

**When Euler meets Lagrange
Particle-Mesh Modeling of Advection Dominated Flows**

Maljaars, Jakob

DOI

[10.4233/uuid:a400512d-966d-402a-a40a-fedf60acf22c](https://doi.org/10.4233/uuid:a400512d-966d-402a-a40a-fedf60acf22c)

Publication date

2019

Document Version

Final published version

Citation (APA)

Maljaars, J. (2019). *When Euler meets Lagrange: Particle-Mesh Modeling of Advection Dominated Flows*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:a400512d-966d-402a-a40a-fedf60acf22c>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

When Euler meets Lagrange

PARTICLE-MESH MODELING OF ADVECTION DOMINATED
FLOWS

When Euler meets Lagrange

PARTICLE-MESH MODELING OF ADVECTION DOMINATED
FLOWS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. T.H.J.J van der Hagen,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 2 december 2019 om 15.00 uur

door

Jakob Marinus MALJAARS

civieltechnisch ingenieur,
Technische Universiteit Delft, Nederland,
geboren te Vlissingen, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: prof. dr. ir. W.S.J. Uijtewaal,
copromotor: dr. ir. R.J. Labeur

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. W.S.J. Uijtewaal,	Technische Universiteit Delft
Dr. ir. R.J. Labeur,	Technische Universiteit Delft

Onafhankelijke leden:

Prof. dr. ir. B. Koren,	Technische Universiteit Eindhoven
Prof. dr. ir. L.J. Sluys,	Technische Universiteit Delft
Prof. dr. ir. C. Vuik,	Technische Universiteit Delft
Prof. dr. G.N. Wells,	University of Cambridge

Overige leden:

Dr. M. Möller,	Technische Universiteit Delft
----------------	-------------------------------



Research School for Fluid Mechanics

Keywords: Lagrangian-Eulerian, finite elements, hybridized discontinuous Galerkin, particle-in-cell, PDE-constrained optimization, conservation, advection dominated flows, advection-diffusion equation, incompressible Navier-Stokes, multiphase flows

Printed by: Ridderprint BV, www.ridderprint.nl

Copyright © 2019 by J.M. Maljaars

ISBN 978-94-6375-581-8

An electronic version of this dissertation is available at
<http://repository.tudelft.nl/>.

*All the rivers run into the sea; yet the sea is not full;
unto the place from whence the rivers come, thither they return again.*

Ecclesiastes 1:7

Contents

Summary	v
Samenvatting	vii
1 Introduction	1
1.1 Background	1
1.2 Numerical modeling	2
1.2.1 Reference frames	2
1.2.2 Conservation laws.	3
1.2.3 Dimensionless numbers	5
1.2.4 Eulerian and Lagrangian numerical approaches.	7
1.2.5 Particle-mesh methods	9
1.3 Present research	11
1.3.1 Objectives and scope	11
1.3.2 Practical Relevance.	12
1.3.3 Outline	13
2 Particles and finite elements	15
2.1 Particle-mesh modeling.	15
2.1.1 Operator splitting	16
2.1.2 From particle space to mesh space and back	17
2.2 Mesh step: the Galerkin finite element method.	19
2.2.1 Function spaces and mesh partitioning	20
2.2.2 Continuous Galerkin method.	22
2.2.3 Discontinuous Galerkin method	22
2.2.4 Hybridized Discontinuous Galerkin method.	23
2.2.5 Continuous, discontinuous or hybrid?	24
2.2.6 FEM and variational formulations.	25
2.3 Particle-mesh interactions	26
2.3.1 A variational framework for the projections.	26
2.3.2 Consistency of projections	28
2.4 Example: 1D diffusion of Gaussian	29
2.5 Conclusion	32
3 Scalar transport	33
3.1 Model problem	33
3.1.1 Governing equations	34
3.1.2 Operator splitting	34
3.1.3 Auxiliary definitions	35
3.1.4 Solution strategy and outline	36

3.2	Semi-discrete formulations for the particle- and mesh sub-problems. . .	37
3.2.1	Lagrangian discretization of the advection problem	37
3.2.2	Eulerian discretization of the diffusion problem.	37
3.3	Semi-discrete formulation of the projection steps	38
3.3.1	Particle-mesh projection via ℓ^2 projections	38
3.3.2	PDE-constrained ℓ^2 projection.	39
3.3.3	Mesh-particle projection	41
3.4	Properties of the semi-discrete formulations.	41
3.4.1	Consistency	42
3.4.2	Conservation	42
3.5	A fully-discrete formulation	43
3.5.1	Particle advection.	44
3.5.2	Particle-mesh projection I: ℓ^2 projection	44
3.5.3	Particle-mesh projection II: PDE-constrained projection. . . .	44
3.5.4	Diffusion equation	46
3.5.5	Mesh-particle projection	47
3.6	Algorithmic aspects.	47
3.6.1	Conservation and consistency	47
3.6.2	Choosing the Lagrange multiplier space.	48
3.6.3	Static condensation.	49
3.6.4	Time accuracy	50
3.7	Extension to moving meshes	50
3.7.1	Conservation	53
3.7.2	Discrete Geometric Conservation Law.	53
3.8	Numerical examples	54
3.8.1	Advection-diffusion: Gaussian hat	55
3.8.2	Advection: rigid body rotation.	57
3.8.3	Advection: advection skew to mesh	60
3.8.4	Advection on a moving mesh.	61
3.9	Conclusion	64
4	Incompressible flow problems	67
4.1	Model problem	67
4.1.1	Governing equations	68
4.1.2	Operator-splitting	68
4.1.3	Auxiliary definitions	69
4.2	Semi-discrete formulations	71
4.2.1	Lagrangian discretization of the particle advection	71
4.2.2	Particle-mesh projection	71
4.2.3	Eulerian discretization of the Stokes problem.	73
4.2.4	Mesh-particle projection	75
4.2.5	Properties of the semi-discrete formulation	75

4.3	Fully-discrete formulations	77
4.3.1	Particle advection.	77
4.3.2	Particle-mesh projection	77
4.3.3	Stokes problem	78
4.3.4	Mesh-particle projection	79
4.3.5	Algorithmic aspects.	79
4.4	Numerical examples	81
4.4.1	Poiseuille flow	81
4.4.2	Taylor-Green flow.	84
4.4.3	Backward facing step.	92
4.4.4	Flow past a circular cylinder	94
4.5	Conclusion	96
5	Application to multiphase flows	99
5.1	Model problem	101
5.1.1	Operator splitting	101
5.1.2	Auxiliary definitions	103
5.2	Model formulation	104
5.2.1	Lagrangian discretization of advection problems	105
5.2.2	PDE-constrained density map	105
5.2.3	PDE-constrained momentum map.	107
5.2.4	Multiphase Stokes solver	108
5.3	Model properties	110
5.3.1	Conservation	110
5.3.2	Hydrostatic equilibrium balance	111
5.4	Implementation considerations.	113
5.4.1	ℓ^2 -projection or conservative PDE projection?	113
5.4.2	Density gradient penalization	114
5.4.3	No-flux boundary condition	114
5.4.4	Particle insertion and deletion	115
5.5	Numerical examples	115
5.5.1	Rising bubble	116
5.5.2	Lock exchange	119
5.5.3	Hydrostatic balance	122
5.5.4	Standing wave	124
5.5.5	Wave train	126
5.5.6	Dambreak.	128
5.5.7	Breaking solitary wave	132
5.6	Conclusion	138
6	Conclusion and outlook	139
6.1	Conclusions	139
6.2	Outlook and perspectives	142

A	The local ℓ^2-projections and moving least squares	145
B	Time accuracy considerations	149
C	Surface tension force	151
D	Computer implementation	153
D.1	FEniCS	153
D.1.1	Static condensation	154
D.2	Particle administration	156
D.2.1	The particle class	156
D.2.2	Particle advection	157
D.2.3	Particle tracking	158
D.2.4	Boundary conditions at particle level	160
D.3	Particle-mesh projections	163
D.4	Performance tests	163
D.4.1	Rotation of Gaussian pulse	163
D.4.2	Lock exchange	165
D.4.3	Three-dimensional Taylor-Green	167
	Bibliography	169
	Acknowledgements	179
	Curriculum Vitæ	181
	List of Publications	183

Summary

This thesis presents a numerical framework for simulating advection-dominated flows which reconciles the advantages of Eulerian mesh-based schemes with those of a Lagrangian particle-based discretization strategy. Particularly, the strategy proposed in this thesis inherits the diffusion-free properties as in Lagrangian particle-based advection, while simultaneously possessing high-order accuracy and local conservation properties as in state-of-the-art Eulerian mesh-based discretization strategies. These properties render the scheme particularly apt for simulating flow- and transport processes in which the physical diffusion is low, such as turbulent flows, or simulating flow problems with sharp and complex-shaped interfaces, such as the air-water interface in breaking ocean waves.

The general approach in the developed strategy is to use Lagrangian particles for the advection part of the flow problem, whereas an Eulerian mesh-based approach is used for an efficient, high-order accurate, and conservative discretization of the constitutive relations. The Lagrangian particles and the Eulerian mesh exchange information via two particle-mesh projections. Given this sequence of steps, the particle-mesh strategy is interpreted in Chapter 2 as an operator-splitting approach in which the particle-mesh projections are required to project information from the Lagrangian particle space to the mesh space and *vice versa*.

Adopting this operator splitting perspective on the developed particle-mesh scheme is essential to understand the role played by the projections. Since the physical processes are conducted at either the particle level (advection) or the mesh level (constitutive modeling), the particle-mesh projections are auxiliary steps which should seamlessly bridge the gap between the particle space and the mesh space and *vice versa*. This notion leads to a number of requirements for the particle-mesh projections: (i) they should not obstruct higher-order accuracy [*accuracy*]; (ii) the combination of the mutual particle-mesh projections should not introduce 'spurious physics' [*consistency*]; (iii) they should comply with the discrete conservation properties of the Eulerian mesh-based solver [*conservation*]; (iv) the computational costs associated with the particle-mesh projections should stay as low as possible [*efficiency*]. Developing particle-mesh projections satisfying these requirements is the central challenge in this research.

A generic variational framework forms the starting point to solve this challenge. This naturally leads to a finite element (FEM) based discretization for the equations at the mesh, in conjunction with an interpretation of the particle-mesh projection as the projection of a particle field onto a finite element basis, and the inverse route for the mesh-particle projection. These projections are casted in terms of ℓ^2 optimization problems. Crucial is that the particles in this approach serve as *sampling points* of the continuum, rather than that they carry integral quantities such as mass and volume. For a simplistic example in which advection is absent, this variational particle-mesh strategy is able to meet all the above requirements, provided that a (hybridized) Dis-

continuous Galerkin ((H)DG) method is adopted for the FEM discretization. The latter is prerequisite for solving the particle-mesh projection in an efficient, cellwise manner.

Extending the particle-mesh operator splitting strategy to advection-dominated problems, while maintaining discrete conservation properties requires a novel formulation for the particle-mesh projection. Central concept is to extend the ℓ^2 objective function for the particle-mesh projection with a properly defined constraint. The key idea in formulating the constraint is that from a mesh-perspective the particle motion must satisfy an advection operator. By expressing the control variable in terms of single-valued functions at cell interfaces, the HDG method naturally provides the necessary ingredients to formulate the optimality system. The resulting ‘PDE-constrained particle-mesh projection’ is provably consistent and possesses (local) conservation properties. Furthermore, it can be reformulated on moving meshes in a straightforward manner. A range of numerical experiments in the context of the linear advection-diffusion equation demonstrates standard convergence rates in space and second-order convergence rates in time. Most notably, the proposed hybrid particle-mesh strategy eliminates numerical diffusion in pure advection problems, without compromising accuracy and local mass conservation properties.

Applying the particle-mesh strategy to the incompressible Navier-Stokes equations, avoids stabilization of the linearized advection term, since the advection of specific momentum is done on the Lagrangian particles. An unsteady Stokes problem is solved at the Eulerian mesh to account for viscous diffusion and to incorporate the incompressibility constraint. By employing a HDG discretization to fit this purpose, the mesh solver conserves (specific) momentum locally, and can provide pointwise divergence-free velocity fields. The former feature, in combination with a PDE-constrained approach for the projection of the specific momentum from the particles to the mesh, ensures overall momentum conservation of the particle-mesh scheme. The ability to advect the particles in velocity fields that are inherently pointwise divergence-free is crucial for maintaining a uniform particle distribution, which is evident by comparing against velocity fields that are only local and/or global solenoidal. Even for challenging benchmarks in which stagnation points and separation points are present, such as the flow over a backward facing step and the flow around a circular cylinder, *ad hoc* particle shifting measures are avoided as long as the particles are advected in pointwise solenoidal velocity fields. Moreover, the particle-mesh approach is robust for high Reynolds number flows, characterized by the presence of sharp shear layers.

For incompressible, immiscible multiphase flow problems, density fields involving sharp jumps are tracked in a diffusion-free manner using Lagrangian particles. Exploiting the developed particle-mesh tools, including the PDE-constrained particle-mesh projection, allows a diffusion-free and mass conservative tracking of the density fields at the underlying mesh. The potential and the challenges of such an approach are investigated for a series of numerical experiments, ranging from problems in which the density ratio is relatively close to unity, to problems in which the density ratio amounts to a factor 1000. As just one of the potential applications, various laboratory wave flume experiments were accurately reproduced numerically using the developed particle-mesh scheme.

Interested readers are encouraged to use the developed numerical tools, which are collected in the open source software package `LEoPart`.

Samenvatting

Deze thesis presenteert een numeriek raamwerk voor het simuleren van advection-gedomineerde transportprocessen waarin de voordelen van Euleriaanse, grid-gebaseerde methodes en Lagrangiaanse, deeltjes-gebaseerde methodes worden gecombineerd. Deze combinatie resulteert in een methode die vrij is van kunstmatige diffusie als in Lagrangiaans transport, terwijl tegelijkertijd hoge nauwkeurigheid en lokale behoudseigenschappen als in moderne Euleriaanse discretisatie schema's beschikbaar blijven. Deze eigenschappen maken de methode een aantrekkelijk alternatief op bestaande numerieke methodes voor het simuleren van stromings- en transportproblemen waarin de fysische diffusie laag is, zoals turbulente stromingen, of het simuleren van stromingsproblemen waarin een scherpe scheiding tussen verschillende fases voorkomt, zoals brekende golven waarin het wateroppervlak lucht en water scheidt.

Centraal in de aanpak is het inzetten van Lagrangiaanse deeltjes voor het advectieve deel van het transportprobleem, terwijl een Euleriaanse, grid-gebaseerde methode gebruikt wordt voor het efficiënt, nauwkeurig, en massa- en momentumbehoudend implementeren van de constitutieve relaties. De Lagrangiaanse deeltjes en het Euleriaanse grid wisselen informatie uit via twee zogenaamde 'deeltjes-grid projecties'. Deze volgorde van stappen leidt tot een interpretatie van de ontwikkelde methode als een splitsen van wiskundige operatoren, waarin de deeltjes-grid projecties vereist zijn om de Lagrangiaanse representatie te projecteren op een functieruimte die gedefinieerd is op het Euleriaanse grid, en *vice versa*.

Dit perspectief op de ontwikkelde methode is cruciaal om de verschillende projectiestappen te interpreteren. Doordat de fysische processen op de deeltjes, dan wel op het niveau van het grid berekend worden, zijn de projecties hulpstappen die de kloof tussen de deeltjes- en de grid-representatie naadloos dienen te overbruggen. Deze notie leidt tot een aantal eisen aan de deeltjes-grid projecties: (i) deze stappen mogen geen belemmering vormen voor het bereiken van hoge nauwkeurigheid [*nauwkeurigheid*]; (ii) ze mogen geen eigen, artificiële fysica introduceren [*consistentie*]; (iii) ze moeten voldoen aan de discrete behoudseigenschappen van de Euleriaanse oplossingsstrategie [*behoud*]; (iv) de vereiste rekentijd voor deze projecties dient zo laag mogelijk te blijven [*efficiëntie*]. Het ontwikkelen van deeltjes-grid projecties die voldoen aan deze eigenschappen, vormt het centrale thema van deze thesis.

Een generiek raamwerk, gebaseerd op variatierekening, vormt het startpunt om deze uitdaging op te lossen. Dit raamwerk leidt logischerwijs tot een eindige elementen (EE) discretisatie voor de vergelijkingen die op het grid opgelost dienen te worden, alsmede tot een interpretatie van de deeltjes-grid interactie als de projectie van een deeltjesveld op een eindige elementen functieruimte. De omgekeerde weg wordt bewandeld voor de grid-deeltjes projectie. Beide projecties worden geformuleerd in termen van een ℓ^2 -minimalisatie probleem. Cruciaal hiervoor is dat de deeltjes in deze aanpak dienen als puntevaluaties van het onderliggende continuüm, in plaats dat de deeltjes daadwerkelijk

een massa en/of een volume toegewezen wordt.

Voor een simplistisch voorbeeld, waarin advection vooralsnog afwezig is, voldoet deze combinatie van deeltjes-grid projecties aan alle gestelde eisen, mits een (hybride) Discontinue Galerkin ((H)DG) methode gebruikt wordt voor de EE discretisaties. Dit laatste is een vereiste om de deeltjes-grid projectie efficiënt op te lossen. Om deze strategie toe te passen op advection-ge domineerde stromingen, zodanig dat discrete behoudseigenschappen gegarandeerd blijven, wordt de ℓ^2 -doelfunctie in de deeltjes-grid projectie onderworpen aan een aanvullende nevenvoorwaarde. De sleutel in het formuleren van deze voorwaarde is dat de deeltjesbeweging moet voldoen aan een advection vergelijking, geredeneerd vanuit het perspectief van het mesh. Door de controlevariabele uit te drukken in termen van een daarvoor bestemde functie op de celranden van het grid, reikt een HDG-methode alle benodigde gereedschappen aan om de optimalisatie vergelijkingen te formuleren. De resulterende '*PDE-constrained particle-mesh projection*' is aantoonbaar consistent en bezit (lokale) behoudseigenschappen. Daarnaast kan de aanpak eenvoudig herformuleerd worden om toegepast te worden op een bewegend grid.

Convergentie in de ruimte en tweede-orde nauwkeurigheid in de tijd worden geïllustreerd middels een reeks numerieke experimenten voor de lineaire advection-diffusie vergelijking. In het bijzonder tonen de resultaten voor pure advection problemen aan dat kunstmatige diffusie geëlimineerd wordt in de voorgestelde deeltjes-grid strategie, zonder daarbij afbreuk te doen aan nauwkeurigheid en behoudseigenschappen.

Toepassen van de deeltjes-mesh methode op de onsamendrukbare Navier-Stokes vergelijkingen maakt stabilisatie van de gelineariseerde advection term overbodig. Om dit te bereiken, wordt een instationair Stokes probleem opgelost op het Euleriaanse grid om de viskeuze effecten en de divergentievrijheid te verdisconteren. Door gebruik te maken van een HDG discretisatie is deze stap lokaal momentumbehoudend, en worden snelheidsvelden verkregen die puntsgewijs divergentievrij zijn. De eerste eigenschap, in combinatie met een deeltjes-mesh projectie in termen van de *PDE-constrained particle-mesh projection* voor de projectie van het specifieke momentum tussen grid en deeltjes, garandeert dat momentum behouden blijft. Advecteren van deeltjes in een snelheidsveld dat puntsgewijs divergentievrij is, is cruciaal voor het in stand houden van een uniforme deeltjes verdeling. Deze eigenschap is evident wanneer resultaten vergeleken worden met de resulterende deeltjesverdeling bij gebruik van lokaal en globaal divergentievrije snelheidsvelden. Zelfs voor tests waarin een stagnatiepunt of loslatingspunten inherent zijn, zoals de stroming over een drempel en de stroming rond een cylinder, blijft de deeltjesverdeling uniform mits de deeltjes in een puntsgewijs divergentievrij snelheidsveld getransporteerd worden. *Ad hoc* deeltjesverplaatsing is hierdoor overbodig. Bovendien laten resultaten zien dat de deeltjes-mesh methode robuust is voor stromingen die gekarakteriseerd worden door een hoog Reynolds getal.

In de context van onsamendrukbare, niet-mengbare multifase stromingen, kunnen dichtheidsvelden met scherpe sprongen gevolgd worden op de Lagrangiaanse deeltjes zonder kunstmatige diffusie toe te voegen. De ontwikkelde deeltjes-mesh gereedschappen, waaronder de projectie met PDE-nevenvoorwaarde, maken het mogelijk om dichtheidsvelden op het onderliggende mesh op een diffusie-vrije en massabehoudende manier te volgen. De potentie en uitdagingen behorend bij een dergelijke aanpak zijn onderzocht voor een reeks numerieke experimenten, uiteenlopend van problemen waarin de

dichtheidsverhouding van de verschillende fases relatief dicht bij 1 ligt, tot experimenten waarbij de dichtheidsverhoudingen oplopen tot een factor 1000. Als slechts één van de potentiële toepassingen zijn verschillende golfgootexperimenten gesimuleerd. De ontwikkelde methodiek reproduceert deze laboratoriumexperimenten nauwkeurig.

Om het gebruik van de ontwikkelde numerieke gereedschappen te stimuleren, is de bijbehorende computercode voor dit onderzoek vrij beschikbaar onder een open source licentie via LEOPart.

1

Introduction

1.1. Background

The dispersion of a sediment rich river plume, the spreading of an ash cloud emitted from an active volcano, or something as peaceful as the breaking of ocean waves at the beach on a calm summer day. Physical problems involving the flow of liquids or gases or the transport of matter are ubiquitous in our daily lives, in the environment, and in industrial processes. Numerous other examples could therefore be brought up, covering situations as ordinary as the stirring of coffee in a cup, and as exotic as the drifting of My Little Ponies in the ocean¹.

Given this widespread presence of flow and transport phenomena, it should come as no surprise that understanding and predicting them can be relevant, important, or simply critical. For example, accurate forecasts of the spreading of a volcanic ash cloud in the atmosphere are indispensable to ensure aviation safety. Also in civil engineering hydraulics, flow and transport problems are manifest to various practical situations. Understanding the mixing of salt and fresh water is for instance important to optimize the design of sea locks, and gaining insight in the forces exerted by breaking waves is indispensable for a proper design of coastal structures.

Over the past decades numerical techniques and computing power have made huge progress, allowing to simulate increasingly complex problems using computers. In the context of transport phenomena, a computer aided simulation strategy entails the promise of complementing or substituting laboratory experiments, particularly so when the relevant time and length scales render experiments arduous and expensive. Overcoming the limitation of a restricted amount of measurement devices as is typical to laboratory experiments, in combination with the repeatability of computer simulations at low additional costs further bolster these developments.

¹At the time of this writing, one of the world's largest containerships lost approximately 300 of its containers in a storm just north of the Netherlands. At least one of the containers was filled with thousands of little plastic ponies, which still might be drifting in the world's oceans <https://www.delta.tudelft.nl/article/oceanographers-wake-my-little-ponies>

To this end, novel numerical techniques are continuously being devised for approximately solving the conservation laws that describe flow and transport processes at a continuum scale. The ongoing effort to simulate these problems more efficiently, more detailed, or on more complex geometries, illustrates that the perfect modeling approach has not yet been formulated. Indeed, it is a non-trivial task to formulate efficient numerical schemes which simultaneously stay as close as possible to the underlying physical phenomena, without introducing artifacts. This is even further compounded when the problem of interest demands for geometrically complex domains. In particular, with the physical dissipation typically being small for many transport problems, over-dissipation and anti-dissipation should be avoided in numerical models as this would obscure the true physics otherwise. The more so because sharp transition regions pertain to many transport problems. These sharp transition regions include, among others, sharp shear layers in turbulent flows, or interfaces of complex shapes in multiphase flows of which the air-water interface in breaking waves is an illustrative example. Numerically simulating these advection-dominated transport problems is therefore often like balancing on the edge of a knife.

This research develops a new modeling approach for simulating transport phenomena which is particularly suited for simulating flows that are characterized by low physical dissipation and sharp transition layers. This is achieved by combining concepts from two existing frameworks, the Eulerian and Lagrangian framework. With these frameworks named after the 18th century mathematicians Euler² and Lagrange³, this thesis essentially aims to organize a contemporary meeting between these giants in mathematics, and aims to join their forces for simulating advection dominated flows. The main challenge in order to achieve this goal, is to seamlessly integrate the two frameworks so as to add-up their advantages, while discarding their disadvantages. The resulting model can be attractive for a range of practical transport problems. In the scope of this research, the specific application in mind is the simulation of multi-phase flows involving large density differences between the phases, and where the separating interface attains complex shapes, such as the simulation of breaking ocean waves on a coastal structure.

