutsakos. Multiscale flow simulations using particles. *Annu. Rev. Fluid Mech.*, 37:457–487, 2005.

[49] P. Koumoutsakos, A. Leonard, and F. Pépin. Boundary conditions for viscous vortex methods. *J. Comput. Phys.*, 113:52–61, 1994.

[50] K. Kuwahara and H. Takami. Numerical studies of two-dimensional vortex motion by a system of point vortices. *J. Phys. Soc. Jap.*, 34 (1):247–253, 1973.

[51] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Math. Comput.*, 37(155):141–158, 1981.

[52] N. Lanson and J.-P. Vila. Convergence des méthodes particulaires renormalisées pour les systèmes de Friedrichs. *C. R. Acad. Sci. Paris I*, 340:465–470, 2005.

[53] N. Lanson and J.-P. Vila. Renormalized meshfree schemes I: Consistency, stability, and hybrid methods for conservation laws. *SIAM J. Numer. Anal.*, 46(4):1912–1934, 2008.

[54] N. Lanson and J.-P. Vila. Renormalized meshfree schemes II: Convergence for scalar conservation laws. *SIAM J. Numer. Anal.*, 46(4): 1935–1964, 2008.

[55] A. Leonard. Vortex methods for flow simulations. *J. Comp. Phys.*, 37:289–335, 1980.

[56] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *Int. J. Numer. Meth. Fluids*, 20:1081–1106, 1995.

[57] W. K. Liu, Y. Chen, R. A. Uras, and C. T. Chang. Generalized multiple scale reproducing kernel particle methods. *Comput. Method. Appl. Mech. Engrg.*, 139:91–157, 1996.

[58] D.-G. Long. Convergence of the random vortex method in two dimensions. *J. Am. Math. Soc.*, 1(4):779–804, 1988.

[59] V. Mammitzsch. Optimal kernels. *Statist. & Dec.*, 25:153–172, 2007.

[60] C. Marchioro and Pulvirenti. Hydrodynamics in two dimensions and vortex theory. *Commun. Math. Phys.*, 84:483–503, 1982.

[61] F. A. Marvasti. *Nonuniform sampling – Theory and Practice*. Springer, illustrated edition, 2001.

[62] S. Mas-Gallic. *Contribution à l'Analyse Numérique des Méthodes Particulaires*. PhD thesis, Université de Pierre et Marie Curie, Paris, 1987.

[63] V. Maz'ya and G. Schmidt. *Approximate Approximations*, volume 141 of *Mathematical surveys and monographs*. American Mathematical Society, 2007.

[64] J. J. Monaghan. Extrapolating B splines for interpolation. *J. Comput. Phys.*, 60:253–262, 1985.

[65] J. J. Monaghan. Smoothed particle hydrodynamics. *Rep. Prog. Phys.*, 68:1703–1759, 2005.

[66] Muñoz-Gómez. Adaptive node refinement collocation method for partial differential equations. In *Seventh Mexican International Conference on Computer Science (ENC'06)*, 2006.

[67] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Comput. Mech.*, 10(5):307–318, 1992.

[68] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant. An improved SPH method: Towards higher order convergence. *J. Comput. Phys.*, 225:1472–1492, 2007.

[69] M. Perlmann. On the accuracy of vortex methods. *J. Comp. Phys.*, 59:200–223, 1985.

[70] P. Poncet. Finite difference stencils based on particle strength schemes for improvement of vortex methods. *J. Turbul.*, 7(23):1–24, 2006.

[71] P. W. Randles and L. D. Libersky. Smoothed particle hydrodynamics: Some recent improvements and applications. *Comput. Method. Appl. Mech. Engrg.*, 139:375–408, 1996.

[72] P. A. Raviart. An analysis of particle methods. In F. Brezzi, editor, *Numerical Methods in Fluid Dynamics*, volume 1127 of *Lecture Notes in Mathematics*, pages 244–323. Springer-Verlag, 1985.

[73] S. Reboux, B. Schrader, and I. F. Sbalzarini. Adaptive multiresolution simulations using self-organizing Lagrangian particles. Submitted to J. Comp. Phys., December 2010.

[74] M. Rechtsman, F. Stillinger, and S. Torquato. Designed interaction potentials via inverse methods for self-assembly. *Phys. Rev. E*, 73(1): 011406, 2006.

[75] G. Russo. A deterministic vortex method for the Navier-Stokes equations. *J. Comput. Phys.*, 108:84–94, 1993.

[76] I. F. Sbalzarini, A. Mezzacasa, A. Helenius, and P. Koumoutsakos. Effects of organelle shape on fluorescence recovery after photobleaching. *Biophys. J.*, 89(3):1482–1492, 2005.

[77] I. F. Sbalzarini, J. H. Walther, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos. PPM – a highly efficient parallel particle-mesh library for the simulation of continuum systems. *J. Comput. Phys.*, 215(2):566–588, 2006.

[78] M. Schlegel, O. Knoth, M. Arnold, and R. Wolke. Multirate Runge–Kutta schemes for advection equations. *J. Comput. Appl. Math.*, 226: 345–357, 2009.

[79] B. Schrader, S. Reboux, and I. F. Sbalzarini. Discretization correction of general integral PSE operators in particle methods. *J. Comput. Phys.*, 229:4159–4182, 2010.

[80] S. Shankar and L. van Dommelen. A new diffusion procedure for vortex methods. *J. Comput. Phys.*, 127:88–109, 1996.

[81] O. Shipilova, H. Haario, and A. Smolianski. Particle transport method for convection problems with reaction and diffusion. *Int. J. Numer. Meth. Fluids*, 54:1215–1238, 2007.

[82] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth & Brooks/Cole, Belmont, CA, 1989.

[83] L. Verlet. Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159(1): 98–103, 1967.

[84] Y.-M. Wang, C. Syuan-Mu, and C.-P. Wu. A meshless collocation method based on the differential reproducing kernel interpolation. *Comput. Mech.*, 45:585–606, 2010.

[85] D. Wee and A. F. Ghoniem. Modified interpolation kernels for treating diffusion and remeshing in vortex methods. *J. Comput. Phys.*, 213: 239–263, 2006.

[86] M. Whitesides, George and B. Grzybowski. Self-assembly at all scales. *Science*, 295:2418–2421, 2002.

[87] G. B. Wright and B. Fornberg. Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.*, 212:99–123, 2006.

[88] C.-P. Wu, J.-S. Wang, and Y.-M. Wang. A DRK interpolation-based collocation method for the analysis of functionally graded piezoelectric hollow cylinders under electro-mechanical loads. *Comput. Model. Eng. & Sc.*, 52(1):1–38, 2009.