1.2. Numerical modeling

1.2.1. Reference frames

At the basis of all continuum models for solving flow and transport phenomena are conservation laws, which essentially describe that in isolated systems, physical quantities cannot be created nor destroyed. In classical mechanics, the conservation laws for transport processes include conservation of mass and conservation of momentum.

Crucial in mathematically formulating these physical laws is the adoption of a frame of reference in which they are described. In one approach, the flow and transport is

²Leonhard Euler (1707 - 1783) was a Swiss mathematician, who laid the foundations for many topics in modern mathematics, and made influential contributions to the study of fluid dynamics of which the so-called 'Euler equations' is probably best-known.

³Joseph-Louis de Lagrange (1736-1813), contemporary of Euler, was another influential 18th century mathematician. He became best-known for the introduction of *Lagrangian mechanics*. This describes the dynamics of mechanical systems in terms of a *stationary action* principle, which requires to track the state of 'particles' composing the mechanical system along their trajectory.

described by using a control volume which is fixed in space and time. In this so-called Eulerian frame of reference, the equations describe the changes of the material properties within the control volume over time, with these changes being induced by the net inflow of mass or momentum through the control volume boundaries and/or sources and sinks for material quantities inside the control volume.

In the so-called Lagrangian frame of reference, on the other hand, an observer moves along with the material volumes at the same velocity. Intuitively, such a path-following approach might be beneficial for tracking sharp and dynamic flow features. The change of the properties of the moving material volumes are considered, rather than the changes over a fixed control volume. The conceptual difference between an Eulerian framework and a Lagrangian framework is schematized in Fig. 1.1, and the practical implications of the chosen reference frame on the formulation of numerical models become apparent when formulating the continuum equations.

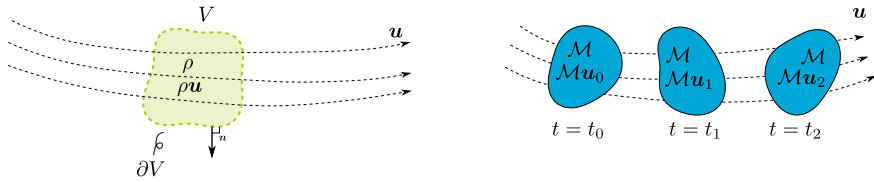


Figure 1.1: Principle sketch of an Eulerian (left) and a Lagrangian (right) frame of reference. For an explanation of the symbols see Section 1.2.2.

1.2.2. Conservation laws

For ease of presentation, the relevant equations for an Eulerian frame of reference, after which the equations for a Lagrangian frame of reference are obtained in a straightforward manner. Consider therefore the Eulerian control volume $V \subset \mathbb{R}^d$ where \mathbb{R}^d denotes the domain of interest with the number spatial dimensions d . The boundary of the control volume V is denoted as ∂V , on which \mathbf{n} is the outward pointing unit normal, Fig. 1.1. Furthermore, $I = (t_0, t_1)$ is the time interval of interest. Ignoring any sources and sinks in the volume V , the mass change over the volume V should balance the net inward transport of mass through the boundary ∂V over the time interval I , i.e.

$$\int_V \Delta \rho \, dV = - \int_I \oint_{\partial V} \rho \mathbf{u} \cdot \mathbf{n} \, d\Gamma dt, \quad (1.1)$$

in which $\Delta \rho$ is the density increment in the time interval I , and \mathbf{u} denotes the transport velocity. With \mathbf{u} and ρ assumed differentiable, and after applying Green's divergence theorem, the *continuity equation* is obtained

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (1.2)$$

in which $\partial \rho / \partial t$ denotes the time derivative of the density and ∇ is the spatial gradient operator. Applying the chain rule for differentiation on the gradient term, gives

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u} = 0. \quad (1.3)$$

To interpret this equation, consider the rate of change in the density as observed by a moving observer at $\mathbf{x} = \mathbf{x}(t)$. Using the chain rule the material or *Lagrangian* derivative for the density ρ is obtained, i.e.:

$$\frac{D\rho}{Dt} := \frac{\partial\rho}{\partial t} + \frac{d\mathbf{x}}{dt} \cdot \frac{\partial\rho}{\partial\mathbf{x}}, \quad (1.4)$$

in which $\partial\rho/\partial t$ represents the change in density at fixed position \mathbf{x} . The term $(d\mathbf{x}/dt) \cdot (\partial\rho/\partial\mathbf{x})$, represents the change due to the fact that an observer moves into a region with a possibly different density. Setting $d\mathbf{x}/dt = \mathbf{u}$, i.e. the velocity of the observer equals the flow velocity, and substituting into Eq.(1.3), the continuity equation is rewritten as

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{u} = 0. \quad (1.5)$$

This equation leads to some important results. Firstly, if the density variations of a fluid parcel are negligible compared to a reference density (Boussinesq approximation), temporal and spatial changes in the density field vanish, and the continuity equation Eq. (1.5) reduces to the *incompressibility constraint*

$$\nabla \cdot \mathbf{u} = 0, \quad (1.6)$$

stating that the flow velocity field is divergence-free. Secondly, the incompressibility constraint is also valid for *immiscible* multiphase flows, for which there is no exchange of mass through the phase interfaces. Substituting Eq. (1.6) into Eq. (1.5) it follows

$$\frac{D\rho}{Dt} = 0. \quad (1.7)$$

Hence, by moving along with the flow speed \mathbf{u} , an observer would not measure any changes in ρ . This corresponds to a Lagrangian perspective on the flow, whereas rewriting Eq. (1.7) as a hyperbolic equation

$$\frac{\partial\rho}{\partial t} + \mathbf{u} \cdot \nabla\rho = 0 \quad (1.8)$$

for the density ρ perceives the flow from an Eulerian perspective.

Analogous to the mass, a conservation statement for the momentum $\rho\mathbf{u}$ can be derived. The change of momentum $\rho\mathbf{u}$ in a control volume is not only governed by the net inflow of momentum, but also by the forces acting on the volume V and its surface ∂V . Thus the momentum balance for the control volume becomes

$$\int_V \Delta(\rho\mathbf{u}) dV = \int_I \int_V \mathbf{f} dV dt - \int_I \oint_{\partial V} (\rho\mathbf{u} \otimes \mathbf{u})\mathbf{n} d\Gamma dt - \int_I \oint_{\partial V} \boldsymbol{\sigma}\mathbf{n} d\Gamma dt. \quad (1.9)$$

In this equation, the non-linear term $\rho\mathbf{u} \otimes \mathbf{u} := \rho u_i u_j$ is the advective transport. Furthermore, \mathbf{f} represents the body forces acting on the fluid volume V (e.g. gravity) and $\boldsymbol{\sigma}$ is the Cauchy stress tensor acting on the control volume surface ∂V . This tensor

includes the pressure and viscous stress, and for an incompressible, Newtonian fluid it is defined as:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\nabla^s\mathbf{u}, \quad (1.10)$$

with p the pressure, \mathbf{I} the identity tensor, μ the dynamic viscosity and ∇^s is the symmetric gradient operator $\nabla^s(\cdot) = \frac{1}{2}\nabla(\cdot) + \frac{1}{2}\nabla(\cdot)^T$. Assuming \mathbf{u} differentiable, and applying the divergence theorem to the surface integrals, while exploiting that the equality must hold for arbitrary volumes V , the momentum equations in conservative form are obtained:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = \mathbf{f} - \nabla p + \nabla \cdot (2\mu\nabla^s\mathbf{u}) \quad (1.11)$$

Using the identity $(\mathbf{a} \otimes \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \otimes (\mathbf{b} \cdot \mathbf{c})$, and after elimination of the continuity equation, Eq. (1.5), the momentum equation in an Eulerian frame of reference reads

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = \mathbf{f} - \nabla p + \nabla \cdot (2\mu\nabla^s\mathbf{u}). \quad (1.12)$$

Expressing the Lagrangian derivative defined in Eq. (1.4) in terms of the velocity \mathbf{u} , this equation is written in Lagrangian form as

$$\rho \frac{D\mathbf{u}}{Dt} = \mathbf{f} - \nabla p + \nabla \cdot (2\mu\nabla^s\mathbf{u}), \quad (1.13)$$

which states that the acceleration of a fluid parcel is caused by the external force \mathbf{f} , the pressure gradient ∇p and the diffusive force $\nabla \cdot (2\mu\nabla^s\mathbf{u})$.

From a physical perspective, the pressure is related to the density by an equation of state. However, for an incompressible flow, the pressure is implicitly determined by the incompressibility constraint, Eq. (1.6). This equation, together with the momentum conservation law in the Eulerian frame of reference, Eq. (1.12), or the momentum conservation law in the Lagrangian frame of reference, Eq. (1.13), form the so-called incompressible Navier-Stokes equations for Newtonian fluids.

Finally, other field quantities such as sediment concentration, temperature, or density can be carried along with the flow. With the mass and thermal energy being conserved quantities, the evolution of these quantities is governed by a scalar-valued conservation law, which in an Eulerian frame of reference reads:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi - \kappa\nabla\phi) = f, \quad (1.14)$$

in which κ a diffusion coefficient, ϕ the conserved quantity, and f a source term. Complemented with proper initial and boundary conditions, approximately solving the transport problems Eq. (1.14) and the more complicated and non-linear Navier-Stokes equations Eqs. (1.6,1.13) will be central to this thesis.

1.2.3. Dimensionless numbers

The general behavior of the physical and numerical solution for transport problems is characterized by dimensionless numbers, which are obtained by weighting the relative importance of the different physical processes. To illustrate this, consider the scalar-valued transport problem, Eq. (1.14). By assuming an incompressible flow, constant

diffusivity κ , and neglecting the source term f , the two different physical processes become apparent by rewriting Eq. (1.14) as

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \kappa \nabla^2 \phi, \quad (1.15)$$

in which the left-hand side is recognized as the material derivative describing the advective transport, and the right-hand side is a diffusion term.

Assuming $\|\mathbf{u}\|$ to be a characteristic magnitude for the advective transport, and introducing a typical problem length scale \mathcal{L} , an order of magnitude estimate for the ratio of the advection term over the diffusion, yields the dimensionless number

$$Pe = \frac{\|\mathbf{u}\| \mathcal{L}}{\kappa}, \quad (1.16)$$

which is known as the Péclet number. In a numerical setting, the typical length scale \mathcal{L} is often set to h , with h being a measure for the resolution of the spatial discretization, such as the mesh size. With this choice, the resulting dimensionless number is also known as the mesh Péclet number.

For $Pe \ll 1$, the diffusive term dominates over the advective transport, and a cloud of material spreads uni-directionally with a size proportional to $\sqrt{\kappa t}$. Since the total mass must be conserved in the absence of sources, while the size of the cloud extends in space, the maximum concentration of material in the cloud decreases. For $Pe \ll 1$ the transport problem, Eq. (1.14), can be classified as *parabolic*, and steep gradients in the concentration field are smoothed out as time progresses. Discretization schemes employing Eulerian control volumes, have been proven to be particularly apt for such parabolic problems.

For $Pe \gg 1$, advection dominates over diffusion. In this regime, which is henceforth labeled as the *advection-dominated regime*, a cloud of material is transported along the characteristic lines in the (\mathbf{x}, t) plane, while the diffusion or dissipation is relatively small, but not necessarily absent. Similar to the diffusion dominated regime, mass cannot (dis)appear in the absence of sources and sinks, and so the total mass must be conserved. Contrary to diffusion dominated flows, however, the physical damping is small and steep gradients persist as time progresses. In the advective limit, i.e. for $\kappa = 0$ and $Pe \rightarrow \infty$, the transport problem can be classified as hyperbolic and no changes are perceived if one moves along with the flow as a Lagrangian observer

Where the Péclet number provides a dimensionless number for the scalar transport equation, the dimensionless equivalent for the vector-valued momentum conservation equation (1.13) is known as the Reynolds number Re . When viscous diffusion dominates - i.e. Stokes flow - , $Re \ll 1$, whereas for advection dominated flows $Re \gg 1$. With the physical dissipation being low for the latter case, instabilities introduced by the non-linear advection term tend to grow, typically resulting in turbulent flow fields. Even more than for the scalar-valued case, high Re flows are challenging for numerical methods employing an Eulerian frame of reference due to dominant influence of the non-linear advection term.

1.2.4. Eulerian and Lagrangian numerical approaches

Unless the domain of interest and/or the boundary and initial conditions meet specific requirements, it is usually impossible to obtain closed analytical expressions satisfying the conservation laws for flow and transport. In such situations, approximately solving these equations using computers is the only viable option. This requires the transformation of the continuous conservation statements into a discrete set of equations that can be solved by computers, and a wealth of numerical methods have been developed to accommodate this. These approaches can be broadly categorized into Eulerian mesh-based and Lagrangian particle-based schemes. Without pretending to be complete, the principal features of these two categories of numerical schemes are briefly highlighted.

Traditionally, numerical techniques for approximately solving the transport equations take on an Eulerian perspective by partitioning the domain of interest into a mesh of non-overlapping and fixed control volumes, also called *cells*. Using this predefined mesh topology, discrete versions of the conservation laws are formulated, where the mesh is used for evaluating spatial gradients and for imposing the boundary conditions. Exploiting the static topology of the mesh, the discretization process yields efficient data structures, particularly so on Cartesian meshes. Well-established examples of mesh-based methods are, among others, the finite difference (FD), the finite volume (FV) and the finite element (FEM) method, for which the distinctive main principles in approximating a function $f(x)$ and its derivative $f'(x)$ are sketched in Fig. 1.2. Characteristic to the FD method is to approximate spatial gradients by means of Taylor-series expansions between the discrete, and typically regularly ordered data points. The FV method expresses the continuous equations in terms of local balances over each cell or ‘finite volume’. The inflow and outflow from a volume is formulated in terms of *fluxes* at the cell boundaries or *facets*. In the FE method, a set of basis functions is used to discretize a weak formulation of the governing conservation laws.

Many of the formulations classifying as either one of these mesh-based frameworks are provably high-order accurate. In addition, flux-based discretization strategies typically possess attractive local (i.e. cellwise) conservation properties. FV methods clearly belong to this category, but more recently the idea of numerical fluxes has also been incorporated in a FEM framework, resulting in the discontinuous Galerkin (DG) methods (see, e.g., [1, 2] and references) and hybridized discontinuous Galerkin (HDG) methods (see, e.g., [3–8]). In the FEM community, DG and HDG formulations are typically considered as a generalization of the FV formulation [1, 9].

Despite the attractive properties of mesh-based discretization techniques, application to advection-dominated flows remains challenging. This mainly roots in the advection term, and mesh-based discretizations of this term tend to exhibit non-physical oscillations. Many different approaches have been developed for stabilizing the advection operator such as (streamline) upwinding [10], Galerkin Least Squares stabilization [11], and stabilization via variational multiscale techniques [12]. These measures, however, typically introduce numerical diffusion which may obscure principal features of advection-dominated problems. The difficulties associated with the stabilization of the advection term are avoided by moving the mesh in a Lagrangian fashion along with the flow. This Lagrangian mesh-based approach, however, leads to mesh distortion and entanglement due to the extreme deformations typical to flow problems. Rezoning and remeshing

strategies are therefore indispensable, see e.g. [13, 14], but come with the price of complicating the solution process considerably.

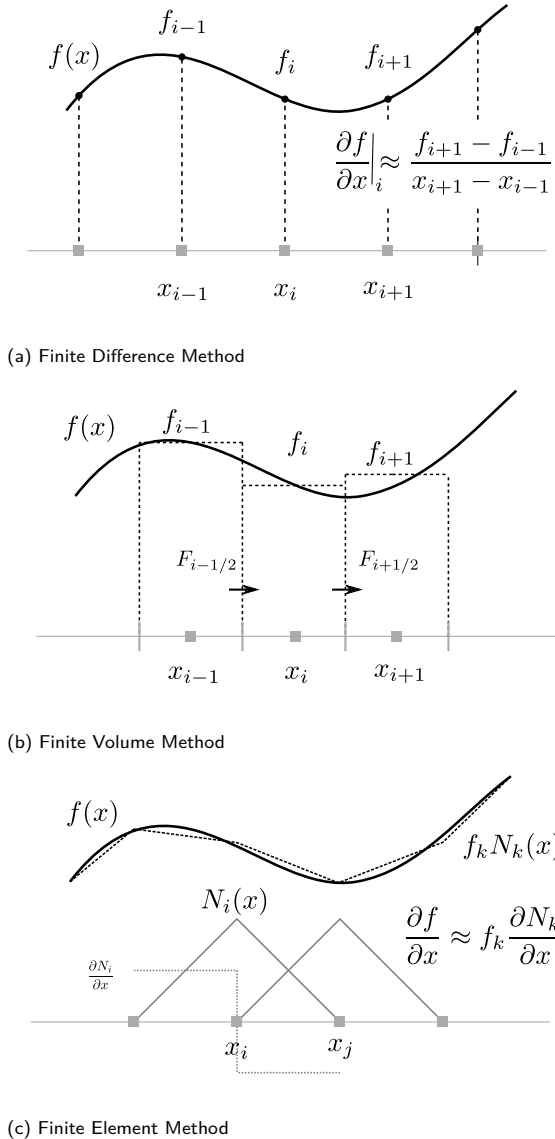


Figure 1.2: Schematic of different mesh-based discretizations, figure modified from [15].

Particle-based methods on the other hand, circumvent the challenges associated with the advective derivative. By using a cloud of unconnected particles - where each particle carries the material properties - discrete advection drops down to updating the particle

positions in a prescribed flow field. This naturally avoids numerical dissipation in the advection part, which is purely kinematic. The interaction between the particles - i.e. the constitutive modeling - is however less trivial in particle-based methods. The root cause for this is two-fold, in that i) particles typically are infinitesimal points, not possessing a metric for evaluating integrals, and ii) the moving particle cloud lacks a predefined topology, and therefore lack a structure for evaluating gradients. In order to overcome this, particle-based methods such as the smoothed particle hydrodynamics (SPH) method [16] and the moving particle semi-implicit (MPS) method [17] equip a particle with an interpolation kernel. This kernel has a finite size, and defines the influence of a particle on its surroundings, while also providing a means for evaluating gradients on a particle cloud, see Fig. 1.3. Though being conceptually simple, its implementation entails various drawbacks. Among these, the most notorious ones are i) the lack of C^0 consistency of the particle field, which hampers the extension to high-order accuracy [18, 19]), ii) the treatment of boundary conditions [20, 21] and iii) the change in the particle positions requires an update of particle neighbor lists and recomputing the gradients.

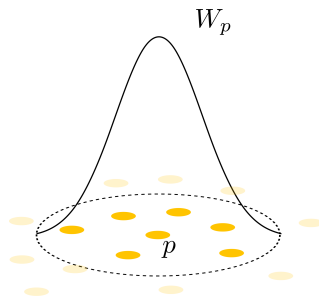


Figure 1.3: Schematic illustration of kernel function W_p limiting the influence domain of a particle p in Lagrangian particle-based schemes.

By comparing Eulerian mesh-based methods versus Lagrangian particle-based strategies for advection-dominated flow and transport modeling, it is clear that the approaches are more or less complementary: the strengths of a mesh-based approach are the weak spots of a particle-based approach and *vice versa*. This is summarized in Table 1.1. A logical question therefore is whether the advantages of the mesh-based schemes can be combined with those of a particle-based approach, while of course avoiding their respective disadvantages.

1.2.5. Particle-mesh methods

In an attempt to reconcile the advantages of a mesh-based with a particle-based approach, hybrid particle-mesh methods make a combined use of Lagrangian particles and an Eulerian background mesh. Historically, these particle-mesh methods trace back to the 1950s with the development of the particle-in-cell method (PIC) [22, 23]. Ever since, many variations to the original approach have been proposed. The fluid implicit particle (FLIP) method was introduced by Brackbill and Ruppel [24] to reduce the numerical diffusion introduced in the particle-mesh coupling of the original PIC formulation. The

Table 1.1: Eulerian mesh-based versus Lagrangian particle-based schemes: summary of pros and cons for advection-dominated flows.

	Eulerian mesh-based	Lagrangian particle-based
Advection	⊗	⊙
Constitutive modelling	⊙	⊗
Efficient	⊙	⊗
Accurate	⊙	⊗

FLIP method became especially popular in the computer graphics industry (see e.g. [25]). In recent work the method has also been applied in engineering applications for simulating free-surface flows by using particles as a convenient means to track free surfaces [26–29]. An extension of the FLIP method to history-dependent materials was made by Sulsky and coworkers [30]. Their method is currently known as the material point method (MPM) and is particularly attractive for simulating large deformation problems in solid mechanics by tracking history dependent quantities - such as plastic strain and strain-hardening parameters - on the Lagrangian particles. Recently, various attempts have been undertaken to use MPM for the simulation of incompressible fluid flows [31, 32].

Typical to particle-mesh schemes is to handle the kinematic advective part of a problem on Lagrangian particles, while a mesh is employed to efficiently solve for the dynamic, constitutive part, see Fig. 1.4. This strategy avoids the difficulties associated with a mesh-based discretization of the advection part, in that no diffusive stabilization measures are required. At the same time, the mesh is used for evaluating spatial gradients, thus avoiding the expensive construction of particle neighbor lists as required in fully Lagrangian, particle-based methods. Furthermore, a wealth of existing FD, FV, or FEM solvers can be used to efficiently solve the constitutive equations at the Eulerian mesh. As seen in Fig. 1.4, two data transfer operations are required to couple the particles with the mesh. In the literature, these steps go by various names such as ‘gather-scatter’ steps [33, 34], ‘forward interpolation - backward estimation’ [35] or ‘particle weighting’ [36]. In this thesis, and for reasons that will become clear in Chapter 2, the data transfer operators will be consistently coined ‘particle-mesh projection’ - to denote the data transfer from the set of scattered particles to the mesh - and ‘mesh-particle projection’ to indicate the opposite route. The plural ‘particle-mesh projections’ will be somewhat loosely used for the combination of the particle-mesh projection and the mesh-particle projection step.

The different conventions by which the data transfer operations between the particles and the mesh are known, indicate that there exists at least the same amount of different formulations for these transfer operations. These are broadly categorized into two classes. Historically, PIC and related methods, such as MPM, consider the particles as moving point masses and formulate the particle-mesh interactions in terms of a summation of particle properties, see, e.g., [24, 37–39]. While it guarantees conservation of total mass and momentum, this approach has low-order accuracy only [40]. More recently, particle-mesh schemes have been proposed in which the particles are

considered as moving sampling points of the continuum so as to achieve high-order accuracy. Reconstruction of the mesh-based data from the scattered particle data is typically done using moving least squares (MLS) [40, 41], which compromise however exact conservation. An as yet unanswered question is whether and how accuracy and exact conservation properties can be reconciled in the particle-mesh projections pertinent to any hybrid particle-mesh scheme.

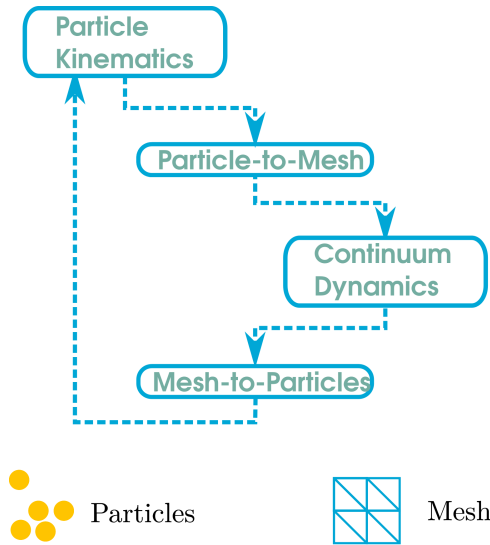


Figure 1.4: Schematic of the four steps constituting a hybrid particle-mesh scheme.

A second challenge, specifically related to particle-mesh methods for incompressible flow problems, is the degradation of an initially uniform particle distribution into a non-uniform distribution. This artifact is shared with fully Lagrangian, particle-based methods such as SPH, see e.g. [42, 43]. Pope [44] identifies inaccuracies in the divergence of the advective velocity field to be among the main reasons for the degradation of the particle distribution. Based on this observation, various authors have proposed velocity field reconstruction techniques in order to maintain uniform particle distributions in the incompressible limit [45–47]. Other authors pursue a more heuristic particle distribution quality control by introducing weak spring forces between particles [26, 42, 48], or using particle reseeding [41] or particle splitting [43] techniques.

1.3. Present research

1.3.1. Objectives and scope

With Eulerian mesh-based and Lagrangian particle-based schemes being largely complementary in terms of their (dis)advantages when applied to advection-dominated flows, this research aims at bridging the gap between these two approaches by further developing PIC concepts. The central question addressed in this thesis is how a method can be

formulated possessing the diffusion-free advection property as in a Lagrangian particle-based scheme, while simultaneously inheriting the accuracy and conservation properties of state-of-the-art mesh-based methods, Fig. 1.5. In doing so, the thesis aims to seamlessly integrate these two modeling paradigms, and to provide a generic particle-mesh framework for advection dominated flows that appeals to both mesh-based as well as particle-based practitioners.

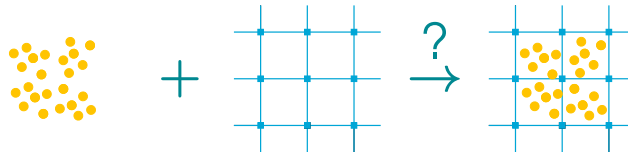


Figure 1.5: Cartoon of the central question of this thesis: how to combine particle-based techniques and mesh-based techniques for solving advection dominated flows?

Several sub-objectives mark the route for achieving this. A first step is to develop coherent formulations for the particle-part, the mesh-part and the role played by the particle-mesh interactions. By formulating a framework for unifying these different components, the question will be answered which fundamental requirements are to be met by particle-mesh schemes. Secondly, the particle-mesh interactions are further investigated. The main challenge in this part is to couple the particle and the mesh in such a way that high-order accuracy can be achieved in time and space, while simultaneously satisfying discrete conservation principles. The resulting particle-mesh scheme is applied to the scalar transport equation and the incompressible Navier-Stokes equations, and the schemes are analyzed in terms of consistency, conservation and accuracy. Successfully applying the particle-mesh strategy to the incompressible Navier-Stokes equations requires that the particle distribution remains of high-quality. A third sub-objective is therefore to answer the question how a uniform particle distribution can be maintained, so that heuristic particle shifting algorithms can be avoided. The building blocks developed in this thesis are finally pieced together and as a fourth sub-objective, the applicability of the particle-mesh scheme to simulate multiphase flow problems is assessed.

1.3.2. Practical Relevance

The nature of the open challenges in particle-mesh schemes, ask for a fundamental approach. Starting with a simple model problem - i.e. the scalar advection-diffusion equation - the approach is gradually extended towards the more complex problem of the (multiphase) incompressible Navier-Stokes equations. This research strategy prevents jumping towards applications without understanding the model. This also implies that the behavior of the scheme is thoroughly tested for academic examples before moving towards more application-oriented examples.

By pursuing a fundamental approach, it is believed that the developed particle-mesh tools can be of use to a wide range of applications. In the second-half of this thesis, one such application is considered in more detail, namely the simulation of multi-phase flows in a civil engineering hydraulics context. In particular, the particle-mesh tools developed

in this thesis are combined to simulate non-breaking and breaking waves in a ‘numerical wave flume’. Other areas of application include, for instance, the estimation of source terms in turbulent flows [35, 49], the simulation of sedimentation of erosion and fouling in turbomachinery [50], the simulation of contaminants in groundwater flow [51], the dispersion of airborne pathogens in hospital rooms [52], or other advection-dominated and/or particle-laden flows.

To foster future developments in maybe yet unforeseen application areas, the computer code containing the essential building blocks for the particle-mesh framework developed in this research is publicly available. This software, coined LEOPart⁴, can be used and modified under an open-source license and is hosted at https://bitbucket.org/jakob_maljaars/leopart.

1.3.3. Outline

The remainder of this thesis is structured as follows. In Chapter 2, definitions and mathematical building blocks are introduced for deriving the particle-mesh schemes in later chapters. This includes the introduction of a mesh-based FEM framework, an explanation of the role played by the particles and the mesh in the hybrid particle-mesh setting, and a first interpretation of the interactions between the particles and the mesh. Building further upon this framework, Chapter 3 presents a particle-mesh scheme for the scalar advection-diffusion equation. Major contribution of this chapter is the formulation and analysis of a novel particle-mesh projection strategy, which can be made high-order accurate while simultaneously possessing (local) conservation properties. Chapter 4 extends this approach to the incompressible Navier-Stokes equations. New light is shed on various long-standing issues in particle-based methods for incompressible flows. Particularly, it is shown under which conditions the particle distribution remains of high quality. Section 5 illustrates the potential and the challenges of the developed scheme for solving multiphase problems, and the applicability of the scheme to reproduce laboratory wave flume tests is demonstrated. Section 6 concludes this thesis by drawing general conclusions and discussing avenues for future research. Appendices A-D present detailed background information on a number of topics, including a discussion of the computer implementation and the computational performance of the particle-mesh tools available in LEOPart.

⁴LEOPart is so much as to say **L**agrangian-**E**ulerian on **P**articles, and might remind of the animal with particles imprinted on its skin

2

Particles and finite elements

Having introduced particle-mesh methods briefly in the preceding chapter, taking a closer look at such methods is the central topic of this chapter. The basic concepts for particle-mesh modeling of advection-dominated flows are developed, and a generic framework is sought to formulate the different components of the method. This requires a coherent interpretation for the interactions between the particles and the mesh, i.e the particle-mesh projections. In order to do so, a mesh-based finite element perspective is adopted, and the finite element method and its relation with the overarching variational methods is introduced. Rather than presenting new theory on these topics or presenting an exhaustive discussion, the aim is to highlight some concepts which are pertinent to this research. For a comprehensive introduction to FEM and variational methods, the interested reader is referred to a number of textbooks such as Zienkiewicz & Taylor [53], Gresho & Sani [54], Donea & Huerta [55], and Reddy [56].

With the variational tools at hand, it becomes clear in the second half of this chapter that a variational approach in conjunction with finite element concepts provides a useful and convenient way of interpreting and formulating particle-mesh schemes. Specifically, a variational approach is shown to provide a generic framework for formulating the mutual data transfer operations between the particles and the mesh, as well as the mesh step.

The ideas in this chapter are developed in an abstract setting, and finally illustrated for a 1D diffusion example. As such, this chapter provides the conceptual interpretation and building-blocks which are extended to advection-dominated problems in subsequent chapters, for which the presence of (particle) advection will pose additional challenges.

2.1. Particle-mesh modeling

Starting point is the blueprint for particle-mesh schemes, sketched in Fig. 1.4. Advection is done using a cloud of Lagrangian particles, which has the advantage in that it avoids the difficulties typically associated with the stabilization of mesh-based discretizations of the advection term. On the other hand, an Eulerian mesh is used as in a purely

mesh-based method to allow an efficient discretization of the constitutive part of the problem in a different step of the algorithm. Specific to the model problems under consideration in this research, this mesh-step amounts to solving a diffusion equation or a Stokes problem. To couple the particle-step with the mesh-step and *vice versa*, a particle-mesh projection and a mesh-particle projection are required. In this section, we interpret the sequence of steps composing a particle-mesh scheme, and take a closer look at the particle-mesh projections in order to formulate a number of requirements for these steps.

2.1.1. Operator splitting

Leaving the data transfer operations aside for the moment, the idea of splitting a complex (partial) differential equation into two or more sub-problems is a well-known strategy in numerical mathematics. To explain this idea, which is also known as ‘fractional step’ or ‘operator splitting’, consider the time-dependent system of linear ordinary differential equations (ODEs)

$$\frac{d\varphi(t)}{dt} = (\mathbf{A} + \mathbf{B})\varphi(t), \quad (2.1)$$

in which \mathbf{A} and \mathbf{B} linear operators, and φ is a vector of coefficients. Given the initial condition $\varphi(0) = \varphi_0$, this ODE has the analytical solution at time $t = t^1$:

$$\varphi(t^1) = \exp((\mathbf{A} + \mathbf{B})t^1) \varphi_0. \quad (2.2)$$

Instead of solving problem Eq. (2.1) in one shot, the solution can be approximated by solving for the operators \mathbf{A} and \mathbf{B} *sequentially*, i.e.

$$\frac{d\psi}{dt} = \mathbf{A}\psi, \quad \text{where} \quad \psi(0) = \varphi_0, \quad (2.3a)$$

$$\frac{d\phi}{dt} = \mathbf{B}\phi, \quad \text{where} \quad \phi(0) = \psi(t^1), \quad (2.3b)$$

yielding the solution at time t^1 :

$$\phi(t^1) = \exp(\mathbf{B}t^1) \exp(\mathbf{A}t^1) \varphi_0. \quad (2.4)$$

For the general case, $\phi(t^1)$ is not equal to the exact solution $\varphi(t^1)$ in Eq. (2.2), but provides a first-order approximation. Albeit inevitably introducing such a *splitting error*, operator splitting techniques are widespread to reduce the problem complexity in the discrete setting by, e.g., splitting the linear and the non-linear contributions of the encompassing equations, splitting the spatial dimensions, or splitting the terms corresponding to different physical processes. In the scope of the incompressible Navier-Stokes equations, for instance, a particularly popular splitting strategy is to decouple the incompressibility constraint from the momentum equations, leading to the Chorin-Temam pressure projection schemes [57]. A vast body of literature exists in which operator splitting schemes of high-order accuracy are developed and analyzed, where it is customary to do so by focusing on time dependent problems only, or assuming a mesh-based spatial discretization, see e.g. [58–61].

Considering the sequence of steps in Fig. 1.4, the particle-mesh approach can also be conceived as an operator splitting method, in which the Lagrangian particle advection stage is separated from the constitutive stage at the mesh. This interpretation of particle-mesh schemes, henceforth interchangeably called *particle-mesh operator splitting*, is important in several aspects. First of all, interpreting the particle-mesh scheme as a special kind of operator splitting provides a framework for developing time integration methods by closely following existing operator splitting schemes. In particular, this avoids the pitfall of overlooking the splitting error. Even more importantly, an operator splitting perspective elucidates where particle-mesh schemes deviate from established mesh-based techniques. Whereas a mesh-based operator splitting typically employs the same spatial discretization principles throughout the sequence of sub-steps, this is not so for particle-mesh schemes in which part of the problem is solved in a *particle space* and the other part is discretized in a *mesh space*. This mismatch in the spatial discretization requires the introduction of data transfer operations, which are interpreted as projection operators between the particle space and the mesh space and *vice versa*. This motivates the adopted convention to call these data transfer operators ‘particle-mesh projection’ for the transfer of information from the particles to the mesh and ‘mesh-particle projection’ for the opposite route, denoted by $\mathcal{P}_E(\cdot)$ and $\mathcal{P}_L(\cdot)$, respectively. Returning to the simple *sequential* split, Eq. (2.3), and assuming that \mathbf{A} and ψ are defined at particle level, whereas φ_0 , ϕ and \mathbf{B} are defined at the mesh, problem Eq. (2.3) is reformulated in a, yet abstract, particle-mesh framework as

$$\frac{d\psi}{dt} = \mathbf{A}\psi, \quad \text{where} \quad \psi(0) = \mathcal{P}_L(\varphi_0), \quad (2.5a)$$

$$\frac{d\phi}{dt} = \mathbf{B}\phi, \quad \text{where} \quad \phi(0) = \mathcal{P}_E(\psi(t^1)). \quad (2.5b)$$

This illustrates that the mesh-particle projection \mathcal{P}_L provides the condition to advance the particle part of the equation, and the particle-mesh projection \mathcal{P}_E provides the condition for advancing the mesh-based solution step. In view of later extensions it is mentioned that the notation $\phi(0) = \phi_0 = \varphi_0$ could have been used in Eq. (2.5a), since φ_0 and ϕ are defined at the mesh and Eq. (2.5) is solved *sequentially*, that is, Eq. (2.5b) is solved after Eq. (2.5a).

2.1.2. From particle space to mesh space and back

With the interpretation of the particle-mesh interactions as projections, the next step is to further specify the role played by the particle-mesh projection \mathcal{P}_E and the mesh-particle-projection \mathcal{P}_L , and to formulate a number of requirements for these projections. To do so, a most fundamental question needs to be answered first: what does a particle actually represent? In the original PIC, FLIP and MPM formulations, particles are interpreted as integral quantities, i.e. blobs of material or *volume particles*. Even though they are infinitesimally small points in space, particles are assigned integral quantities, such as volume, mass and momentum [24, 37–39]. With the mesh being composed of finite sized cells, it follows that the mesh and the particles are different representations of the same fields. The transfer of information from the particle representation to the mesh representation of the material is then typically casted as a summation of particle

properties, see e.g. [24, 37–39]. While this approach readily yields global conservation of linear quantities, such as mass and momentum, it is restricted to low-order accuracy [40]. In addition, the summation of particle properties does not ensure that the particle fields are locally consistent with the mesh description of the material. For example, the summed volume represented by the particles within a discretization mesh cell typically does not match the volume associated with the cell. This impediment is in fact a manifestation of the lack of the earlier mentioned C^0 consistency in purely particle-based methods, and results in e.g. non-optimal accuracy of the numerical integration in MPM, and the so-called ‘grid-crossing’ error [62, 63]. Attempts have been made in the MPM community to mitigate these problems by assigning a finite shape to the particles instead of a Dirac delta shape [62, 64], by using basis functions being at least C^1 -continuous [63, 65], or by using a heuristic blend between optimal quadrature and a quadrature based on moving material points [31].

A different viewpoint gaining traction in recent literature, is to consider the particles as moving *sampling points* of the continuum at the mesh, rather than the traditional perception of particles being volume particles [40, 41, 66]. Hence, particles do not carry integral quantities such as mass and momentum, but rather carry point evaluations of an underlying density, velocity or momentum field. This seemingly subtle difference has large implications on the particle-mesh projection: rather than that mesh fields are obtained by *summation* over the integral quantities of particles, mesh fields are *reconstructed* from the particle properties. This reconstruction of the continuous fields from the scattered particle data is done using, e.g., spline reconstruction techniques [65], leading to global systems for the particle-mesh projection step, or a moving least-squares (MLS) projection [40, 41], which requires the choice of an appropriate particle weighting function. The major advantage over the volume particle approach is that higher-order accuracy can be achieved. On the downside, such an approach as yet compromises exact conservation properties [41]. The difference between a ‘volume particle’ and a ‘sampling point’ perspective for obtaining a density field ρ at the mesh, is schematically depicted in Fig. 2.1.

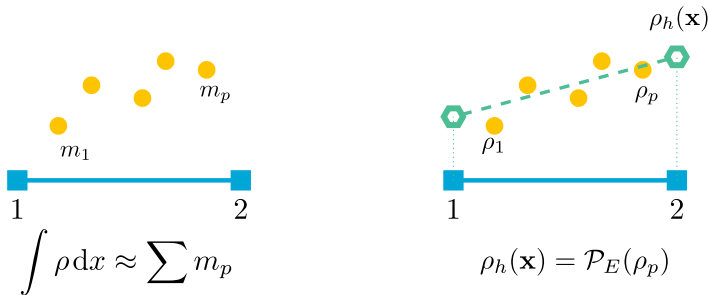


Figure 2.1: Sketch of a ‘volume particle’ approach (left) versus a ‘sample particle’ approach (right) for the approximation of a density field on a one-dimensional cell.

Despite this lack of conservation properties, it is hypothesized that a point-based approach is more appealing than a volume particle approach in several aspects. Particularly,

it is expected that the low physical damping in advection-dominated flows is prohibitive to allow noise, introduced by low-order accurate or inconsistent particle-mesh interaction schemes. When applied to advection-dominated flows, particle and particle-mesh based methods using mass particles typically rely on some form of numerical diffusion to oppress noise in the particle-mesh interactions, see e.g. [31, 32].

Having decided upon a point-based interpretation of particles, the precise role of the particle-mesh projection steps are to be further elucidated. From the foregoing, it follows that the physical processes are conducted at either the particle level (advection) or the mesh level (constitutive modeling). This implies that the particle-mesh projections themselves are merely auxiliary steps, and should seamlessly bridge the gap between the particle space and the mesh space and *vice versa*. To make this more precise, a number of requirements are formulated for the projection steps:

- **Accuracy:** the particle-mesh projections should not obstruct higher-order accuracy. In terms of the spatial accuracy, this translates into the requirement that the projections should have at least the same order of spatial accuracy as the mesh step or the particle advection step. In terms of the temporal accuracy, it implies that the particle-mesh projections should not violate the accuracy of the time integration scheme employed for the particle-mesh operator splitting.
- **Consistency:** the particle-mesh projections are purely auxiliary, hence these steps clearly should not introduce ‘spurious physics’. This imposes a consistency requirement on the particle-mesh and mesh-particle projection in that a repeated back-and-forth mapping between particles and the mesh should not induce artifacts. Phrasing this more mathematically: the compositions $\mathcal{P}_E \circ \mathcal{P}_L$ and $\mathcal{P}_L \circ \mathcal{P}_E$ should return the identity operator.
- **Conservation:** with a wealth of mesh-based approaches possessing (local) conservation properties, the particle-mesh projections should comply with these discrete conservation properties.
- **Efficiency:** the computational costs associated with the projections should stay as low as possible, the more so since the primary understanding of the projections is that they contribute to the overhead costs.

These requirements show the tight relationship between the particle projections and the choices made in the other steps of a particle-mesh scheme. Hence, a generic framework for gluing together the particle and mesh part by virtue of the projection steps is considered a necessity rather than a luxury. To build such a framework, a mesh-based perspective is adopted, and it will be argued next that a Galerkin finite element method - as a branch of *variational methods* - provides the necessary ingredients for doing so.

2.2. Mesh step: the Galerkin finite element method

Therefore, let’s make a short digression by highlighting the essentials of the Galerkin approach - which became more or less a synonym for modern finite element approaches. To fit this purpose, consider an elliptic boundary value problem with non-homogeneous Dirichlet boundary conditions. The elliptic operator in this problem is classical to many physical problems and also serves as a model problem for later extensions to

the advection-diffusion equation, and the incompressible Navier-Stokes equations. The problem is, for the sake of illustration, written as a system of first-order equations and reads: given a source term f on the domain Ω , and a Dirichlet condition g on the boundary $\partial\Omega$, find $\boldsymbol{\sigma}$ and ϕ such that

$$\boldsymbol{\sigma} = \nabla\phi \quad \text{in} \quad \Omega, \quad (2.6a)$$

$$-\nabla \cdot \boldsymbol{\sigma} = f \quad \text{in} \quad \Omega, \quad (2.6b)$$

$$\phi = g \quad \text{on} \quad \partial\Omega. \quad (2.6c)$$

For $f = 0$, Eq. (2.6b) is mathematically similar to the incompressibility constraint, Eq. (1.6).

Multiplying Eq. (2.6a) with a test function \boldsymbol{w} , and Eq. (2.6b) with a test function q , and integrating over the domain Ω , results in the following weak formulation of the problem: given a source function f , find $\boldsymbol{\sigma} \in \boldsymbol{W}$ and $\phi \in Q$ such that

$$\int_{\Omega} (\boldsymbol{\sigma} - \nabla\phi) \cdot \boldsymbol{w} \, d\Omega + \int_{\Omega} (\nabla \cdot \boldsymbol{\sigma} + f) q \, d\Omega = 0 \quad \forall \quad (\boldsymbol{w}, q) \in (\boldsymbol{W}, Q), \quad (2.7)$$

where the unknowns $\boldsymbol{\sigma}$ and ϕ are known as *trial functions*, and \boldsymbol{w} and q are known as *test functions* spanning the function spaces \boldsymbol{W} and Q . Furthermore, it has been assumed that ϕ satisfies the Dirichlet boundary condition on $\partial\Omega$ and that test functions q vanish on this boundary.

With the unknowns $\boldsymbol{\sigma}, \phi, \boldsymbol{w}, q$ still belonging to infinite dimensional function spaces it is not immediately obvious that the weak formulation, Eq. (2.7), is easier to solve than the strong form of the problem Eq. (2.6). However, the weak formulation allows relaxing the stringent continuity requirements on the solution space and opens the possibility to seek approximate solutions in finite dimensional function spaces.

To take the step from the weak formulation to a set of algebraic equations, a choice for a finite dimensional basis has to be made. The assumption introduced at this stage is that the quantity of interest can be expressed in terms of a set of basis functions of a known shape. Of particular interest for this thesis are continuous Galerkin (CG) and discontinuous Galerkin (DG) methods, and a blend between these two coined hybridized discontinuous Galerkin (HDG).

2.2.1. Function spaces and mesh partitioning

To pave the way for introducing CG, DG and HDG methods, a short primer on function spaces is given. The weak formulation Eq. (2.7) is defined as long as the gradients $\nabla\phi$ and $\nabla \cdot \boldsymbol{\sigma}$ are square integrable on Ω . For a function, say u , this square integrability condition implies

$$\int_{\Omega} u^2 \, d\Omega < \infty. \quad (2.8)$$

Functions possessing this property are known as members of the Sobolev space of degree zero, denoted by $L^2(\Omega)$ for scalar valued functions, i.e. $u \in L^2(\Omega)$ or $\boldsymbol{L}^2(\Omega)$ for vector-valued functions. Members of the Sobolev space of degree one, denoted $H^1(\Omega)$ will be

frequently used also, and possess square-integrability of the derivative in addition to the $L^2(\Omega)$ property. For a function u , this means

$$\int_{\Omega} (u^2 + \nabla u \cdot \nabla u) \, d\Omega < \infty, \quad (2.9)$$

in which case u belongs to the Hilbert space $H^1(\Omega)$, or $\mathbf{H}^1(\Omega)$ for vector-valued functions. Clearly, it holds that $H^1(\Omega) \subset L^2(\Omega)$, and functions $u \in H^1(\Omega)$ are at least C^0 continuous.

Discrete function spaces are defined on a partitioning of the domain of interest Ω into a set of disjoint simplicial *elements* K , having boundary ∂K , see Fig. 2.2. Consistent with notations which will be employed throughout this thesis, $\mathcal{T} := \{K\}$ denotes the set of cells, and the closure of a cell, i.e the cell including its boundaries, is denoted by $\bar{K} := K \cup \partial K$. Furthermore, adjacent cells K_i and K_j ($i \neq j$) share a common *facet* $F = \partial K_i \cap \partial K_j$. The set of all facets, including the exterior boundary facets $F = \partial K \cap \partial\Omega$, is denoted by \mathcal{F} .

On the cell partitioning of the domain, discrete function spaces are defined. For example, a popular choice due to Courant [67] is to choose piecewise continuous Lagrangian polynomials as the basis for these finite dimensional function spaces, with such a function space defined as

$$W_h := \{w_h \in H^1(\mathcal{T}) : w_h|_{\bar{K}} \in P_k(\bar{K}) \, \forall K \in \mathcal{T}\}, \quad (2.10)$$

in which P_k with $k > 1$ the Lagrange polynomials of degree k .

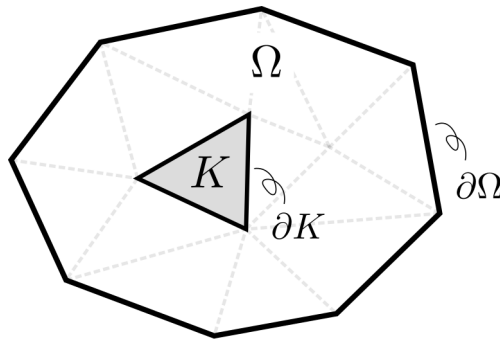


Figure 2.2: Schematic illustration of domain partitioning into simplices.

Given \mathcal{T} , many different alternatives to Eq. (2.10) can be devised. To limit the scope of this thesis, two *a priori* choices are made with respect to the function spaces:

- Trial- and test functions are usually chosen from the same discrete function space (Bubnov-Galerkin approach). For the weak formulation of Eq. (2.7) this implies that σ and w come from the same function space, and so do ϕ and q .
- The basis functions spanning the finite dimensional function spaces are assumed to be piecewise Lagrange polynomials.

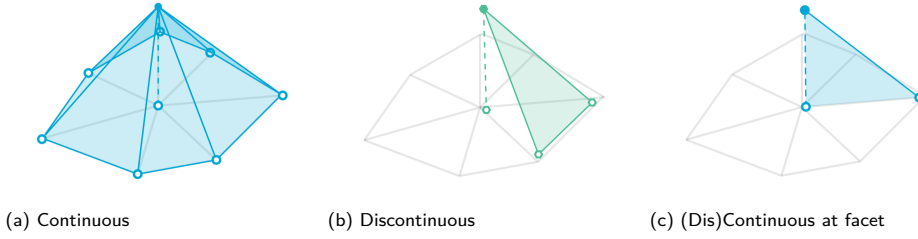


Figure 2.3: Schematic illustration of linear two-dimensional basis functions.

2.2.2. Continuous Galerkin method

In the continuous Galerkin (CG) method, the function spaces are defined such that the basis functions are continuous between elements similar to Eq. (2.10), see Fig. 2.3a. Therefore, the piecewise continuous function spaces

$$Q_h := \{q_h \in H^1(\mathcal{T}) : q_h|_{\bar{K}} \in P_k(\bar{K}) \forall K \in \mathcal{T}\}, \quad (2.11)$$

$$Q_{h,g} := \{q_h \in H^1(\mathcal{T}) : q_h|_{\bar{K}} \in P_k(\bar{K}) \forall K \in \mathcal{T}, q_h = g \text{ on } \partial\Omega\}, \quad (2.12)$$

are introduced, where $Q_{h,g}$ differs from Q_h only in that the Dirichlet boundary condition is satisfied at the domain boundary $\partial\Omega$. Given these function space definitions, an irreducible continuous Galerkin method of the mixed form Eq. (2.7) is obtained by substituting $\boldsymbol{\sigma} = \nabla\phi_h$ and $\boldsymbol{w} = \nabla q_h$. Applying integration by parts to relax the continuity requirement on the discrete function space, the variational problem, Eq. (2.7), can be restated as: find $\phi_h \in Q_{h,g}$ such that

$$\int_{\Omega} \nabla\phi_h \cdot \nabla q_h \, d\Omega - \sum_K \oint_{\partial K} \nabla\phi_h \cdot \boldsymbol{n} q_h \, d\Gamma - \int_{\Omega} f q_h \, d\Omega = 0 \quad \forall q_h \in Q_h, \quad (2.13)$$

with \sum_K denoting summation over all cells, and \boldsymbol{n} the outward pointing unit normal on cell boundaries or *facets*. Since the function space Q_h is continuous across facets, the flux terms over the facets cancel, so that Eq. (2.13) becomes: find $\phi_h \in Q_{h,g}$ such that

$$\int_{\Omega} \nabla\phi_h \cdot \nabla q_h \, d\Omega - \int_{\Omega} f q_h \, d\Omega = 0 \quad \forall q_h \in Q_h. \quad (2.14)$$

2.2.3. Discontinuous Galerkin method

Typical to discontinuous Galerkin (DG) formulations is to use broken function spaces. That is, functions from these spaces are assumed to be piecewise continuous on the interior of a cell K but can be discontinuous and thus double-valued at the facets ∂K , see Fig. 2.3b. This weaker continuity requirement has some distinct advantages as will become apparent shortly. To obtain a DG formulation of the weak form Eq. (2.7), consider the broken function space

$$Q_h := \{q_h \in L^2(\mathcal{T}) : q_h|_{\bar{K}} \in P_k(K) \forall K \in \mathcal{T}\}, \quad (2.15)$$

with the important difference compared to the CG function spaces in that functions $q_h \in Q_h$ are only square integrable on the *collection* of cells \mathcal{T} . On the *interior* of cells, functions q_h are assumed to have square integrable derivatives. Furthermore, as for Eq. (2.12), the subscript g implies that the Dirichlet boundary condition is satisfied.

Given these definitions, and by assuming $\mathbf{w}_h = \nabla q_h$ for all $q_h \in Q_h$ a DG formulation for the weak form Eq. (2.7) becomes, see [15] for details of the derivation:

$$\begin{aligned} \int_{\Omega} \nabla \phi_h \cdot \nabla q_h \, d\Omega + \sum_K \oint_{\partial K} (\hat{\phi}_h - \phi_h) \mathbf{n} \cdot \nabla q_h \, d\Gamma \\ - \sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_h \cdot \mathbf{n} q_h \, d\Gamma - \int_{\Omega} f q_h \, d\Omega = 0 \quad \forall q_h \in Q_h \end{aligned} \quad (2.16)$$

in which $\hat{\boldsymbol{\sigma}}_h$ and $\hat{\phi}_h$ are *numerical fluxes* on the facets ∂K .

With trial and test functions $(\phi_h, q_h) \in Q_h$ being discontinuous between elements, the DG formulation essentially expresses the problem as a local balance for each cell, and inter-continuity between cells is enforced by appropriately chosen numerical fluxes. Various formulations for these numerical fluxes have been proposed in literature, see [68], the exact formulation of which is irrelevant in the scope of this chapter. What is important, however, is that these numerical fluxes are expressed in terms of the traces of the primal variable ϕ_h , which is double-valued at facets, i.e. $\hat{\boldsymbol{\sigma}}_h(\phi_h)$ and $\hat{\phi}_h(\phi_h)$.

2.2.4. Hybridized Discontinuous Galerkin method

Hybridized DG (HDG) formulations have emerged in the past decade with the aim of retaining the advantages of DG methods, while alleviating the increased cost due to the larger number of degrees-of-freedom (dofs) compared to a CG formulation. Analogous to DG formulations, the conservation laws are cast as a summation of weak cellwise balances, where inter-element continuity is enforced by means of numerical fluxes as in Eq. (2.16). However, when it comes to defining these fluxes, HDG departs from DG approaches in that an additional variable $\bar{\phi}_h$ is introduced at the facets. This function belongs to a function space which is only defined on the cell facets or the mesh-skeleton, i.e.,

$$\bar{Q}_h := \{\bar{q}_h \in L^2(\mathcal{F}) : \bar{q}_h|_F \in P_k(F) \, \forall F \in \mathcal{F}\}, \quad (2.17)$$

where $k \geq 0$ a polynomial order. A schematic of such a HDG function space for $k = 1$ is given in Fig. 2.3c, where the additional remark is made that functions $\bar{q}_h \in \bar{Q}_h$ are single valued and continuous on facets but can in principle be discontinuous between facets.

Slightly more restrictive facet function spaces are obtained when C^0 continuity of the facet function spaces in the mesh vertices is enforced, i.e. $\bar{Q}_h \cap C^0(\mathcal{F})$. Methods employing these continuous facet function spaces became known as embedded Discontinuous Galerkin (EDG) methods [69], although a similar approach was independently invented in the context of the advection-diffusion problems as ‘Galerkin interface stabilization’ (GIS) [3]. A schematic of a linear EDG facet function space is depicted in Fig. 2.4, clearly illustrating the difference with a HDG facet function space.



Figure 2.4: Schematic of difference between L^2 trace space (left) and $H^{1/2}$ trace space (right) for two-dimensional element.

According to the definitions, functions $\bar{q}_h \in \bar{Q}_h$ are uniquely defined on facets - i.e. single-valued - and an obvious choice for the scalar-valued numerical flux is $\hat{\phi}_h = \bar{\phi}_h$. The exact formulation of the vector-valued flux $\hat{\sigma}_h$ is irrelevant for the time being, but crucial is to define this flux in terms of the single valued facet function $\bar{\phi}_h$ and the values ϕ_h belonging to a cell, i.e. $\hat{\sigma}_h(\phi_h, \bar{\phi}_h)$. The latter renders the numerical flux $\hat{\sigma}_h$ discontinuous across cell facets, and the additional requirement

$$\sum_K \oint_K \hat{\sigma}_h \cdot \mathbf{n} \bar{q}_h \, d\Gamma = 0 \quad \forall \bar{q}_h \in \bar{Q}_{h,0}, \quad (2.18)$$

is imposed to enforce weak continuity between adjacent cells. This additional equation is in fact required, due to the introduction of a new unknown $\bar{\phi}_h$.

Combining the *local equation*, Eq. (2.16), with the *global equation*, Eq. (2.18), yields the following variational problem: given the source term f , find $\phi_h \in Q_h$ and $\bar{\phi}_h \in \bar{Q}_{h,g}$ such that

$$\begin{aligned} & \int_{\Omega} \nabla \phi_h \cdot \nabla q_h \, d\Omega + \sum_K \oint_{\partial K} (\bar{\phi}_h - \phi_h) \mathbf{n} \cdot \nabla q_h \, d\Gamma \\ & + \sum_K \oint_{\partial K} \hat{\sigma}_h \cdot \mathbf{n} (\bar{q}_h - q_h) \, d\Gamma - \int_{\Omega} f q_h \, d\Omega = 0 \quad \forall (q_h, \bar{q}_h) \in (Q_h, \bar{Q}_{h,g}). \end{aligned} \quad (2.19)$$

2.2.5. Continuous, discontinuous or hybrid?

To appreciate the (H)DG method, recall that these approaches are based on cellwise balances, augmented with properly defined numerical fluxes. As a result of this localized approach, DG methods can be formulated such that the conservation laws are satisfied in a weak sense in each cell, see Fig. 2.5b. This in contrast to CG methods, which provide global conservation at best, see Fig. 2.5a. By introducing post-processing operators, some DG implementations push the conservation statement even further, and satisfy the incompressibility constraint, Eq. (1.6) - which is mathematically similar to Eq. (2.6b) with $f = 0$ - in a *pointwise* fashion [70]. This level of conservation, bears one-to-one correspondence with the strong form of the governing equations, and is schematized in Fig. 2.5c. The importance of this pointwise enforcement of the continuity equation in the context of the incompressible Navier-Stokes equations, was emphasized in recent literature, see e.g. [8, 71]. Apart from the (local) conservation properties, other

advantages of (H)DG approaches - but slightly less relevant to this research - are the amenability to incorporate adaptivity strategies, e.g. locally increasing the spatial resolution or increasing the polynomial order of the basis functions, its ability to handle sharp discontinuities, and the native implementation of upwinding to stabilize the advection operator.

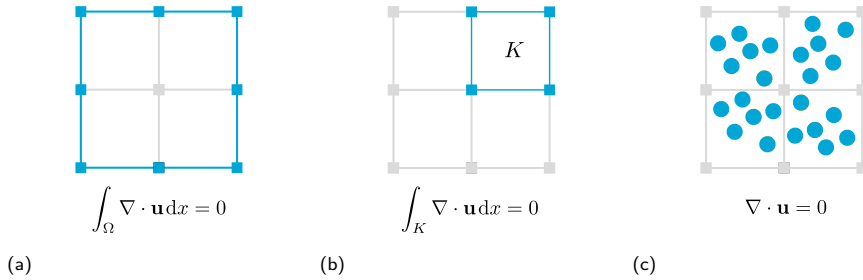


Figure 2.5: Schematic of different levels of discrete conservation (a) globally, (b) locally, and (c) pointwise div-free.

Finally, the distinguishing features of HDG methods compared to the classical DG formulations need some further elaboration. At first, it seems that the introduction of facet unknowns in HDG obfuscate the formulations, and increases the problem size. However, the element field, ϕ_h in Eq. (2.16), can be expressed in terms of the facet unknowns, $\bar{\phi}_h$. This *static condensation* process, whereby the local variables are eliminated in favor of the global flux variables, results in a global system which is only slightly larger - for an HDG trace space - or of the same size - for an EDG trace space - as a CG formulation. At the same time, HDG inherits many of the attractive properties of the DG method. Particularly relevant to mention is that a HDG formulation for the incompressible Navier-Stokes with pointwise divergence-free velocity fields has been proposed recently [8, 72]. In addition, expressing the flux variable in terms of additional facet unknowns offers increased flexibility in defining the numerical fluxes. This feature will be explored later on in this thesis to ensure local conservation properties in particle-mesh schemes, and much of the developments presented in this thesis therefore hinge on concepts provided by a hybridized discontinuous Galerkin framework.

2.2.6. FEM and variational formulations

The motivation for choosing the finite element method as the method of choice becomes clear by interpreting FEM as a subclass of *variational techniques*. In fact, the FEM in fact has evolved from this branch of mathematics known as *variational calculus*, see e.g. [73, 74] for some historic comments. As an interesting aside: both Euler and Lagrange - indeed, the same ones - made significant contributions to this field.

To illustrate the relation between FEM and variational formulations, consider the minimization problem [15]

$$\min_{\phi \in Q} J(\phi) = \int_{\Omega} \nabla \phi \cdot \nabla \phi \, d\Omega - \int_{\Omega} f \phi \, d\Omega. \quad (2.20)$$

The central idea in Lagrange's variational approach is to perturb the dependent function ϕ in Eq. (2.20) with arbitrary variations $\epsilon\delta\phi$, and using that this perturbed objective function has to be minimal for all functions $\delta\phi$ at $\epsilon = 0$. In other words, the derivative of the perturbed objective function with respect to ϵ must be zero at $\epsilon = 0$, i.e.

$$\left. \frac{\partial J(\phi + \epsilon\delta\phi)}{\partial \epsilon} \right|_{\epsilon=0} = 0. \quad (2.21)$$

Following this approach, Eq. (2.14) is obtained after some straightforward manipulations - including the replacement of ϕ by the finite dimensional trial functions ϕ_h and $\delta\phi$ by the test functions q_h .

Perturbing an objective function with arbitrary variations, computing a directional derivative, and applying integration by parts where needed, is the 'central highway of variational calculus' [74]. This branch of mathematics is key to Lagrangian mechanics, forms loosely speaking the fundamental principle underlying the finite element method, and is the basis to solve a wide range of (un)constrained optimization problems. In this research, variational techniques are used in the latter two contexts. More specifically, variational techniques not only form the basis for the finite element solver in the mesh part, but will be used also in an 'optimization context' to formulate the particle-mesh projections, which is to be investigated next.

2.3. Particle-mesh interactions

In Section 2.1 the particle-mesh approach was interpreted as a special type of operator splitting scheme, in which the particle-mesh and the mesh-particle transfer act as projections between the particle space and the mesh space and *vice versa*. Moreover, the particle space consists of a collection of infinitesimal point particles which are samples to the continuum, Section 2.1.2, and using the finite element method introduced in Section 2.2 the mesh space is interpreted in terms of finite dimensional function spaces. Equipped with this knowledge, all ingredients are in place to sketch the contours of a generic interpretation for the particle-mesh projections.

2.3.1. A variational framework for the projections

One way or the other, the variational methods introduced in Section 2.2.6 attempt to minimize an objective function. A similar viewpoint may be adopted regarding the particle-mesh projections: what the particle-mesh and the mesh-particle projection essentially should do, is to minimize the misfit between a particle field and a mesh field and *vice versa*. Returning to the abstract particle-mesh operator splitting from Eq. (2.5) for the sake of illustration, the objective of the mesh-particle projection \mathcal{P}_L is to minimize the difference between the initial condition φ_0 at the mesh and a particle field, indicated by ψ_p , i.e.

$$\min_{\psi_p} J = \sum_p \frac{1}{2} (\varphi_0(x_p) - \psi_p(0))^2, \quad (2.22)$$

in which x_p is the spatial coordinate of particle p . Conversely, the particle-mesh projection can be formulated based on the same objective function, but using the mesh field

ϕ as the minimizer instead, i.e.

$$\min_{\phi} J = \sum_p \frac{1}{2} (\phi(x_p, 0) - \psi_p(t^1))^2 \quad (2.23)$$

Applying the idea of taking variations and computing the directional derivative, yields the following condition for the minimum of Eq. (2.23)

$$\delta J = \sum_p (\phi(x_p, 0) - \psi_p(t^1)) \delta \phi = 0 \quad (2.24)$$

where this equation must hold for arbitrary variations $\delta \phi(x)$.

To make this idea meaningful, one additional step needs to be taken. The interpretation of the particle field as being composed of point values is clear, but what about the mesh field variables and the variations $\delta \phi$, which up till now are formulated in the infinite dimensional solution space? At this point, ideas from the Galerkin FEM method can be exploited by replacing the continuous functions ϕ by discrete counterparts $\phi_h \in Q_h$, where Q_h is an as yet undefined but finite dimensional function space.

By assuming that functions ϕ_h can be expanded as

$$\phi_h = \sum_{i=1}^n N_i(x) \phi_i, \quad (2.25)$$

in which N_i the basis function and ϕ_i is the value of the mesh field at a node i . Similarly, following the Bubnov-Galerkin approach, the variations are given by

$$\delta \phi \approx w_h = \sum_{i=1}^n N_i(x) w_i, \quad (2.26)$$

in which w_i the weights associated with the test function.

Upon substitution in Eq. (2.24), and noting that the expression for the minimum must hold for arbitrary variations, the following expression for the particle-mesh projection \mathcal{P}_E is obtained

$$\sum_p N_i(x_p) N_j(x_p) \phi_j = \sum_p N_i(x_p) \psi_p. \quad (2.27)$$

Applying the same variational procedure to the mesh-particle projection \mathcal{P}_L in Eq. (2.22), leads to

$$\psi_p = \sum_i N_i(x_p) \varphi_j, \quad (2.28)$$

where it remains to specify the basis functions N_i .

Interpreting the particle-mesh projections as the projection from a particle space to a mesh space and *vice versa*, the choice for the basis functions is prescribed by the FEM discretization used at the mesh. That is, the particle-mesh projection projects the particle field onto the function space employed for the FEM discretization of the mesh-based equations, and the mesh-particle projection projects the mesh-based solution

onto the particle space. Hence, the basis functions for the particle-mesh projections are judiciously taken from the same (Lagrange polynomial) function space which is employed for the FEM discretization at the mesh.

The intertwining of the projections with the mesh step thus becomes apparent. Indeed, it is possible to deviate from this, but only at the expense of introducing additional projection operators between different finite dimensional function spaces, or departing from the variational approach altogether by, e.g., opting for a FV or FD discretization of the equations at the mesh. These paths are not further investigated in this thesis, and the advantage of employing the same finite dimensional function space throughout the particle-mesh projections and the mesh-based discretization is appreciated and will be further explored.

To further limit the scope, consider the consequences on the particle-mesh projection by choosing either piecewise continuous basis functions as in CG methods versus piecewise discontinuous basis functions as in a DG method. For a piecewise continuous basis, the particle-mesh projection Eq. (2.27) entails a global problem, in which a global matrix $N_i(x_p)N_j(x_p)$ has to be inverted. By choosing a piecewise discontinuous basis, the particle-mesh projection can be rewritten as a summation of local problems over each cell separately, and requires the inversion of small matrices only. This is typically much faster than solving one large system. As such, a (H)DG framework starts to appear as an attractive approach for formulating particle-mesh schemes for advection dominated flows, the more so given the rigorous mathematical and physical basis underpinning (H)DG methods for these type of problems as discussed earlier.

2.3.2. Consistency of projections

The interpretation of the particle-mesh interactions as projection operators for which a (discontinuous) FEM framework seems particularly convenient, appears trivial. However, it readily leads to an important result concerning consistency of the mutual projections.

Consider to this end the subsequent application of the mesh-particle projection \mathcal{P}_L on a mesh field, say $\phi_h^0 \in Q_h$, followed by the particle-mesh projection \mathcal{P}_L . Using Eqs. (2.27,2.28) this composition of projections $\mathcal{P}_E \circ \mathcal{P}_L(\phi_h(0))$ results in

$$N_i(x_p)N_j(x_p)\phi_j^1 = N_i(x_p) (N_j(x_p)\phi_j^0), \quad (2.29)$$

$$\phi_j^1 = (N_i(x_p)N_j(x_p))^{-1} N_i(x_p)N_j(x_p)\phi_j^0 \quad (2.30)$$

in which summation over repeated indices is implied. Assuming that $N_j(x_p)$ has a left-inverse - which requires $p \geq j$, and column rank $r = j$ - this will yield $\phi_j^1 = \phi_j^0$, and hence under the aforementioned premise it holds

$$\phi_h^1 = \mathcal{P}_E \circ \mathcal{P}_L(\phi_h^0) = \phi_h^0, \quad (2.31)$$

i.e. $\phi_h^1 \in Q_h$ reconstructs ϕ_h^0 exactly for fixed particle positions. Hence, in the absence of advection, the back-and-forth projection of quantities between particles and the mesh does not interfere with the mesh-based solver, provided $p \geq j$, and $x_p \neq x_P$ for $P = \{1, 2, \dots, N_p : P \neq p\}$. The latter conditions pose practical restrictions upon the required number of particles per cell and the particle distribution as will become clear later.

By assuming for the time being that these conditions are met, the scheme will behave for pure diffusion problems as if there are no particles at all. This is further illustrated for a simple example in Section 2.4. Arguably, this consistency property of the back-and-forth projections is a prerequisite for any particle-mesh method to avoid spurious artifacts resulting from the particle-mesh data transfer operations, and contrasts the perception in that the two projections are separate steps for which one is free to make different choices, see e.g. [26, 35, 41].

2.4. Example: 1D diffusion of Gaussian

As a minimal, yet illustrative example for the more complicated problems in later chapters, the diffusion of a Gaussian pulse is considered on a one-dimensional spatial domain $\Omega := [-1, 1]$, and time interval $I = (0, T]$. Particles are included to illustrate the consistency criterion as derived in the preceding section.

The problem of interest reads

$$\frac{\partial \phi}{\partial t} = \kappa \frac{\partial^2 \phi}{\partial x^2} \quad \text{in} \quad \Omega \times I, \quad (2.32a)$$

$$\phi(x, 0) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad \text{in} \quad \Omega, \quad (2.32b)$$

complemented with homogeneous Neumann boundary conditions

$$\frac{\partial \phi}{\partial x} = 0 \quad \text{at} \quad x = \pm 1, \quad (2.32c)$$

where κ a diffusion coefficient and σ the variance of the Gaussian.

The following two solution strategies are considered:

- **Case 1:** a purely mesh-based discretization of Eq. (2.32a), using a discontinuous Galerkin discretization in space and a backward Euler discretization in time.
- **Case 2:** the same mesh and time discretization is used as in Case 1, but now a particle-mesh projection (Eq. (2.27)) prepends, and a mesh-particle projection appends the mesh-based discretization of the diffusion equation. Partitioning the time interval of interest I into a sequence of $N + 1$ discrete time levels $\{t^0, t^1, \dots, t^N\}$, each time step thus comprises the following steps for each subinterval $I_n = (t^n, t^{n+1}]$:
 1. Reconstruct a mesh field, indicated by $\phi_h^{*,n}$, from a particle field, say ψ_p^n by virtue of the particle-mesh projection, i.e. $\phi_h^{*,n} = \mathcal{P}_E(\psi_p^n)$.
 2. Solve the diffusion equation at the mesh, where $\phi_h^{*,n}$ provides the condition for advancing the time step $n \rightarrow n + 1$.
 3. Update the particle field to ψ_p^{n+1} given the mesh solution ϕ_h at time level $n + 1$, using the mesh-particle projection, i.e. $\psi_p^{n+1} = \mathcal{P}_L(\phi_h^{n+1})$.

Using particles in Case 2 does not bear any advantages for the diffusion equation in which advection is absent. Therefore, the ideal scenario is that results obtained for Case 2 are identical to Case 1.

To perform the spatial discretization of the diffusion equation, consider the discontinuous function space

$$W_h := \{w_h \in L^2(\mathcal{T}), w_h|_K \in P_1(K) \forall K\}, \quad (2.33)$$

in which P_1 indicates that piecewise linear polynomials are used as a basis for the DG approximation on cell K , see Figure 2.3b.

Furthermore, numerical flux definitions as in the *interior penalty* method are used, i.e. [68]:

$$\hat{\phi}_h = \langle \phi_h \rangle, \quad (2.34)$$

$$\hat{\sigma}_h = -\langle \kappa \nabla \phi_h \rangle + \kappa \frac{\alpha}{h_e} \llbracket \phi_h \rrbracket, \quad (2.35)$$

in which α a penalty parameter, which is set equal to 4, and h_e a characteristic mesh size which will be set equal to Δx in this 1D example. Furthermore, the jump operator $\llbracket \cdot \rrbracket$ and average operator $\langle \cdot \rangle$ are defined on interior facets as

$$\llbracket \varphi \rrbracket = \varphi^+ n^+ + \varphi^- n^-, \quad (2.36)$$

$$\langle \varphi \rangle = \frac{1}{2} (\varphi^+ + \varphi^-). \quad (2.37)$$

A discrete implementation of the 1D diffusion problem, Eq. (2.32), is obtained by rewriting the cell boundary integrals in the time-dependent counterpart of form Eq. (2.16) in terms of integrals over the facets, and substituting Eq. (2.34) and Eq. (2.35) for the numerical fluxes [68, 75]. After applying a backward Euler time integration, the following fully-discrete form results: given $\phi_h^n \in W_h$ (Case 1) or $\phi_h^{*,n} \in W_h$ (Case 2), and uniform cell size Δx and step size Δt , find $\phi_h^{n+1} \in W_h$ such that

$$\begin{aligned} & \int_{\Omega} \frac{\phi_h^{n+1} - \phi_h^{(*,n)}}{\Delta t} w_h \, d\Omega + \int_{\Omega} \kappa \frac{\partial \phi_h^{n+1}}{\partial x} \frac{\partial w_h}{\partial x} \, d\Omega - \sum_{F \in \mathcal{F}} \oint_{F \setminus \partial\Omega} \kappa \llbracket \phi_h^{n+1} \rrbracket \langle \frac{\partial w_h}{\partial x} \rangle \, d\Gamma \\ & - \sum_{F \in \mathcal{F}} \oint_{F \setminus \partial\Omega} \kappa \langle \frac{\partial \phi_h^{n+1}}{\partial x} \rangle \llbracket w_h \rrbracket \, d\Gamma + \sum_{F \in \mathcal{F}} \oint_{F \setminus \partial\Omega} \kappa \frac{\alpha}{\Delta x} \llbracket \phi_h^{n+1} \rrbracket \llbracket w_h \rrbracket \, d\Gamma = 0 \quad \forall w_h \in W_h. \end{aligned} \quad (2.38)$$

For Case 2, in which the particles carry the solution between consecutive time steps, the reconstruction of the mesh field $\phi_h^{*,n}$ from the particle field ψ_p^n can be written as

$$\phi_h^{*,n} = \mathcal{P}_E(\psi_p^n) = \mathcal{P}_E \circ \mathcal{P}_L(\phi_h^n) = \phi_h^n, \quad (2.39)$$

where this results from Eq. (2.30). And so Case 1 and Case 2 are expected to be exactly identical for the example considered.

To illustrate this, the domain $\Omega = [-1, 1]$ is partitioned into 40 equally sized cells, so that $\Delta x = 0.05$. The diffusivity κ in Eq. (2.32a) is set to $5e-3$ and the variance σ in Eq. (2.32b) is set to $\sigma = 0.125$. The discrete time step Δt is set to 0.1. For Case 2,

100 particles are randomly seeded in the domain of interest, in such a way that every cell K contains at least two particles at disjoint locations. Results for Case 1 and Case 2 at $t = 5$ for this model set-up are depicted in Fig. 2.6. Visually, there is no difference between both cases, and for Case 2 the particle values coincide with the corresponding mesh solution. These observations are quantitatively confirmed by assessing the L^2 error $\|\phi_h(5) - \phi(x, 5)\|$ and the area error

$$\epsilon_{\Delta\phi\Omega} = \frac{\int_{-1}^1 [\phi_h(x, 5) - \phi_h(x, 0)] dx}{\int_{-1}^1 \phi_h(x, 0) dx} \quad (2.40)$$

Results for these two errors are reported for Case 1 and Case 2 in Table 2.1. As expected, Case 2 returns exactly the same results as Case 1, which illustrates that the projection operators do not affect the solution of the diffusion step at the mesh in the absence of (particle) advection, which owes to the consistency between the particle-mesh and the mesh-particle projections. As an aside, with the area errors being zero to machine precision the global conservation property of the DG discretization is illustrated. For the pure diffusion problem in which particles have a fixed position, this property is inherited by the particle-mesh scheme.

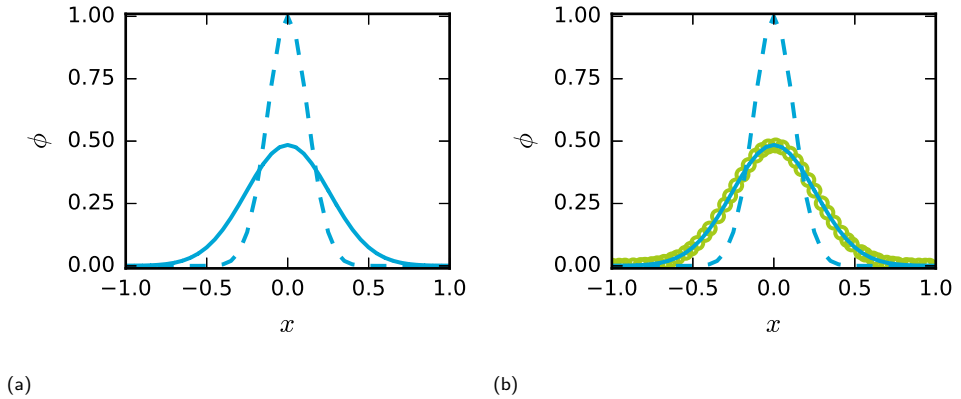


Figure 2.6: Diffusion of Gaussian: solid line is the computed mesh solution at $t = 5$ for (a) Case 1 (mesh-based) and (b) Case 2 (particle-mesh). For the particle-mesh approach (b), the particle solution is indicated with \circ for every other particle. The dashed line indicates the initial condition.

Table 2.1: Diffusion of Gaussian: L^2 -error and mass conservation error for mesh-based case (Case 1) and particle-mesh scheme (Case 2) at time $T = 2$.

	$\ \phi_h(5) - \phi(x, 5)\ $	$\epsilon_{\Delta\phi\Omega}$
Case 1	8.21059248925e-4	1.42e-15
Case 2	8.21059248926e-4	8.85e-16

2.5. Conclusion

This chapter introduced the various steps comprising particle-mesh schemes. It was argued that these methods are a special type of operator splitting in which advection - the kinematic part of the problem - is conducted at the particles, and a mesh is used for the discretization of the constitutive part. Two data transfer steps, i.e. the particle-mesh projection and the mesh-particle projection, are required to bridge the gap between the particles and the mesh. As such, these transfer operations are to be interpreted as projection steps, mapping information from the 'particle space' to the 'mesh space' and *vice versa*. For seamlessly integrating these projections with the mesh step, a combination of variational techniques for the projections in conjunction with a finite element (FE) discretization of the mesh part appears to be particularly apt. The fundamentals of these two building blocks were introduced.

The mutually consistent formulation for the particle-mesh and the mesh-particle projection is a first important result of this variational framework for particle-mesh schemes. This consistency implies, among others, that in the absence of advection, particles do not lead to spurious side-effects, and properties of the mesh-based solver are inherited by the particle-mesh scheme. A one-dimensional diffusion problem further illustrated this property. In addition, it was shown that the discrete function space employed throughout the mesh discretization dictates the implementation of the particle-mesh projection operators. This observation makes clear that the projection steps and the mesh step are essentially intertwined. Also, it renders a (hybridized) discontinuous Galerkin (H)DG framework appealing, since this allows to perform the particle-mesh projections in a cellwise fashion. The unique features of (H)DG schemes are further exploited and extended to formulate high-order accurate and conservative particle-mesh discretizations for the advection-diffusion and the incompressible Navier-Stokes equations in subsequent chapters.

3

Scalar transport

This chapter presents a particle-mesh scheme for the scalar advection-diffusion equation. This allows focusing on the intricacies related to the particle-mesh projections in the presence of advection, before moving to the more complicated incompressible Navier-Stokes equations in the next chapter. The main part of this chapter is therefore dedicated to developing and mathematically analyzing the particle-mesh projections, for which accurate and locally conservative formulations are presented. The relatively uncomplicated structure of the advection-diffusion equation allows experimenting and testing the scheme for a range of problems, covering pure advection problems as well as diffusion-dominated problems.

It would be an underestimation of the relevance of the scalar advection-diffusion equation to value it as a building block for more complicated equations only. Instead, this equation itself is of practical relevance to a number of practical applications, most notably problems involving the transport of suspended or dissolved matter, e.g. sediment transport problems, the dispersal of pollutants in the atmosphere or the transport of groundwater contaminants. In the scope of this thesis it is particularly the scalar advection problem which is of importance as it will be used in a later chapter for advecting the density fields in immiscible multiphase flows.

3.1. Model problem

Consider a domain of interest $\Omega \subset \mathbb{R}^d$ (with $d = 2, 3$), having a Lipschitz continuous boundary $\Gamma = \partial\Omega$. The boundary Γ is partitioned into complementary Dirichlet and Neumann boundaries Γ_D and Γ_N , such that $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$. The

Material presented in this chapter is partly based on:

[76]: Optimization Based Particle-Mesh Algorithm for High-Order and Conservative Scalar Transport, J.M. Maljaars et al. In: *Lecture Notes in Computational Science and Engineering*, **132**, 2019.

[77]: Conservative, high-order particle-mesh scheme with applications to advection-dominated flows, J.M. Maljaars et al. In: *Computer Methods in Applied Mechanics and Engineering*, **348**, 443-465, 2019.

outward pointing unit vector normal to Γ is denoted by \mathbf{n} . The time interval of interest is $I = (t^0, t^N]$, where t^0 and t^N are the start and end time of the simulation.

3.1.1. Governing equations

On the space-time domain $\Omega \times I$, the scalar-valued linear advection-diffusion equation is defined as a system of two first-order equations as follows: given a solenoidal velocity field $\mathbf{a} : \Omega \times I \rightarrow \mathbb{R}^d$, an initial condition $\phi_0 : \Omega \rightarrow \mathbb{R}$, diffusivity κ , and boundary conditions $h : \Gamma_N \times I \rightarrow \mathbb{R}$ and $g : \Gamma_D \times I \rightarrow \mathbb{R}$, find the scalar quantity $\phi : \Omega \times I \rightarrow \mathbb{R}$ such that

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \boldsymbol{\sigma} = f \quad \text{in } \Omega \times I, \quad (3.1a)$$

$$\boldsymbol{\sigma} = \mathbf{a}\phi - \kappa \nabla \phi \quad \text{in } \Omega \times I, \quad (3.1b)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = (1 - \gamma)(\mathbf{a} \cdot \mathbf{n})\phi + h \quad \text{on } \Gamma_N \times I, \quad (3.1c)$$

$$\phi = g \quad \text{on } \Gamma_D \times I, \quad (3.1d)$$

$$\phi(\mathbf{x}, t^0) = \phi_0 \quad \text{in } \Omega. \quad (3.1e)$$

The parameter γ in Eq. (3.1c) is equal to one at inflow boundaries (where $\mathbf{a} \cdot \mathbf{n} < 0$) and equal to zero on outflow Neumann boundaries (where $\mathbf{a} \cdot \mathbf{n} \geq 0$), with h specifying the total flux on inflow Neumann boundaries and the diffusive flux on outflow Neumann boundaries.

3.1.2. Operator splitting

As argued in Chapter 2, the particle-mesh method is conceived as a spatiotemporal operator splitting procedure. To this end, let the time interval of interest I be partitioned using a sequence of $N + 1$ discrete time levels $\{t^0, t^1, \dots, t^{N-1}, t^N\}$ which for $n = 0, N - 1$ define the half-open subintervals $I_n = (t^n, t^{n+1}]$ such that $\bigcup_n I_n = I$, while $\mathcal{I} := \{I_n\}$ defines the ordered sequence of subintervals. Furthermore, the total flux, given by Eqs. (3.1b), is decomposed into an advective part $\boldsymbol{\sigma}_a$ and a diffusive part $\boldsymbol{\sigma}_d$.

A spatiotemporal operator splitting procedure for the advection-diffusion problem, Eq. (3.1), now involves a scalar field $\psi : \Omega \times I_n \rightarrow \mathbb{R}$ satisfying an advection problem,

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \boldsymbol{\sigma}_a = 0 \quad \text{in } \Omega \times I_n, \quad (3.2a)$$

$$\boldsymbol{\sigma}_a = \mathbf{a}\psi \quad \text{in } \Omega \times I_n, \quad (3.2b)$$

$$\boldsymbol{\sigma}_a \cdot \mathbf{n} = (1 - \gamma)(\mathbf{a} \cdot \mathbf{n})\psi + \gamma h_a \quad \text{on } \Gamma_N \times I_n, \quad (3.2c)$$

$$\psi = g \quad \text{on } \Gamma_D^- \times I_n, \quad (3.2d)$$

$$\psi(\mathbf{x}, t^n) = \mathcal{P}_L(\phi(\mathbf{x}, t^n)) \quad \text{in } \Omega, \quad (3.2e)$$

and a scalar field $\phi : \Omega \times I_n \rightarrow \mathbb{R}$ satisfying a diffusion equation,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \boldsymbol{\sigma}_d = f \quad \text{in } \Omega \times I_n, \quad (3.3a)$$

$$\boldsymbol{\sigma}_d = -\kappa \nabla \phi \quad \text{in } \Omega \times I_n, \quad (3.3b)$$

$$\boldsymbol{\sigma}_d \cdot \mathbf{n} = h_d \quad \text{on } \Gamma_N \times I_n, \quad (3.3c)$$

$$\phi = g \quad \text{on } \Gamma_D \times I_n, \quad (3.3d)$$

$$\phi(\mathbf{x}, t^n) = \mathcal{P}_E(\psi(\mathbf{x}, t^{n+1})) \quad \text{in } \Omega. \quad (3.3e)$$

to be applied *sequentially* for every $I_n \in \mathcal{I}$. In the advection stage the Dirichlet boundary condition can only be prescribed at inflow Dirichlet boundaries, denoted with Γ_D^- . The flux prescribed on Γ_N is split additively into an advective flux h_a and a diffusive flux h_d , to be applied in the advective and diffusive step of the procedure, respectively. Note that the advective flux cannot be specified at outflow Neumann boundaries, which is automatically taken care of by virtue of Eq. (3.2c). Furthermore, \mathcal{P}_L and \mathcal{P}_E are projection operators, which are introduced in order to couple the fields ψ and ϕ , with these fields being naturally defined on the particles and the mesh, respectively, in PIC methods. More precisely, the projection operator \mathcal{P}_L provides the initial condition at t^n to advance the Lagrangian advection problem to t^{n+1} , and the projection operator \mathcal{P}_E provides the initial condition at t^n to advance the Eulerian diffusion problem to t^{n+1} . As discussed in Section 2.1.2, it is required that the compositions $\mathcal{P}_E \circ \mathcal{P}_L$ and $\mathcal{P}_L \circ \mathcal{P}_E$ equal the identity operator in order to remain consistent with Eq. (3.1).

3.1.3. Auxiliary definitions

Function spaces

The Eulerian mesh is the triangulation $\mathcal{T} := \{K\}$ of Ω into open, non-overlapping cells K . A measure of the cell size is denoted by h_K , and the outward pointing unit normal vector on the boundary ∂K of a cell is denoted by \mathbf{n} . Adjacent cells K_i and K_j ($i \neq j$) share a common facet $F = \partial K_i \cap \partial K_j$. The set of all facets (including the exterior boundary facets $F = \partial K \cap \partial \Omega$) is denoted by \mathcal{F} .

The following scalar finite element spaces are defined on \mathcal{T} and \mathcal{F} :

$$W_h := \{w_h \in L^2(\mathcal{T}), w_h|_K \in P_k(K) \forall K \in \mathcal{T}\}, \quad (3.4)$$

$$T_h := \{\tau_h \in L^2(\mathcal{T}), \tau_h|_K \in P_l(K) \forall K \in \mathcal{T}\}, \quad (3.5)$$

$$\bar{W}_h := \{\bar{w}_h \in L^2(\mathcal{F}), \bar{w}_h|_F \in P_k(F) \forall F \in \mathcal{F}\}, \quad (3.6)$$

in which $P(K)$ and $P(F)$ denote the spaces spanned by Lagrange polynomials on K and F , with subscripts $k \geq 1$ and $l \geq 0$ indicating the polynomial orders. The trace space $\bar{W}_{h,g}$ contains single-valued functions that are piecewise continuous on facets $F \in \mathcal{F}$. Furthermore, the facet function space $\bar{W}_{h,g}$ satisfies the inhomogeneous Dirichlet boundary condition on Γ_D , with the related space $\bar{W}_{h,0}$ satisfying the homogeneous Dirichlet boundary condition on Γ_D . Functions in W_h and T_h are formally defined only at the cell boundary ∂K via a trace operator, but this technicality is omitted in the sequel to avoid notational clutter.

Particle definitions

Let χ_t define the configuration of Lagrangian particles in the domain Ω at a time instant t , i.e.

$$\chi_t := \{\mathbf{x}_p(t) \in \Omega\}_{p=1}^{N_p}, \quad (3.7)$$

in which \mathbf{x}_p denotes the spatial coordinates of particle p , and N_p is the number of particles. Furthermore, the index set of all particles and the index set of particles hosted by cell K , at a fixed time instant t are defined as

$$\mathcal{S}_t := \{p \in \mathbb{N} : \mathbf{x}_p(t) \in \mathcal{X}_t\}, \quad (3.8)$$

$$\mathcal{S}_t^K := \{p \in \mathbb{N} : \mathbf{x}_p(t) \in \overline{K}, \mathbf{x}_p(t) \in \mathcal{X}_t\}. \quad (3.9)$$

A scalar field on the Lagrangian particles is denoted by

$$\Psi_t := \{\psi_p(t) \in \mathbb{R}\}_{p=1}^{N_p}, \quad (3.10)$$

where ψ_p denotes the scalar quantity associated with particle p .

Importantly, henceforth Lagrangian particle data and Eulerian mesh fields are consistently distinguished by using the subscripts p and h .

3.1.4. Solution strategy and outline

Key to the presented particle-mesh method is the splitting of the semi-discrete problem into a Lagrangian step and an Eulerian step. This strategy requires a transfer of information from the Eulerian mesh to the Lagrangian particles and *vice versa* by means of two projection steps, i.e. Eq. (3.2e) and Eq. (3.3e). These auxiliary steps involve the reconstruction of a scalar field at the mesh from the scattered particle data, and the update of the scalar values carried by the particles, given the solution at the mesh.

In summary, the proposed particle-mesh method has the following principal components:

1. *Lagrangian discretization of the advection problem*; in order to solve Eqs. (3.2a-3.2d).
2. *Eulerian discretization of the diffusion problem*; in order to solve Eqs. (3.3a-3.3d).
3. *Particle-mesh projection*; in order to reconstruct a scalar field at the Eulerian mesh from the scattered particles in order to advance the diffusion problem (Eq. (3.3e)).
4. *Mesh-particle projection*; given the solution at the Eulerian mesh, update the scalar field defined at the particles in order to advance the advection problem in time (Eq. (3.2e)).

Section 3.2 briefly presents semi-discrete formulations for components 1 and 2. The main endeavor of this chapter, however, is to present strategies for the particle-mesh projection - component 3, which projects data from the particles onto the mesh - and the mesh-particle projection - component 4, which takes care of the projection of mesh fields onto the particles. Semi-discrete formulations for these steps are presented in Section 3.3. In Section 3.4 the properties of the semi-discrete formulation are further analyzed, in which particular attention is paid to the projection steps, with the novel formulations for these steps being crucial to bridge the gap between a Lagrangian particle-based and an Eulerian mesh-based approach.

The presentation for the semi-discrete setting slightly departs from the blueprint of particle-mesh schemes in Fig. 1.4, with a view to presenting a particular fully-discrete solution procedure in Section 3.5. Algorithmic aspects and implementation considerations are briefly discussed in Section 3.6.

3.2. Semi-discrete formulations for the particle- and mesh sub-problems

3.2.1. Lagrangian discretization of the advection problem

In a Lagrangian, particle-based frame of reference, the advection problem Eqs. (3.2a-3.2d) is decomposed into two ordinary differential equations (ODEs) for the particle scalar quantity and the particle position. On the time interval I_n , these equations are given by

$$\dot{\psi}_p(t) = 0 \quad \forall p \in \mathcal{S}_t, \quad (3.11a)$$

$$\dot{\mathbf{x}}_p(t) = \mathbf{a}(\mathbf{x}_p(t), t) \quad \forall p \in \mathcal{S}_t, \quad (3.11b)$$

where $\dot{\psi}_p(t)$ and $\dot{\mathbf{x}}_p(t)$ are the total derivatives at time $t \in I_n$ of the scalar quantity and the position of particle p . Furthermore, $\mathbf{a}(\mathbf{x}, t^n)$ is a prescribed solenoidal velocity field at time t^n . An important observation is that ψ_p stays constant throughout the particle advection stage by virtue of Eq. (3.11a).

3.2.2. Eulerian discretization of the diffusion problem

We next present the semi-discrete formulation of the diffusion problem Eq. (3.3), using a hybridized discontinuous Galerkin (HDG) approach (see Section 2.2.4 for a short introduction to HDG). This HDG discretization strategy is pursued in view of integration with the particle-mesh projections, to be introduced in Section 3.3, and specifically, the HDG formulation presented by Labeur & Wells [3] is used to seek solutions $\phi_h \in W_h$. Referring to [3] for further details, the HDG discretization results in a set of local and global problems which are respectively stated as: at time $t \in I_n$, given the initial condition $\phi_h^n \in W_h$, the diffusive Neumann boundary condition $h_d : \Gamma_N \rightarrow \mathbb{R}$ and the diffusivity κ , find $\phi_h \in W_h$ and $\bar{\phi}_h \in \bar{W}_{h,g}$ such that

$$\begin{aligned} \int_{\Omega} \frac{\partial \phi_h}{\partial t} w_h \, d\Omega + \sum_K \int_K \kappa \nabla \phi_h \cdot \nabla w_h \, d\Omega + \sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h} \cdot \mathbf{n} w_h \, d\Gamma \\ + \sum_K \oint_{\partial K} \kappa (\bar{\phi}_h - \phi_h) \mathbf{n} \cdot \nabla w_h \, d\Gamma = \int_{\Omega} f w_h \, d\Omega \quad \forall w_h \in W_h, \end{aligned} \quad (3.12a)$$

and

$$\sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h} \cdot \mathbf{n} \bar{w}_h \, d\Gamma = \int_{\Gamma_N} h_d \bar{w}_h \, d\Gamma \quad \forall \bar{w}_h \in \bar{W}_{h,0}, \quad (3.12b)$$

where $\hat{\sigma}_{d,h}$ is a diffusive flux at cell facets defined by

$$\hat{\sigma}_{d,h} = -\kappa \nabla \phi - \frac{\alpha}{h_K} \kappa (\bar{\phi}_h - \phi_h) \mathbf{n}, \quad (3.12c)$$

in which α is a dimensionless parameter as is typical to interior penalty methods [68], and h_K is a measure of the cell size. Solving Eq. (3.12) yields $\bar{\phi}_h$ and ϕ_h , where the latter is to be used for updating the particle properties ψ_p in the mesh-particle projection step.

3.3. Semi-discrete formulation of the projection steps

The preceding section presented the Lagrangian and the Eulerian steps as separate components. Crucial is to couple these steps in an efficient, accurate and physically consistent way. To this end, two particle-mesh projections are proposed in this section. Firstly, an accurate and efficient implementation of the variational approach from Section 2.3.1 is presented. Using this strategy, discrete conservation properties are lost when the particle field is advected. Therefore, an extension of this approach which retains global and local conservation properties is presented in Section 3.3.2. Section 3.3.3 presents a strategy for the mesh-particle projection closely following the variational approach of the particle-mesh projection.

3.3.1. Particle-mesh projection via ℓ^2 projections

This section presents a first strategy to project the particle properties (i.e. the scalar-valued field ψ_p carried by the particles) from the scattered set of Lagrangian particles onto W_h , that is, the space of piecewise-continuous basis functions defined on the mesh. To this end, an ℓ^2 -projection is adopted to formulate the operator $\mathcal{P}_E : \Psi_t \rightarrow W_h$, by minimizing the objective function

$$\min_{\psi_h \in W_h} J := \sum_{p \in \mathcal{S}_t} \frac{1}{2} (\psi_h(\mathbf{x}_p(t), t) - \psi_p(t))^2, \quad (3.13)$$

where $\psi_p(t) \in \Psi_t$ and $\mathbf{x}_p(t) \in \mathcal{X}_t$ are the velocity and position of particle p at time instant t , respectively.

Equating the variations δJ with respect to the variations $\delta \psi_h$ to zero leads to

$$\sum_{p \in \mathcal{S}_t} (\psi_h(\mathbf{x}_p(t), t) - \psi_p(t)) \cdot \delta \psi_h(\mathbf{x}_p(t)) = 0 \quad \forall \delta \psi_h \in W_h. \quad (3.14)$$

This equation can be written as a summation of local equations on cells $K \in \mathcal{T}$: at time t , given the particle scalar value $\psi_p(t)$, find $\psi_h(t) \in W_h$ such that

$$\sum_K \sum_{p \in \mathcal{S}_t^K} (\psi_h(\mathbf{x}_p(t), t) - \psi_p(t)) \cdot w_h(\mathbf{x}_p(t)) = 0 \quad \forall w_h \in W_h, \quad (3.15)$$

in which the variations $\delta \psi_h$ are replaced by the local test functions $w_h \in W_h$.

Since $\psi_h, w_h \in W_h$ are discontinuous across cell boundaries, Eq. (3.15) can be solved in

a cellwise fashion. Hence, the mapping of the particle data to the mesh basis functions only involves the inversion of small, local matrices. This approach is expected to give regular and accurate solutions, provided that the particle locations satisfy unisolvency with respect to W_h [78]. In practice, this requires the particle locations to be disjoint and the number of particles in a cell is bounded from below by the number of local basis functions.

On simplicial meshes, the above presented ℓ^2 -projection onto the discontinuous HDG basis is similar to an moving least-squares reconstruction [40, 41] for specific choices of the basis and the particle weighting function in the latter approach, with the notable distinction that the influence of a particle remains strictly local to a cell in the ℓ^2 -projection. This is further detailed in Appendix A.

Two additional remarks concerning the particle-mesh projection close this section:

Remark 1: The function ψ_h which is to be reconstructed from the particles lives in the piecewise discontinuous function space W_h , also used in the diffusion step from Section 3.2.2 for seeking the solution ϕ_h . As a matter of fact, this choice allows for trivial projections between fields ψ_h and ϕ_h .

Remark 2: The minimization problem, Eq. (3.13), can be interpreted as a quadratic programming problem. This class of problems has been thoroughly analyzed in literature, and well-known techniques exist to extend these problems with equality, inequality, and box constraints, see e.g. [79] and references. In the context of the particle-mesh projection, imposing box constraints can be useful to ensure bound-preserving projections. Extending the quadratic objective function with an equality constraint, will be extensively used below for rendering the projection conservative.

3.3.2. PDE-constrained ℓ^2 projection

Starting point to derive a conservative particle-mesh projection in this section, is the notion that the particles are merely used as sampling points of the continuum in the ℓ^2 -projection. Hence, particles by definition do not possess a metric to evaluate integral quantities (e.g. mass or momentum). As illustrated in Section 2.4 for a one-dimensional diffusion example, this does not affect the overall properties of the scheme as long as the particles do not move. However, the whole idea of introducing Lagrangian particles is to conveniently accommodate advection, in which case conservation properties are lost when projecting the advected particle field back to the mesh. One way to preserve these properties is by keeping track of the integral quantities on the mesh. This can be accomplished by constraining the particle-mesh projection to obtain fields ψ_h that satisfy the hyperbolic conservation law, Eq. (3.2).

To this end, the functional in Eq. (3.13) is augmented with terms multiplying the hyperbolic conservation law Eq. (3.2) with a Lagrange multiplier $\lambda_h \in T_h$. Integration by parts leaves an unknown flux on interior cell facets which is formulated in terms of a variable $\tilde{\psi}_h \in \bar{W}_{h,g}$, while the Neumann boundary condition Eq. (3.2c) is substituted on exterior facets. For a given particle field $\psi_p \in \Psi_t$, an advective velocity field $\mathbf{a} : \Omega \times I_n \rightarrow \mathbb{R}^d$, the initial condition $\psi_h^n \in W_h$, and an advective Neumann boundary condition $h_a : \Gamma_N \times I_n \rightarrow \mathbb{R}$, the minimization problem then involves finding the

stationary points of the Lagrangian functional

$$\begin{aligned}
\mathcal{L}(\psi_h, \bar{\psi}_h, \lambda_h) &= \sum_{p \in \mathcal{S}_t} \frac{1}{2} (\psi_h(\mathbf{x}_p(t), t) - \psi_p(t))^2 + \sum_K \oint_{\partial K} \frac{1}{2} \beta (\bar{\psi}_h - \psi_h)^2 d\Gamma \\
&+ \int_{\Omega} \frac{\partial \psi_h}{\partial t} \lambda_h d\Omega - \sum_K \int_K \mathbf{a} \psi_h \cdot \nabla \lambda_h d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \bar{\psi}_h \lambda_h d\Gamma \\
&+ \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \psi_h \lambda_h d\Gamma + \oint_{\Gamma_N} \gamma h_a \lambda_h d\Gamma, \quad (3.16)
\end{aligned}$$

for every $t \in I_n$. The collection of terms containing λ_h constitutes a weak form of the advection subproblem, Eq. (3.2). Furthermore, the unknown facet-based field $\bar{\psi}_h \in \bar{W}_{h,g}$ determines the advective flux at interfaces. The additional regularization term containing $\beta > 0$ penalizes the jumps between ψ_h and $\bar{\psi}_h$ on cell interfaces which avoids the problem to become singular in cases with vanishing normal velocity $\mathbf{a} \cdot \mathbf{n}$.

Before proceeding with the derivation of the resulting optimality system, let's take a brief moment to interpret the Lagrangian, Eq. (3.16). Firstly, the single-valued facet variable $\bar{\psi}_h \in \bar{W}_{h,g}$ takes on the role of control variable. With this ingredient being provided by the facet fields used in the hybridized discontinuous Galerkin (HDG) method, it follows that HDG is a natural choice for imposing the optimality control on the solution. Embedding the optimality system in other, cell-based spatial discretization methods (e.g., finite volumes) is possible as well, however, this would require the definition of an additional interface-based control variable. Secondly, the optimal solution for ψ_h based on Eq. (3.16) will in general not minimize the ℓ^2 -error norm, Eq. (3.13), for the unconstrained and non-conservative case. Indeed, by adding the PDE-constraint we restrict the minimization of this error to the space of physically admissible functions, i.e., those that satisfy the hyperbolic conservation law. Thirdly, the objective function itself is modified compared to Eq. (3.13) by adding the regularization term containing β . The influence of this term on the optimal solution is kept small by choosing the parameter β such that the regularization term is small compared to the first term at the right-hand side in Eq. (3.16).

Equating the variations of Eq. (3.16) with respect to the three unknowns $(\psi_h, \lambda_h, \bar{\psi}_h) \in (W_h, T_h, \bar{W}_{h,g})$ to zero, results in the following system of variational equations. At time $t \in I_n$, variation with respect to the scalar field ψ_h gives the co-state equation

$$\begin{aligned}
&\sum_{p \in \mathcal{S}_t} (\psi_h(\mathbf{x}_p(t), t) - \psi_p(t)) \delta \psi_h(\mathbf{x}_p(t)) - \sum_K \oint_{\partial K} \beta (\bar{\psi}_h - \psi_h) \delta \psi_h d\Gamma + \int_{\Omega} \frac{\partial \delta \psi_h}{\partial t} \lambda_h d\Omega \\
&- \sum_K \int_K \mathbf{a} \cdot \nabla \lambda_h \delta \psi_h d\Omega + \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \lambda_h \delta \psi_h d\Gamma = 0 \quad \forall \delta \psi_h \in W_h.
\end{aligned} \quad (3.17a)$$

Variation with respect to the Lagrange multiplier λ_h gives the state equation,

$$\begin{aligned} \int_{\Omega} \frac{\partial \psi_h}{\partial t} \delta \lambda_h \, d\Omega - \sum_K \int_K \mathbf{a} \psi_h \cdot \nabla \delta \lambda_h \, d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \bar{\psi}_h \delta \lambda_h \, d\Gamma \\ + \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \psi_h \delta \lambda_h \, d\Gamma + \oint_{\Gamma_N} \gamma h_a \delta \lambda_h \, d\Gamma = 0 \quad \forall \delta \lambda_h \in T_h. \end{aligned} \quad (3.17b)$$

Finally, variation with respect to the control variable $\bar{\psi}_h$ leads to the optimality condition,

$$\sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \lambda_h \delta \bar{\psi}_h \, d\Gamma + \sum_K \oint_K \beta (\bar{\psi}_h - \psi_h) \delta \bar{\psi}_h \, d\Gamma = 0 \quad \forall \delta \bar{\psi}_h \in \bar{W}_{h,0}. \quad (3.17c)$$

After an appropriate discretization of the time derivatives in Eqs. (3.17a and 3.17b) - to be discussed in Section 3.5.3 - a field $\psi_h \in W_h$ can be reconstructed from the particle data $\psi_p \in \Psi_t$ by solving the optimality system Eq. (3.17).

3.3.3. Mesh-particle projection

The updating of particle properties given the solution at the mesh is done via the projection $\mathcal{P}_L : W_h \rightarrow \Psi_t$. This mesh-particle projection is based on the following minimization problem

$$\min_{\psi_p(t)} J := \sum_{p \in \mathcal{S}_t} \frac{1}{2} (\phi_h(\mathbf{x}_p(t), t) - \psi_p(t))^2, \quad (3.18)$$

where it is emphasized that the objective functional J is also at the basis of the particle-mesh projection, see Eq. (3.13) and Eq. (3.16). Carrying out the minimization, and equating the functional derivative to zero for arbitrary variations $\delta \psi_p$, yields the particularly simple result for the semi-discrete particle update:

$$\psi_p(t) = \phi_h(\mathbf{x}_p(t), t) \quad \forall p \in \mathcal{S}_t. \quad (3.19)$$

The mesh-particle projection is not restricted to the mapping of the scalar field $\phi_h \in W_h$ itself, but can be applied to project arbitrary fields in W_h - e.g. the temporal increments of ϕ_h , see Section 3.5.5 - onto the particles. As such, it is used for updating the particle quantities, which completes the semi-discrete sequence of steps comprising the particle-mesh operator splitting of the advection-diffusion equation.

3.4. Properties of the semi-discrete formulations

We now demonstrate consistency and global and local conservation of the semi-discrete particle-mesh operator splitting method formulated in Eqs. (3.17), (3.12) and (3.19).

3.4.1. Consistency

As a prerequisite for obtaining a consistent particle-mesh operator splitting scheme for the scalar advection-diffusion equations, certain consistency requirements are to be met by the different model components. This entails three different aspects, (i) the Eulerian part of the operator splitting (step 2) has to be consistent with the diffusion subproblem (Eqs. (3.3)), (ii) the projection operators \mathcal{P}_E (particle-mesh) and \mathcal{P}_L (mesh-particle) have to be mutually consistent, and specific to the PDE-constrained particle-mesh projection (iii), the constraint imposed weakly in the projection operator \mathcal{P}_E (step 3) has to be consistent with the advection subproblem (Eqs. (3.2)).

Concerning (i), consistency of the HDG method used in the diffusion step was proven in [3]. Consistency of the projection operators (ii) implies that in absence of advection the subsequent application of \mathcal{P}_L and \mathcal{P}_E must recover an initially mesh-based field $\psi_h \in W_h$ exactly, which can be expressed mathematically as,

$$\mathcal{P}_E \circ \mathcal{P}_L(\psi_h) = \psi_h \quad \forall \psi_h \in W_h. \quad (3.20)$$

For an arbitrary initial field ψ_h , the mesh-particle projection \mathcal{P}_L formulated in Eq. (3.19) gives $\psi_p = \psi_h(\mathbf{x}_p)$. Employing the local ℓ^2 projections, Section 2.3.2 showed that $\mathcal{P}_E(\psi_p)$ returns ψ_h , with this feature owing to the symmetry in the objective functions underpinning both the particle-mesh and the mesh-particle projection.

For the PDE-constrained particle-mesh interaction, the consistency criterion Eq. (3.20) is verified as follows. Setting $\mathbf{a} = \mathbf{0}$ in the particle-mesh projection $\mathcal{P}_E(\psi_p)$, and substituting $\psi_p = \psi_h(\mathbf{x}_p)$ in the co-state equation, Eq. (3.17a), it follows that in the limit $\beta \downarrow 0$ the initial field ψ_h is recovered exactly, while $\lambda_h = 0$ everywhere. This again owes to the symmetry in the objective functions underpinning the particle-mesh and the mesh-particle projection for vanishing β , see Eq. (3.16) and Eq. (3.18). Since β must be non-zero in order to explicitly couple the state variable ψ_h and the control variable $\bar{\psi}_h$, we choose $\beta > 0$, yet sufficiently small to accurately approximate the consistency criterion on the projection operators.

To prove consistency of the PDE-constraint with the strong form of the advection equation (iii), consider a sufficiently smooth scalar field ψ . Substitution into Eq. (3.17b) gives, after integration by parts,

$$\begin{aligned} \sum_K \int_{\Omega} \left(\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{a}\psi) \right) \delta \lambda_h \, d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} (\bar{\psi}_h - \psi_h) \delta \lambda_h \, d\Gamma \\ - \oint_{\Gamma_N} \gamma \mathbf{a} \cdot \mathbf{n} \psi \delta \lambda_h \, d\Gamma + \oint_{\Gamma_N} \gamma h_a \delta \lambda_h \, d\Gamma = 0 \quad \forall \delta \lambda_h \in T_h, \end{aligned} \quad (3.21)$$

which demonstrates consistency with the strong form of the advection problem Eqs. (3.2a-3.2b) and the Neumann boundary condition Eq. (3.2c), with the enforcement of $\bar{\psi} = \psi$ on interior cell facets and the Dirichlet boundary Γ_D .

3.4.2. Conservation

The main motivation for extending the non-conservative ℓ^2 particle-mesh projection with the PDE-constraint is to guarantee conservation. From a physical point of view, this is

a desirable feature for any numerical scheme since physical quantities, such as mass or momentum, should not get lost as a result of discretization artifacts.

Since the particles carry point evaluations of the underlying field and lack a metric to evaluate mass or volume, conservation can be satisfied at the mesh level only. The latter requires that the mesh-based parts of the scheme, i.e. step 4 (diffusion) and step 2 (advection) are both conservative. Since global and local conservation of the HDG method for the diffusion step was demonstrated in [3], it remains to prove (mass) conservation of the PDE-constrained particle-mesh projection.

Setting $\delta\lambda_h = 1$ in Eq. (3.17b), rearranging now yields

$$\int_{\Omega} \frac{\partial\psi_h}{\partial t} d\Omega = - \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \bar{\psi}_h d\Gamma - \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \psi_h d\Gamma - \oint_{\Gamma_N} \gamma h_a d\Gamma. \quad (3.22)$$

For a point-wise divergence free vector field \mathbf{a} the boundary integral on the union of interior cell facets vanishes, due to $\bar{\psi}_h$ being single-valued on facets $F \in \mathcal{F}$. The right-hand side therefore equals the total ingoing advective flux at the exterior boundary Γ , thereby proving global mass conservation.

For local mass conservation, setting $\delta\lambda_h = 1$ on cell K and $\delta\lambda_h = 0$ on $\Omega \setminus K$ gives, after rearrangement,

$$\int_K \frac{\partial\psi_h}{\partial t} d\Omega = - \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \bar{\psi}_h d\Gamma - \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \psi_h d\Gamma - \oint_{\Gamma_N} \gamma h d\Gamma \quad (3.23)$$

The right-hand side of Eq. (3.23) constitutes the ingoing advective flux on the cell facet ∂K which proves local conservation in terms of the numerical flux on \mathcal{F} .

Note that the proofs for global and local conservation only require the Lagrange multiplier space to contain piecewise constant polynomials. This observation is further exploited in Section 3.6.2 to obtain a significant simplification in the fully-discrete setting.

3.5. A fully-discrete formulation

Equipped with the semi-discrete formulations from the previous sections, this section introduces a fully-discrete formulation for the particle-mesh operator splitting by presenting a suitable time-discrete combination of the four model components.

On the subinterval $I_n = (t^n, t^{n+1}]$ the fully-discrete formulation of the scheme consists of the following sequence of consecutive steps:

1. *Particle advection*: $\mathbf{x}_p^n \rightarrow \mathbf{x}_p^{n+1}$; the particle position is updated in a Lagrangian manner using an advective velocity field, see Section 3.5.1.
2. *Particle-mesh projection*: $\psi_p^n \rightarrow \psi_h^{*,n} \rightarrow \phi_h^{*,n}$; an intermediate mesh-based field $\phi_h^{*,n}$ is constructed from the particle scalar quantity ψ_p^n and the updated particle positions \mathbf{x}_p^{n+1} , using either the local ℓ^2 -projections (Section 3.5.2) or the PDE-constrained projections (Section 3.5.3).
3. *Solution of the diffusion equation on the mesh*: $\phi_h^{*,n} \rightarrow \phi_h^{n+1}$; the intermediate mesh field $\phi_h^{*,n}$ is corrected to account for the diffusive flux and external sources, see Section 3.5.4.

4. *Mesh-particle projection*: $(\phi_h^{*,n}, \phi_h^{n+1}) \rightarrow \psi_p^{n+1}$; the particle scalar quantity is updated given the solution at the mesh, see Section 3.5.5.

The operator splitting approach is evident from the steps outlined above since the advection step at the particle level leads to an intermediate mesh field that needs correction at the mesh level to account for the diffusive fluxes. A careful formulation for the different steps is crucial to minimize the associated splitting error and to achieve higher-order accuracy in time. In particular, special care is required to render the fully-discrete particle-mesh projection in Section 3.5.3 compatible with the fully-discrete mesh-particle projection in Section 3.5.5. Below, a scheme is presented that makes it possible to obtain second-order time accuracy.

3.5.1. Particle advection

From Eq. (3.11a) it follows that particle quantities other than the position remain constant throughout the Lagrangian particle advection stage. A fully-discrete implementation of this step is therefore obtained by integrating Eq. (3.11b) in time to advance the particle position from $\mathbf{x}_p^n \rightarrow \mathbf{x}_p^{n+1}$. For this purpose, we use a three-stage third-order accurate Runge-Kutta scheme [80].

At inflow boundaries, particles have to enter the domain. To this end, particles are seeded in the layer of cells contiguous to the inflow boundaries, in such a way to keep the number of particles constant in these cells. In order to remain consistent with the boundary conditions in Eq. (3.2), properties of the inserted particles are interpolated at the intersection point of the particle trajectory with the domain boundary, using the corresponding values imposed by the boundary conditions.

3.5.2. Particle-mesh projection I: ℓ^2 projection

In a time-discrete setting, the local minimization problem of Eq. (3.15) reads as follows: given the particle values $\psi_p^n \in \Psi_t$ and particle positions $\mathbf{x}_p^{n+1} \in \chi_t$, find $\psi_h^{n+1} \in W_h$ such that

$$\sum_{p \in \mathcal{S}_t} (\psi_h^{n+1}(\mathbf{x}_p^{n+1}) - \psi_p^n) w_h(\mathbf{x}_p^{n+1}) = 0. \quad (3.24)$$

This problem belongs to the class of quadratic cone programs, for which numerous and well-established techniques exist to extend such problems with equality and/or inequality constraints, see e.g. [79] and references. In the sequel, box constraints

$$l \leq \psi_h^{n+1} \leq u \quad (3.25)$$

are frequently imposed in order to require the mesh field to be bounded by $[l, u]$. For a computer implementation of constrained quadratic cone programs, use is made of the existing packages CVXOPT¹ (Python) or QuadProg++² (C++).

3.5.3. Particle-mesh projection II: PDE-constrained projection

The fully-discrete PDE-constrained particle-mesh projection is formulated with the objective to find the optimal scalar field ψ_h at time level $n + 1$ given the particle field

¹<https://cvxopt.org/>

²<https://github.com/liuq/QuadProgpp>

ψ_p at time t^{n+1} . Employing the θ -method, where $1/2 \leq \theta \leq 1$, the constraint in the Lagrangian functional \mathcal{L} is evaluated at time $t^{n+\theta} := (1-\theta)t^n + \theta t^{n+1}$ using linear interpolation between discrete time levels. To this end, the scalar field ψ_h at time level $n + \theta$ is approximated by

$$\psi_h(t^{n+\theta}) \approx (1-\theta)\psi_h^{*,n} + \theta\psi_h^{n+1} \quad (3.26)$$

in which $\psi_h^{*,n} \in W_h$ is an initial field given by

$$\psi_h^{*,n} = \psi_h^n + \Delta t_n \left((1-\theta_L)\dot{\phi}_h^{n-1} + \theta_L\dot{\phi}_h^n \right), \quad (3.27)$$

where $\Delta t_n = t^{n+1} - t^n$ is the time step size. Furthermore, θ_L is an additional time stepping parameter ($1/2 \leq \theta_L \leq 1$, but possibly different from θ). For $\theta_L = 1/2$, the term within the brackets yields a second-order explicit diffusion term, since the increments $\dot{\phi}_h^m$ (with $m = n-1, n$) are defined by

$$\dot{\phi}_h^m = \frac{\phi_h^m - \phi_h^{*,m-1}}{\Delta t_{m-1}}, \quad (3.28)$$

with ϕ_h^m and $\phi_h^{*,m-1} = \psi_h^{*,m}$ being fields in W_h related to the particular time stepping scheme used in the diffusion problem (step 3) and mesh-particle projection (step 4), see Sections 3.5.4 and 3.5.5. The use of $\psi_h^{*,n}$ in Eq. (3.27) in place of ψ_h^n ensures that the fully-discrete projection operators \mathcal{P}_L and \mathcal{P}_E are mutually consistent, as will be further elaborated in Section 3.6.1. The time derivative of the scalar field ψ_h at time level $n + \theta$ is now given by

$$\left. \frac{\partial \psi_h}{\partial t} \right|_{t^{n+\theta}} \approx \frac{\psi_h^{n+1} - \psi_h^{*,n}}{\Delta t_n}, \quad (3.29)$$

which follows from the linear interpolation used in Eq. (3.26).

Next, variations of the dependent fields are taken with respect to the degrees of freedom at time level $n + 1$, which involves the replacement of variations $(\delta\phi_h, \delta\lambda_h, \delta\bar{\phi}_h) \in (W_h, T_h, \bar{W}_h)$ in the optimality system Eq. (3.17) with test functions $(w_h, \tau_h, \bar{w}_h) \in (W_h, T_h, \bar{W}_h)$. Using the expression for the time derivative of ψ_h given in Eq. (3.29), the time derivative appearing in the co-state equation (3.17a) is approximated as follows

$$\left. \frac{\partial \delta \psi_h}{\partial t} \right|_{t^{n+\theta}} \approx \frac{\delta \psi_h^{n+1} - \delta \psi_h^{*,n}}{\Delta t_n} = \frac{w_h}{\Delta t_n}, \quad (3.30)$$

since variations $\delta \psi_h^{*,n} \in W_h$ vanish.

Given these approximations, the fully-discrete co-state equation reads: given the particle field $\psi_p^n \in \Psi_t$, the particle positions $\mathbf{x}_p^{n+1} \in \mathcal{X}_t$, and the intermediate field $\psi_h^{*,n} \in W_h$, find $(\psi_h^{n+1}, \lambda_h^{n+1}, \bar{\psi}_h^{n+1}) \in (W_h, T_h, \bar{W}_{h,g})$ such that

$$\begin{aligned} & \sum_{p \in \mathcal{S}_t} (\psi_h^{n+1}(\mathbf{x}_p^{n+1}) - \psi_p^n) w_h(\mathbf{x}_p^{n+1}) - \sum_K \oint_{\partial K} \beta (\bar{\psi}_h^{n+1} - \psi_h^{n+1}) w_h \, d\Gamma + \int_{\Omega} \frac{w_h}{\Delta t_n} \lambda_h^{n+1} \, d\Omega \\ & - \theta \sum_K \int_K (\mathbf{a} w_h) \cdot \nabla \lambda_h^{n+1} \, d\Omega + \theta \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \lambda_h^{n+1} w_h \, d\Gamma = 0 \quad \forall w_h \in W_h. \end{aligned} \quad (3.31a)$$

Correspondingly, the fully-discrete counterpart of the state equation Eq. (3.17b) reads:

$$\begin{aligned}
& \int_{\Omega} \frac{\psi_h^{n+1} - \psi_h^{*,n}}{\Delta t_n} \tau_h \, d\Omega - \theta \sum_K \int_K (\mathbf{a} \psi_h^{n+1}) \cdot \nabla \tau_h \, d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \bar{\psi}_h^{n+1} \tau_h \, d\Gamma \\
& \quad + \theta \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \psi_h^{n+1} \tau_h \, d\Gamma + \oint_{\Gamma_N} \gamma h_a^{n+\theta} \tau_h \, d\Gamma \\
& = (1 - \theta) \sum_K \int_K (\mathbf{a} \psi_h^{*,n}) \cdot \nabla \tau_h \, d\Omega - (1 - \theta) \oint_{\Gamma_N} (1 - \gamma) \mathbf{a} \cdot \mathbf{n} \psi_h^{*,n} \tau_h \, d\Gamma \quad \forall \tau_h \in T_h.
\end{aligned} \tag{3.31b}$$

Finally, the fully-discrete optimality condition becomes

$$\sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{a} \cdot \mathbf{n} \lambda_h^{n+1} \bar{w}_h \, d\Gamma + \sum_K \oint_{\partial K} \beta (\bar{\psi}_h^{n+1} - \psi_h^{n+1}) \bar{w}_h \, d\Gamma = 0 \quad \forall \bar{w}_h \in \bar{W}_{h,0}. \tag{3.31c}$$

In these equations, the Lagrange multiplier λ_h and the control variable $\bar{\psi}_h$ are conveniently taken at time level $n+1$, which is allowed since these variables are fully-implicit, not requiring differentiation in time.

The reconstructed field ψ_h^{n+1} which is obtained after solving Eq. (3.31) for $(\psi_h^{n+1}, \lambda_h^{n+1}, \bar{\psi}_h^{n+1})$, provides the condition at t^n to advance the diffusion equation to t^{n+1} .

3.5.4. Diffusion equation

Using a backward Euler time integration and using the fact that $(\psi_h, \phi_h) \in W_h$ allows for trivial projections. The fully-discrete counterparts of Eq. (3.12) reads: given the initial field $\phi_h^{*,n} = \psi_h^{n+1} \in W_h$, the source term f^{n+1} , the diffusive Neumann boundary condition h_d^{n+1} and the diffusivity κ , find $\phi_h^{n+1} \in W_h$ and $\bar{\phi}_h^{n+1} \in \bar{W}_{h,g}$ such that

$$\begin{aligned}
& \int_{\Omega} \frac{\phi_h^{n+1} - \phi_h^{*,n}}{\Delta t_n} w_h \, d\Omega + \sum_K \int_K \kappa \nabla \phi_h^{n+1} \cdot \nabla w_h \, d\Omega + \sum_K \oint_{\partial K} \hat{\sigma}_{d,h}^{n+1} \cdot \mathbf{n} w_h \, d\Gamma \\
& \quad + \sum_K \oint_{\partial K} \kappa (\bar{\phi}_h^{n+1} - \phi_h^{n+1}) \mathbf{n} \cdot \nabla w_h \, d\Gamma = \int_{\Omega} f^{n+1} w_h \, d\Omega \quad \forall w_h \in W_h,
\end{aligned} \tag{3.32a}$$

and

$$\sum_K \oint_{\partial K} \hat{\sigma}_{d,h}^{n+1} \cdot \mathbf{n} \bar{w}_h \, d\Gamma = \oint_{\Gamma_N} h_d^{n+1} \bar{w}_h \, d\Gamma \quad \forall \bar{w}_h \in \bar{W}_{h,0}, \tag{3.32b}$$

in which the fully-discrete diffusive flux $\hat{\sigma}_{d,h}^{n+1}$ is given by

$$\hat{\sigma}_{d,h}^{n+1} = -\kappa \nabla \phi_h^{n+1} - \frac{\alpha}{h_K} \kappa (\bar{\phi}_h^{n+1} - \phi_h^{n+1}) \mathbf{n}. \tag{3.32c}$$

Solving Eq. (3.32) for $(\phi_h^{n+1}, \bar{\phi}_h^{n+1})$ provides the ingredients necessary to update the particle field ψ_p in the subsequent mesh-particle projection step.

3.5.5. Mesh-particle projection

Finally, a fully-discrete mesh-particle projection is formulated by mapping the increments of the mesh related field (ϕ_h) to the particles using

$$\psi_p^{n+1} = \psi_p^n + \Delta t_n ((1 - \theta_L) \dot{\phi}_h^n(\mathbf{x}_p^n) + \theta_L \dot{\phi}_h^{n+1}(\mathbf{x}_p^{n+1})) \quad \forall p \in \mathcal{S}_t, \quad (3.33)$$

where $1/2 \leq \theta_L \leq 1$, and $\dot{\phi}_h^n \in W_h$ is defined according to Eq. (3.28). We emphasize the similarity between the formulation for the mesh-particle update (Eq. (3.33)) and the definition of $\psi_h^{*,n} \in W_h$ in Eq. (3.27), which is required to respect the consistency condition, Eq. (3.20), in the fully-discrete setting. This will be topic of further investigation in Section 3.6.1. Furthermore, it readily follows that in the advective limit (i.e. for $\kappa = 0$) it holds that $\psi_p^{n+1} = \psi_p^n$, since $\dot{\phi}_h^n = \dot{\phi}_h^{n+1} = 0$.

This mesh-particle projection concludes the discrete sequence of steps for the hybrid particle-mesh operator splitting of the scalar advection-diffusion equation. The time accuracy of the resulting combination of steps is further addressed in Section 3.6.4.

3.6. Algorithmic aspects

3.6.1. Conservation and consistency

The local conservation property of the discrete diffusion step was proven in [3], the local conservation proof for the PDE-constrained particle-mesh scheme in the time-discrete setting is analogous to the semi-discrete proof from Section 3.4.2. As such, the resulting particle-mesh scheme possesses local conservation properties at the mesh level for ψ_h^{n+1} in terms of the implicit interface flux variable $\bar{\psi}_h^{n+1}$, and for ϕ_h^{n+1} in terms of the numerical flux $\hat{\sigma}_{d,h}^{n+1}$.

Consistency in the fully-discrete setting requires that the operator splitting has to be consistent with the unsplit governing equations, Eq. (3.1). Splitting this advection-diffusion equation into a kinematic part (advection problem) and a constitutive part (diffusion equation) has been the subject of numerous studies, and is known to be consistent up to a time step dependent splitting error which vanishes in the continuous time limit, see [58–60] among many others. For a consistency proof of the Eulerian diffusion step, reference is made to [3]. Mutual consistency of the projection-steps when the local ℓ^2 particle-mesh projection is employed is unaffected by the time discretization, since the particle-mesh projection does not involve any time derivatives in this case. What remains, is to show consistency of the PDE-constrained particle-mesh projection operator in the fully-discrete setting.

Consider therefore a repeated back-and-forth mapping of a scalar-valued quantity between the particles and the mesh via \mathcal{P}_E and \mathcal{P}_L , while omitting the particle advection step (i.e. $\mathbf{a} = \mathbf{0}$) and the diffusion step (i.e. $\kappa = 0$), and choosing a polynomial order $l = 0$ for the Lagrange multiplier space, Eq. (3.5). For ease of presentation, we further assume that the regularization term governed by β is negligibly small as motivated in Section 3.3.2. Consistency in the fully-discrete setting now requires the constraint equations to be inherently satisfied, i.e. $\lambda_h = 0$ everywhere.

To verify this consistency criterion, consider the discrete optimality system Eq. (3.31) at time level $n + 1$. For a cell K and with $\mathbf{a} = \mathbf{0}$ and $l = 0$, these conditions simplify to

$$\sum_{p \in \mathcal{S}_t^K} \psi_h^{n+1}(\mathbf{x}_p) w_h(\mathbf{x}_p) + \int_K \frac{w_h}{\Delta t} \lambda_h^{n+1} d\Omega = \sum_{p \in \mathcal{S}_t^K} \psi_p^n w_h(\mathbf{x}_p), \quad (3.34a)$$

$$\int_K \frac{\psi_h^{n+1}}{\Delta t} \tau_h d\Omega = \int_K \frac{\psi_h^{*,n}}{\Delta t} \tau_h d\Omega, \quad (3.34b)$$

where we used that $\bar{\psi}_h^{n+1}$ and ψ_h^{n+1} coincide in a weak sense over the facets by virtue of the discrete optimality condition for $\mathbf{a} = \mathbf{0}$, Eq. (3.31c).

With the element contributions following from Eq. (3.34), the algebraic system for a single element K , expressed in terms of the local unknowns ψ^{n+1} and the unknowns λ^{n+1} for the Lagrange multiplier, can be written as

$$\begin{bmatrix} \mathbf{M}_p & \mathbf{G} \\ \mathbf{G}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \psi^{n+1} \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \chi_p \psi_p^n \\ \mathbf{G}^\top \psi^{*,n} \end{bmatrix}, \quad (3.35)$$

in which \mathbf{M}_p resembles a mass matrix, sampled on the particle locations \mathbf{x}_p , the matrix \mathbf{G} collects the terms involving the Lagrange multiplier λ_h , and the vector $\chi_p \psi_p^n$ contains the particle contributions.

Performing a Gaussian elimination, results in the following algebraic equation for λ^{n+1} :

$$\mathbf{G}^\top \mathbf{M}_p^{-1} \mathbf{G} \lambda^{n+1} = \mathbf{G}^\top (\mathbf{M}_p^{-1} \chi_p \psi_p^n - \psi^{*,n}). \quad (3.36)$$

Using the definition of $\psi_h^{*,n}$ in Eq. (3.27) and updating the particle field ψ_p^n via Eq. (3.33), it follows that the right hand side of Eq. (3.36) equals zero. Since the mass matrix \mathbf{M}_p is symmetric and non-negative definite [24, 37], the Schur-complement $\mathbf{G}^\top \mathbf{M}_p^{-1} \mathbf{G}$ is non-singular. The solution of Eq. (3.36) is then given by $\lambda_h^{n+1} = \mathbf{0}$ which, after substitution in Eq (3.35), leads to $\psi_h^{n+1} = \mathbf{M}_p^{-1} \chi_p \psi_p^n = \psi_h^{*,n}$. This proves mutual consistency of the operators \mathcal{P}_E and \mathcal{P}_L .

3.6.2. Choosing the Lagrange multiplier space

A significant simplification of the fully-discrete optimality system is obtained when choosing the polynomial basis for the Lagrange multiplier of the lowest possible order, i.e. $l = 0$. For this particular choice, the cell integrals in Eqs. (3.31) containing gradients of the Lagrange multiplier, or of its associated test function, vanish.

This choice will not affect the conservation proofs given by Eqs. (3.22) and (3.23). Indeed, for the choice $l = 0$ it readily follows from Eq. (3.31b) that the cell-integrated temporal increments of the conserved variables remain in balance with the facet fluxes. Except for the outflow Neumann boundaries, these facet fluxes are implicitly controlled by the interface variable $\bar{\psi}_h^{n+1} \in \bar{W}_{h,g}$, not requiring differentiation in time. Hence, for the presented time integration scheme the major advantage of choosing $l = 0$ is that it avoids time stepping dependencies in the PDE-constrained particle-mesh projection, thus rendering the particle-mesh projection independent of θ . In view of this evident

advantage, we will restrict the discussion in the sequel to the choice $l = 0$. In fact, it is only for this particular choice of the Lagrange multiplier space that second-order accuracy in time can be expected for the given time integration scheme, see Section 3.6.4. Furthermore, independent of the choice for l , we expect optimal spatial convergence rates of the scheme of order $k + 1$, provided a sufficiently accurate particle advection scheme is used.

Finally, albeit $l = 0$ is sufficient to guarantee local conservation of linear quantities (e.g. mass and momentum), it is conjectured that conservation of quadratic quantities (e.g. energy) requires $l > 0$. This topic - in combination with alternative time stepping strategies - is left as an interesting area for future investigation.

3.6.3. Static condensation

The optimality system Eq. (3.31) leads to a seemingly large global system. However, this system is efficiently implemented by eliminating the unknowns local to a cell, i.e. $(\psi_h, \lambda_h) \in (W_h, T_h)$, in favor of the global control variable $\bar{\psi}_h^{n+1} \in \bar{W}_h$, leading to a much smaller global system which is to be solved for $\bar{\psi}_h^{n+1}$ only. The local unknowns ψ_h^{n+1} and λ_h^{n+1} can be found in a subsequent backsubstitution step.

This procedure, known as *static condensation*, is sketched by considering the problem on a single cell K , which can be cast into the following algebraic form for the local unknowns ψ^{n+1} , the Lagrange multiplier unknowns, λ^{n+1} and the flux degrees of freedom at the facets $\bar{\psi}^{n+1}$:

$$\begin{bmatrix} \mathbf{M}_p + \mathbf{N} & \mathbf{G}(\theta) & \mathbf{L} \\ \mathbf{G}(\theta)^\top & \mathbf{0} & \mathbf{H} \\ \mathbf{L}^\top & \mathbf{H}^\top & \mathbf{B} \end{bmatrix} \begin{bmatrix} \psi^{n+1} \\ \lambda^{n+1} \\ \bar{\psi}^{n+1} \end{bmatrix} = \begin{bmatrix} \chi_p \psi_p^n \\ \mathbf{G}(1 - \theta)^\top \psi^{*,n} \\ \mathbf{0} \end{bmatrix}. \quad (3.37)$$

In this equation, \mathbf{M}_p resembles a mass matrix, sampled at the particle locations \mathbf{x}_p , matrices \mathbf{N} , \mathbf{L} and \mathbf{B} result from a non-zero regularization parameter β , \mathbf{G} collects the terms involving the Lagrange multiplier, \mathbf{H} represents the flux term over interior facets, and the vector, and $\chi_p \psi_p^n$ contains the particle contributions. Reference to Eq. (3.31) is made for a full account of the individual contributions.

Local elimination of ψ^{n+1} and λ^{n+1} leads to

$$\begin{aligned} \bigwedge_K \left(\mathbf{B} - \begin{bmatrix} \mathbf{L} \\ \mathbf{H} \end{bmatrix}^\top \begin{bmatrix} \mathbf{M}_p + \mathbf{N} & \mathbf{G}(\theta) \\ \mathbf{G}(\theta)^\top & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{L} \\ \mathbf{H} \end{bmatrix} \right) \bar{\psi}^{n+1} \\ = - \bigwedge_K \begin{bmatrix} \mathbf{L} \\ \mathbf{H} \end{bmatrix}^\top \begin{bmatrix} \mathbf{M}_p + \mathbf{N} & \mathbf{G}(\theta) \\ \mathbf{G}^\top(\theta) & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \chi_p \psi_p^n \\ \mathbf{G}(1 - \theta)^\top \psi^{*,n} \end{bmatrix}, \end{aligned} \quad (3.38)$$

where \bigwedge_K denotes assembly of the element contributions into the global matrix, which requires the inversion of a small saddle-point problem for each cell K independently. Hence, the assembly procedure is amenable to a fast parallel implementation.

After solving Eq. (3.38) for $\bar{\psi}^{n+1}$, the local unknowns are found by the backsubstitution

$$\begin{bmatrix} \psi^{n+1} \\ \lambda^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_p + \mathbf{N} & \mathbf{G}(\theta) \\ \mathbf{G}^\top(\theta) & \mathbf{0} \end{bmatrix}^{-1} \left(\begin{bmatrix} \chi_p \psi_p^{n+1} \\ \mathbf{G}(1 - \theta)^\top \psi^{*,n} \end{bmatrix} - \begin{bmatrix} \mathbf{L} \\ \mathbf{H} \end{bmatrix} \bar{\psi}^{n+1} \right), \quad (3.39)$$

which is performed for each cell K independently.

Similarly to the static condensation procedure for the particle-mesh projection, the fully-discrete diffusion equation can be implemented in an efficient fashion using a static condensation strategy. This topic is postponed to the next chapter when discussing the more complicated algebraic system for the Stokes equations.

3.6.4. Time accuracy

Because of the operator splitting nature of the scheme, the time-accuracy of the particular time stepping strategy as presented in Section 3.5 requires further consideration. A first and almost self-evident requirement is that the particle advection scheme should not obstruct the overall convergence. In practice, this implies that the accuracy of the particle advection scheme should be at least the same or higher than the accuracy which can be achieved in the remaining steps. In this chapter, an off-the-shelf third-order accurate Runge-Kutta scheme is used [80].

More interesting is the analysis of the combination of the projection steps in conjunction with the time integration strategy during the diffusion step. A backward Euler scheme for the Eulerian diffusion step is proposed (Section 3.5.4), in combination with a θ -update of the particle properties in the mesh-particle projection (Section 3.5.5). With the particle-mesh projection being formulated in terms of local ℓ^2 -projections, it is shown in Appendix B that this combination gives overall second-order accuracy by choosing $\theta_L = 0.5$. The latter requires saving the particle field from the previous time step. The key in this derivation is that Backward Euler suffices on the Eulerian step since ϕ_h^{n+1} is advanced over one time step from the second-order field $\phi_h^{*,n}$, with this field being reconstructed from the particles. Thus, the local, second-order accuracy of backward Euler pertains.

Care must be taken to retain this second-order accuracy in time when the PDE-constrained optimization procedure is used in the particle-mesh projection, and the challenge is to ensure second-order time accuracy of the reconstructed field $\phi_h^{*,n}$. Firstly, this requires the Lagrange multiplier space to be of the lowest polynomial order, i.e. $l = 0$. This choice avoids time stepping dependencies in the particle-mesh projection, and thus render the particle-mesh projection independent of θ , see Section 3.6.2. In addition, ensuring compatibility between the mesh-particle update, Eq. (3.33), and the PDE-constrained particle-mesh interaction in the fully-discrete setting, Eq. (3.31) leads to a definition for $\psi_h^{*,n}$ via Eq. (3.27), rendering this field second-order accurate in the time step for $\theta_L = 0.5$.

3.7. Extension to moving meshes

The flexibility to redefine the mesh every timestep is often considered an advantage of hybrid particle-mesh methods [81], and can be used in large deformation solid mechanics problems, or for implementing moving boundary conditions in fluid mechanics problems [82]. This feature hinges on the requirement that the mesh is ‘history-free’, i.e. no information is carried at the mesh level between consecutive time steps. This condition is obviously met when using the local ℓ^2 approach for the particle-mesh projection, with the mesh field being reconstructed from the particle data without knowledge of prior

mesh fields.

This does not hold true anymore for the constrained particle-mesh projection, since the mesh field at the old time level is required for evaluating the time derivative in the state equation. Hence, the mesh cannot be *redefined* arbitrarily. This section, however, demonstrates that the constrained particle-mesh projection can be reformulated in a straightforward manner to accommodate *moving* meshes, while global and local conservation properties are retained, and where the so-called 'discrete geometric conservation law' (DGCL) [83] is circumvented.

To this end, consider a spatial domain, denoted by $\Omega \subset \mathbb{R}^d$, a material domain $\Omega_0 \subset \mathbb{R}^d$, and a reference domain, denoted by $\hat{\Omega} \subset \mathbb{R}^d$. This reference domain can be the domain at $t = 0$ or, alternatively, it can be the spatial domain from an earlier time step as henceforth will be the case. Points $\mathbf{x} \in \Omega$ are fixed in space, points $\mathbf{X} \in \Omega_0$ move along with the flow in a Lagrangian fashion, and points $\boldsymbol{\chi} \in \hat{\Omega}$ are attached to an arbitrary reference frame. With these definitions, the convective velocity of a material point relative to the spatial domain is defined as [84]

$$\mathbf{c}(\mathbf{x}, t) = \mathbf{a} - \hat{\mathbf{a}} \quad (3.40)$$

in which $\mathbf{a}(\mathbf{X}, t)$ the Eulerian velocity of a Lagrangian point, and $\hat{\mathbf{a}}(\boldsymbol{\chi}, t)$ the Eulerian velocity of a reference point, indicated with $\boldsymbol{\chi}$. In practice, \mathbf{a} equals the physical transport velocity from earlier sections, whereas $\hat{\mathbf{a}}$ is the velocity of the moving mesh.

To account for the changes in the mesh volume and the fluxes over the moving facets as a result of the mesh movement, the Lagrangian underpinning the PDE-constrained particle-mesh projection on fixed meshes, Eq. (3.17), is restated as

$$\begin{aligned} \mathcal{L}(\psi_h, \bar{\psi}_h, \lambda_h) &= \sum_p \frac{1}{2} (\psi_h(\mathbf{x}_p(t), t) - \bar{\psi}_p(t))^2 + \sum_K \oint_{\partial K} \frac{1}{2} \beta (\bar{\psi}_h - \psi_h)^2 d\Gamma \\ &+ \frac{d}{dt} \int_{\hat{\Omega}} J \psi_h \lambda_h d\hat{\Omega} - \int_{\hat{\Omega}} J \psi_h (\mathbf{F}^{-1} \mathbf{c}) \cdot \hat{\nabla} \lambda_h d\hat{\Omega} + \sum_{\hat{K}} \oint_{\partial \hat{K} \setminus \hat{\Gamma}_N} J \bar{\psi}_h (\mathbf{F}^{-1} \mathbf{c}) \cdot \hat{\mathbf{n}} \lambda_h d\hat{\Gamma} \\ &+ \oint_{\hat{\Gamma}_N} (1 - \gamma) J \psi_h (\mathbf{F}^{-1} \mathbf{c}) \cdot \hat{\mathbf{n}} \lambda_h d\hat{\Gamma} + \oint_{\hat{\Gamma}_N} \gamma \hat{h} \lambda_h d\hat{\Gamma} \quad (3.41) \end{aligned}$$

in which the hyperbolic constraint is formulated on the reference domain $\hat{\Omega}$, see also [15]. Furthermore, $\mathbf{F} = \hat{\nabla} \mathbf{x}$ is the deformation tensor, and $J = \det \mathbf{F}$.

To obtain a straightforward implementation of this optimization problem, with minimal differences compared to the discrete optimality system on fixed meshes, let's first consider a time discrete formulation of the constraint, where it is assumed that the polynomial order of the Lagrange multiplier is $l = 0$, see Section 3.6.2. Given this

assumption, and integrating the constraint equation in time yields

$$\begin{aligned} & \frac{1}{\Delta t} \int_{\hat{\Omega}} (J^{n+1} \psi_h^{n+1} - J^n \psi_h^{*,n}) \lambda_h^{n+1} \, d\hat{\Omega} + \sum_{\hat{K}} \oint_{\partial \hat{K} \setminus \hat{\Gamma}_N} J^n \bar{\psi}_h^{n+1} (\mathbf{F}^{-1} \mathbf{c})^n \cdot \hat{\mathbf{n}} \lambda_h^{n+1} \, d\hat{\Gamma} \\ & + \oint_{\hat{\Gamma}_N} (1 - \gamma) J^n \psi_h (\mathbf{F}^{-1} \mathbf{c})^n \cdot \hat{\mathbf{n}} \lambda_h^{n+1} \, d\hat{\Gamma} + \oint_{\hat{\Gamma}_N} \gamma \hat{h}^{n+1} \lambda_h^{n+1} \, d\hat{\Gamma} = 0. \quad (3.42) \end{aligned}$$

Next, all the terms except the term involving $\psi_h^{*,n}$ are pushed from the reference domain to the spatial domain. This requires the introduction of the inverse mapping $\mathcal{R}: \mathbf{x} \rightarrow \boldsymbol{\chi}$, e.g. $\tilde{\lambda}_h^{n+1}(\mathbf{x}, t) \equiv \lambda_h^{n+1}(\mathcal{R}(\mathbf{x}), t)$. For simplicity, the latter is assumed implicit in the notation by making use of a ‘time-honored abuse of notation’ [85] in the ALE literature, by simply writing $\lambda_h^{n+1}(\mathbf{x}, t) = \lambda_h^{n+1}(\mathcal{R}(\mathbf{x}), t)$. This notation appears convenient, since the push-forward only involves implicit variables, i.e. variables defined at time level $n+1$. Furthermore, by choosing the reference domain equal to the spatial domain at the old time level n , so that $\mathbf{F}^n = \mathbf{I}$ is identity and $J^n = 1$, the time-discrete constraint, Eq. (3.42), reads

$$\begin{aligned} & \frac{1}{\Delta t} \left(\int_{\Omega} \psi_h^{n+1} \lambda_h^{n+1} \, d\Omega - \int_{\hat{\Omega}} \psi_h^{*,n} \lambda_h^{n+1} \, d\hat{\Omega} \right) + \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{c}^n \cdot \mathbf{n} \bar{\psi}_h^{n+1} \lambda_h^{n+1} \, d\Gamma \\ & + \oint_{\Gamma_N} (1 - \gamma) \mathbf{c}^n \cdot \mathbf{n} \psi_h^{n+1} \lambda_h^{n+1} \, d\Gamma + \oint_{\Gamma_N} \gamma h^{n+1} \lambda_h^{n+1} \, d\Gamma = 0, \quad (3.43) \end{aligned}$$

where the integral involving $\psi_h^{*,n}$ is evaluated at the reference configuration, i.e. at the mesh configuration at time level n . Furthermore, in the absence of mesh motion, i.e. for convective velocity $\mathbf{c} = \mathbf{a}$, Eq. (3.43) reduces to the discrete constraint on a fixed mesh.

Given this fully-discrete constraint, the same optimization procedure is followed as above by taking variations with respect to $(\psi_h^{n+1}, \bar{\psi}_h^{n+1}, \lambda_h^{n+1})$, resulting in the following fully-discrete optimality system on the moving mesh. The discrete co-state equation reads

$$\begin{aligned} & \sum_{p \in \mathcal{S}_t} (\psi_h^{n+1}(\mathbf{x}_p^{n+1}) - \psi_p^n) w_h(\mathbf{x}_p^{n+1}) - \sum_K \oint_{\partial K} \beta (\bar{\psi}_h^{n+1} - \psi_h^{n+1}) w_h \, d\Gamma \\ & + \frac{1}{\Delta t} \int_{\Omega} w_h \lambda_h^{n+1} \, d\Omega + \oint_{\Gamma_N} (1 - \gamma) \mathbf{c}^n \cdot \mathbf{n} \lambda_h^{n+1} w_h \, d\Gamma = 0 \quad \forall w_h \in W_h. \quad (3.44a) \end{aligned}$$

Correspondingly, the fully-discrete counterpart of the state equation Eq. (3.17b) reads:

$$\begin{aligned} \frac{1}{\Delta t} \left(\int_{\Omega} \psi_h^{n+1} \tau_h \, d\Omega - \int_{\hat{\Omega}} \psi_h^{*,n} \hat{\tau}_h \, d\hat{\Omega} \right) + \sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{c}^n \cdot \mathbf{n} \bar{\psi}_h^{n+1} \tau_h \, d\Gamma \\ + \oint_{\Gamma_N} (1 - \gamma) \mathbf{c}^n \cdot \mathbf{n} \psi_h^{n+1} \tau_h \, d\Gamma + \oint_{\Gamma_N} \gamma h^{n+1} \tau_h \, d\Gamma = 0 \quad \forall \tau_h \in T_h, \end{aligned} \quad (3.44b)$$

where all integrals are evaluated at the current spatial configuration of the mesh, except for the integral over the reference configuration $\hat{\Omega}$, which is set equal to the domain at time level t^n . Finally, the fully-discrete optimality condition becomes

$$\sum_K \oint_{\partial K \setminus \Gamma_N} \mathbf{c}^n \cdot \mathbf{n} \lambda_h^{n+1} \bar{w}_h \, d\Gamma + \sum_K \oint_{\partial K} \beta (\bar{\psi}_h^{n+1} - \psi_h^{n+1}) \bar{w}_h \, d\Gamma = 0 \quad \forall \bar{w}_h \in \bar{W}_{h,0}. \quad (3.44c)$$

It is emphasized that the only notable difference between the optimality system on the moving mesh and the discrete optimality system at the fixed mesh is in the evaluation of the time derivative in the state equation, Eq. (3.44b).

3.7.1. Conservation

Conservation proofs for the constrained projection on moving meshes proceed along the same line as for the fixed mesh, see Section 3.4.2. Setting $\tau_h = \hat{\tau}_h = 1$ in Eq. (3.44b), it follows that the net change in ψ_h integrated over the moving domain $\hat{\Omega} \rightarrow \Omega$ equals the net ingoing flux expressed in terms of \mathbf{c} . Local conservation is obtained in terms of the implicit facet flux variable $\bar{\psi}_h^{n+1}$. This implicit flux variable, however, needs additional attention in view of the geometric conservation law which is to be discussed next.

3.7.2. Discrete Geometric Conservation Law

Formulating the flow equations on moving meshes, incurs the evaluation of geometric quantities such as the mesh position and the mesh velocity, as well as the mapping of quantities between various domains. In mesh-based methods, a useful principle for evaluating these quantities is the so-called discrete geometric conservation law (DGCL), indicating whether a uniform field remains uniform under an arbitrary mesh motion [83]. The DGCL amounts to the following differential identity (see, e.g., [86])

$$\frac{d}{dt} \int_K d\Omega = \oint_{\partial K} \hat{\mathbf{a}} \cdot \mathbf{n} \, d\Gamma. \quad (3.45)$$

Satisfying this condition in the fully-discrete setting turns out non-trivial, and requires a careful selection of the time integration scheme, see e.g. [84, 86–88] among many others.

Particle-mesh methods typically avoid the complications associated with the DGCL, since no information is stored at the background mesh between consecutive time steps. This is easily verified by considering the local ℓ^2 projections from Section 3.5.2: a constant field at the particles is reconstructed exactly at an arbitrary (moving) mesh, provided a sufficient amount of particles on a per-element basis.

Even though mesh fields needs to be carried over to the next time level in the PDE-constrained particle-mesh projection, the attractive property of the unconstrained ℓ^2 -projection is retained in that a uniform field at particles is reconstructed exactly at an arbitrary moving mesh, up to an error introduced by the parameter β . To illustrate this, consider - without loss of generality - the situation where the convective velocity amounts to the mesh velocity, i.e. $\mathbf{c} = \hat{\mathbf{a}}$. Furthermore, let the initial constant mesh field be $\psi_h^{*,n} = 1$ and particle field $\psi_p^n = 1$. Under the assumption, $\beta \downarrow 0$ it follows from the discrete optimality condition, Eq. (3.44c), that $\lambda_h^{n+1} = 0$, since $\mathbf{c} \neq \mathbf{0}$. With this result, the co-state simplifies significantly, and what remains is the ℓ^2 term as in the unconstrained case, returning the desired uniform solution $\psi_h^{n+1} = \psi_h^{*,n} = 1$.

To elucidate the relation with the DGCL in Eq. (3.45), the solution $\psi_h^{n+1} = \psi_h^{*,n} = 1$ is substituted in the discrete state equation, which for a single cell K in the interior of Ω then becomes

$$\frac{1}{\Delta t} \left(\int_K d\Omega - \int_{\hat{K}} d\hat{\Omega} \right) = \oint_{\partial K} \hat{\mathbf{a}} \cdot \mathbf{n} \bar{\psi}_h^{n+1} d\Gamma, \quad (3.46)$$

in which \hat{K} the cell in the reference configuration, i.e. the cell at time level n .

If $\bar{\psi}_h^{n+1} = 1$, this equation indeed would be a discrete counterpart of Eq. (3.45). Proving this, is not possible, however, but it nevertheless can be argued that such a proof can be circumvented. Indeed, with the control variable being fully implicit, it attains an additional role on moving meshes so as to compensate for the changes in cell volume. At the same time, by virtue of the ℓ^2 contribution, constant fields can be reconstructed exactly for vanishing β as argued above.

In view of this argumentation, two concluding remarks are in place. First and foremost, the above argument obviously only holds exactly for $\beta = 0$. Since $\beta > 0$ in order to couple the state variable ψ_h^{n+1} to the optimality variable $\bar{\psi}_h^{n+1}$, β is kept small, which aligns with the consistency requirement from Section 3.4.1. The influence of this regularization parameter is further verified for a numerical example in Section 3.8.4. Secondly, with the implicit control variable $\bar{\psi}_h$ also accounting for the volume change of a cell - see Eq. (3.46) - the actual values may deviate from the values for the reconstructed field ψ_h . This, however, is not considered a problem as long as one is interested in the state variable ψ_h rather than the precise values of the control variable $\bar{\psi}_h$.

3.8. Numerical examples

In this section, the properties of the proposed method are illustrated for a selection of numerical examples for the linear advection-diffusion equation. Particular attention is paid to the accuracy and the conservation properties of the scheme. Throughout, domains $\Omega \in \mathbb{R}^2$ are considered and the time interval of interest is partitioned using constant time step sizes Δt . Furthermore, the regularization term β is set to a fixed

value of 10^{-6} , unless otherwise specified. The penalty parameter α in the diffusion equation Eq. (3.32) is set to $12k^2$, which is expected to be a safe choice given the values reported in [89]. Unless specified otherwise, we choose the polynomial order $l = 0$ for the Lagrange multiplier space, thus rendering the PDE-constrained particle-mesh projection independent of θ , see Section 3.6.2. The polynomial order k for the diffusion solver is specified per example.

Tools from the finite element framework FEniCS [90] are used to assemble and solve the equations on the mesh arising from the discretization of the PDE-constrained projections and the diffusion equation. A static condensation procedure is applied in all cases and the resulting global systems are solved using direct Gaussian elimination. A discussion of the computer implementation, including some performance tests, is presented in Appendix D.

3.8.1. Advection-diffusion: Gaussian hat

The accuracy of the presented method is assessed by considering a (rotating) Gaussian hat in the diffusive limit, for moderate diffusion, and in the advective limit. The domain of interest is the circular disk $\Omega := \{(x, y) | x^2 + y^2 \leq 0.5\}$ and the velocity field is either set to $\mathbf{a} = \mathbf{0}$ (diffusive limit), or given by

$$\mathbf{a} = \pi(-y, x)^\top. \quad (3.47)$$

The corresponding analytical solution for a rotating Gaussian pulse is given by

$$\phi(\mathbf{x}, t) = \frac{2\sigma^2}{2\sigma^2 + 4\kappa t} \exp\left(-\frac{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2}{2\sigma^2 + 4\kappa t}\right), \quad (3.48)$$

in which (x_c, y_c) is the position vector of the center, and (\bar{x}, \bar{y}) is $(x \cos(\pi t) + y \sin(\pi t), -x \sin(\pi t) + y \cos(\pi t))$. Furthermore, σ is the initial standard deviation, and κ is a constant diffusivity. The initial condition $\phi(\mathbf{x}, 0)$ is deduced from Eq. (3.48), with the standard deviation σ set to 0.1 and center coordinates $(x_c, y_c) = (-0.15, 0)$. The disk-shaped domain Ω is triangulated using a sequence of mesh refinements, and particles are randomly seeded in Ω such that each cell contains on average 30 particles, initially.

Table 3.1: Gaussian hump: Overview of model settings for advection-diffusion.

	Particle-mesh projection	k	l	θ	θ_L	\mathbf{a}
Case 1	PDE	2	0	-	1/2	$\mathbf{0}$
Case 2	PDE	1	0	-	1/2	Eq. (3.47)
Case 3	PDE	2	0	-	1/2	Eq. (3.47)
Case 4	PDE	2	1	1/2	1/2	Eq. (3.47)
Case 5	ℓ^2	2	-	-	1/2	$\mathbf{0}$
Case 6	ℓ^2	2	-	-	1/2	Eq. (3.47)

Six different cases (listed in Table 3.1) are considered for three values of the diffusivity, $\kappa = 0.01$, $\kappa = 0.001$, and $\kappa = 0$ (advective limit). The Dirichlet boundary condition for the diffusion step is deduced from the analytical solution Eq. (3.48), and we emphasize that in the advective limit the particle specific mass ψ_p stays constant by virtue of

Eq. (3.33). The same meshes, initial conditions, and particle distributions are used in all four cases.

Table 3.2: Gaussian hump: L^2 -error for advection-diffusion after one full revolution. Convergence rates based on $h_{K,\max}$.

	Mesh			$\kappa = 0.01$		$\kappa = 0.001$		$\kappa = 0.0$	
	Δt	$h_{K,\min}$	$h_{K,\max}$	$\ \phi - \phi_h\ $	Rate	$\ \phi - \phi_h\ $	Rate	$\ \phi - \phi_h\ $	Rate
Case 1	0.08	6.6e-2	1.2e-1	7.3e-5	-	5.3e-4	-	-	
	0.04	3.2e-2	6.2e-2	1.5e-5	2.3	6.0e-5	3.2		
	0.02	1.6e-2	3.1e-2	3.5e-6	2.1	7.5e-6	3.0		
	0.01	7.9e-3	1.6e-2	8.7e-7	2.0	9.6e-7	3.0		
Case 2	0.08	6.6e-2	1.2e-1	1.2e-3	-	7.2e-3	-	1.3e-2	-
	0.04	3.2e-2	6.2e-2	3.3e-4	1.9	1.8e-3	2.0	3.9e-3	1.8
	0.02	1.6e-2	3.1e-2	6.3e-5	2.4	2.7e-4	2.7	9.6e-4	2.0
	0.01	7.9e-3	1.6e-2	1.2e-5	2.4	9.4e-5	1.6	2.4e-4	2.0
Case 3	0.08	6.6e-2	1.2e-1	1.6e-4	-	8.9e-4	-	2.9e-3	-
	0.04	3.2e-2	6.2e-2	1.6e-5	3.3	1.3e-4	2.8	2.5e-4	3.5
	0.02	1.6e-2	3.1e-2	3.0e-6	2.4	1.9e-5	2.8	3.0e-5	3.1
	0.01	7.9e-3	1.6e-2	7.4e-7	2.0	2.4e-6	2.9	4.4e-6	2.8
Case 4	0.08	6.6e-2	1.2e-1	3.2e-3	-	1.4e-2	-	2.8e-2	-
	0.04	3.2e-2	6.2e-2	2.0e-3	0.7	3.4e-3	2.0	1.5e-2	0.9
	0.02	1.6e-2	3.1e-2	1.1e-3	0.9	9.1e-4	1.9	3.1e-3	2.3
	0.01	7.9e-3	1.6e-2	5.6e-4	0.9	5.1e-4	0.8	8.2e-4	1.9
Case 5	0.08	6.6e-2	1.2e-1	7.3e-5	-	5.3e-4	-	-	
	0.04	3.2e-2	6.2e-2	1.5e-5	2.3	6.0e-5	3.1		
	0.02	1.6e-2	3.1e-2	3.5e-6	2.1	7.5e-6	3.0		
	0.01	7.9e-3	1.6e-2	8.7e-7	2.0	9.6e-7	3.0		
Case 6	0.08	6.6e-2	1.2e-1	3.4e-4	-	1.1e-3	-	1.8e-3	-
	0.04	3.2e-2	6.2e-2	4.1e-5	3.1	1.4e-4	3.0	2.3e-4	3.0
	0.02	1.6e-2	3.1e-2	4.9e-6	3.1	1.8e-5	2.9	2.9e-5	2.9
	0.01	7.9e-3	1.6e-2	7.7e-7	2.7	2.3e-6	3.0	3.7e-6	3.0

Results for the L^2 error after a full revolution are presented in Table 3.2. For Cases 1-3, at least second-order convergence is obtained. More precisely, for the largest value of the diffusivity (i.e. $\kappa = 0.01$), the convergence rate tends to second-order, whereas near-optimal convergence is obtained for moderate diffusion, with diffusivity $\kappa = 0.001$, resulting in third-order convergence for Cases 1 and 3 ($k = 2$), and second-order convergence for Case 2 ($k = 1$). In the advective limit, optimal convergence rates are obtained for Cases 2 and 3.

Case 3 and Case 4 only differ in the polynomial orders of the Lagrange multiplier space. However, the convergence rates drastically reduce for Case 4 to approximately second-order for the pure advection test (with $\kappa = 0$) and to approximately first order for the mixed advection-diffusion regime. This behavior illustrates the difference between the choice $l = 0$ compared to $l \geq 1$ in combination with the chosen time stepping scheme, as discussed in Section 3.6.2. The results for Case 5 and Case 6 finally demonstrate that the ℓ^2 and the PDE-constrained particle-mesh projection give similar results in terms of accuracy.

In short, it is observed that in the advective limit, optimal spatial convergence of order $k + 1$ is obtained provided that $l = 0$, and for (advection-)diffusion problems, the method tends to second-order accuracy.

3.8.2. Advection: rigid body rotation

In order to qualitatively assess the scheme's ability to preserve point and line singularities, and to demonstrate discrete conservation properties, the advection test proposed in [91] is considered next. This test comprises the rigid body rotation of a pointy cone (initially centered at $(x, y) = (-0.3, 0)$), a slotted disk (initially centered at $(x, y) = (0, -0.3)$), and a Gaussian hump (initially centered at $(x, y) = (0.15, 0.15)$) on the circular disk $\Omega := \{(x, y) | x^2 + y^2 \leq 0.5\}$. The velocity field is given by Eq. (3.47). By virtue of Eq. (3.33) it follows that in the advective limit the particle specific masses need not be updated, so that any discontinuities at the particle level persist. For this test case, three different configurations for the particle-mesh projection are considered: the local ℓ^2 projection (Case 1), the local ℓ^2 projection extended with bound constraints (Case 2), and the particle-mesh projection using the PDE-constrained optimization (Case 3), see Table 3.3. The polynomial orders are set to $k = 1$ (Case 1-2) or $(k, l) = (1, 0)$ (Case 3). The results are visually inspected and assessed in terms of conservation properties on

Table 3.3: Rigid body rotation: Overview of model settings.

	Projection Method	k	l	Remark
Case 1	ℓ^2	1	-	-
Case 2	ℓ^2	1	-	Bound constraint $0 \leq \psi_h \leq 1$ imposed
Case 3	PDE	1	0	-

two different mesh configurations. Mesh 1 is of moderate spatial resolution and consists of 16189 cells, Mesh 2 is of high spatial resolution and consists of 64561 cells. Approximately 30 particles are assigned per cell initially, resulting in a total number of slightly over 5×10^5 and 2×10^6 particles in total for Mesh 1 and Mesh 2, respectively. Time step sizes are chosen such to keep the CFL-number at an approximate value of 1 on both meshes, see Table 3.4, and a three-stage RK3 scheme is used for the particle advection. The mesh fields obtained for Case 2 and 3 on the coarse resolution mesh

Table 3.4: Rigid body rotation: Overview of mesh, particle and time step settings.

	$ \mathcal{T} $	$h_{K,\min}$	$h_{K,\max}$	$ \mathcal{S}_t $	Δt
Mesh 1	16189	7.9e-3	1.6e-2	502480	1e-2
Mesh 2	64561	4.0e-3	7.8e-2	2010783	5e-3

(Mesh 1) are visually assessed after a half and a full rotation in Fig. 3.1. Results for the mathematically simpler Case 1 are visually similar to results obtained for Case 3, and are hence omitted. Since the particle values are not updated, the initial discontinuities are maintained at the particle level. The shapes of the pointy cone and the Gaussian hump are well-preserved at the mesh level - and without numerical diffusion - both for a half

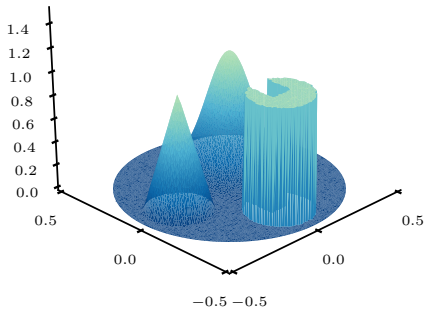
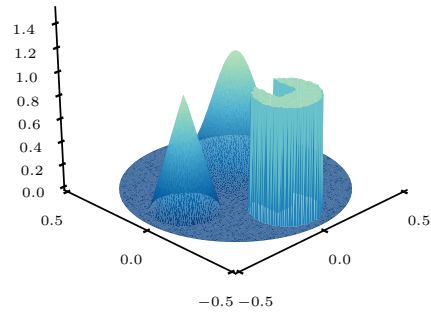
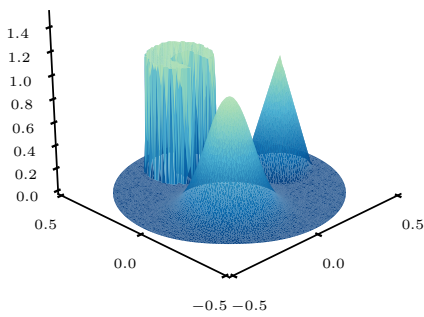
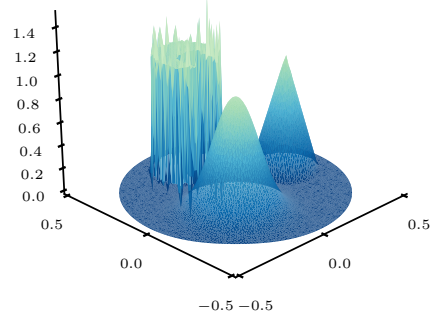
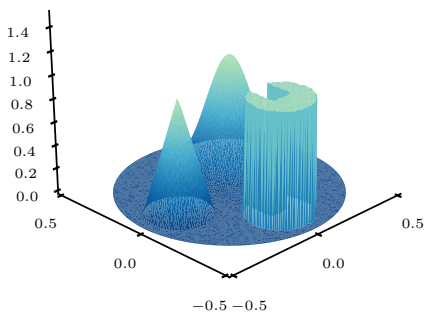
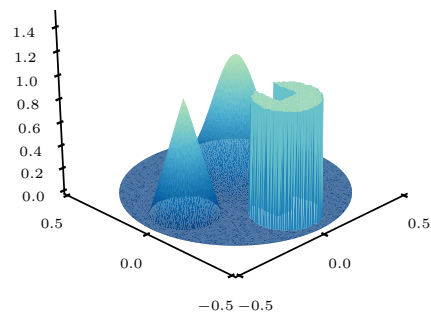
(a) Case 2, $t = 0$.(d) Case 3, $t = 0$.(b) Case 2, $t = 1$.(e) Case 3, $t = 1$.(c) Case 2, $t = 2$.(f) Case 3, $t = 2$.

Figure 3.1: Rigid body rotation: numerical solution ψ_h for particle-mesh projection based on bounded ℓ^2 projection (Case 2, left) and PDE-constrained particle-mesh projection (Case 3, right) with $(k, l) = (1, 0)$ at various time instants.

and a full rotation, for all cases. Despite this, localized overshoot is observed near the discontinuities for a half rotation in Case 3 (and so for Case 1). Rather than being a dispersion artifact, the over- and undershoot should be interpreted as a resolution issue with the mesh resolution being too coarse to capture the sharp discontinuity at the particle level monotonically in Case 3. This is clearly illustrated by Fig. 3.1f, showing that the initial condition is accurately recovered after a full rotation at $t = 2$, which indicates that the mesh-based solution is able to follow the moving particle field. By imposing a box constraint on the local ℓ^2 projection for Case 2, the solution respects the imposed bounds without exhibiting overshoot or undershoot, see Fig. 3.1b.

The L^2 -error after one rotation relative to the initial condition is investigated for all three cases. This is done for a coarse resolution run, in which the mesh contains 3987 cells and the total number of particles amounts to 125,764, the different cases considered correspond to those listed in Table 3.3. Important in the scope of this test is that two different particle advection schemes are used: the third-order accurate, three step Runge-Kutta scheme (RK3) and the fourth-order accurate, four step Runge-Kutta scheme (RK4). Furthermore, in Case 3 the β parameter is set to the low value of $1e-12$. Results are tabulated in Table 3.5.

Table 3.5: Rigid body rotation: error at the mesh after one full-rotation ($T = 2$) relative to the initial condition $\psi_h(0)$ for different particle advection schemes at a mesh containing 3987 cells and 125764 particles.

		Runge-Kutta 3		Runge-Kutta 4	
Δt		$\ \psi_h(0) - \psi_h(T)\ $	Rate	$\ \psi_h(0) - \psi_h(T)\ $	Rate
Case 1	0.04	1.1e-3	-	2.5e-5	-
	0.02	1.4e-4	3.0	1.6e-6	4.0
	0.01	1.8e-5	3.0	9.9e-8	4.0
	0.005	2.2e-6	3.0	6.1e-9	4.0
	0.0025	2.8e-7	3.0	2.1e-10	4.8
Case 2	0.04	1.0e-3	-	1.4e-4	-
	0.02	1.9e-4	2.5	1.4e-4	0.0
	0.01	1.4e-4	0.4	1.4e-4	0.0
	0.005	1.4e-4	0.0	1.4e-4	0.0
	0.0025	1.4e-4	0.0	1.4e-4	0.0
Case 3	0.04	1.1e-3	-	2.5e-5	4.0
	0.02	1.4e-4	3.0	1.6e-6	4.0
	0.01	1.8e-5	3.0	9.9e-8	4.0
	0.005	2.2e-6	3.0	6.1e-9	4.0
	0.0025	2.8e-7	3.0	5.9e-10	3.4

Clearly, for Case 1 and Case 3 the accuracy to which the initial condition is recovered after one rotation ($T = 2$) depends on the particle-advection scheme, resulting in an expected third-order convergence for the RK3 scheme, and fourth-order convergence for the RK4 scheme. For Case 3, a slightly lower convergence rate is observed for the small time step test, which owes to the small, but non-zero value for β . The accuracy to which the initial condition is recovered for Case 2, drastically reduces, and levels at an error

value of approximately $1.4\text{e-}4$ irrespective of time step and particle advection scheme. This behavior owes to the imposition of the box constraint in the local ℓ^2 projection which introduces a spatial error on top of the temporal error.

It is stressed that this test highlights one of the attractive features of the particle-mesh scheme in that an advected field is traced without loss of resolution, and recovered at a mesh with a high degree accuracy (Case 1 and 3). This is in contrast to existing purely mesh-based methods, which typically exhibit numerical diffusive results for this test [91–93].

Finally, the mass conservation errors for the three cases are investigated on the two different mesh configurations from Table 3.4. To this end, a measure for the relative global mass conservation error at time T is defined as

$$\epsilon_{\Delta\psi_{\Omega}} = \left| \frac{\int_{\Omega} (\psi_h(\mathbf{x}, T) - \psi_h(\mathbf{x}, 0)) \, \text{d}\Omega}{\int_{\Omega} \psi_h(\mathbf{x}, 0) \, \text{d}\Omega} \right|, \quad (3.49)$$

in which $\psi_h(\mathbf{x}, 0)$ and $\psi_h(\mathbf{x}, T)$ are the mesh related fields at time 0 and time T , respectively.

The local mass conservation error is investigated via the L^2 -norm of the time-discrete counterpart of the local conservation statement, Eq. (3.23). For the problem under consideration, this local mass conservation error norm at time level $n + 1$ is given by

$$\epsilon_{\Delta\psi_K} = \left(\sum_K \left(\int_K \frac{\psi_h^{n+1} - \psi_h^{*,n}}{\Delta t} \, \text{d}\Omega + \oint_{\partial K} \mathbf{a} \cdot \mathbf{n} \bar{\psi}_h^{n+1} \, \text{d}\Gamma \right)^2 \right)^{1/2}, \quad (3.50)$$

where it is noted that this conservation metric is only defined for the PDE-constrained particle-mesh projection (Case 3), since the flux variable $\bar{\psi}_h$ is defined for this case only.

The mass conservation errors as defined by Eqs. (3.49) and (3.50) after a half rotation ($T = 1$) and a full rotation ($T = 2$) are tabulated for the three cases in Table 3.6. Clearly, the particle-mesh projections based on the local ℓ^2 projection (Case 1 and 2) lack conservation, whereas global and local mass conservation of the PDE-constrained projection to machine precision are confirmed, irrespective of the resolution of the Eulerian mesh. With this result, it is ultimately concluded that the PDE-constrained particle-mesh projection (Case 3) reconciles global and local conservation properties with high-order accuracy.

3.8.3. Advection: advection skew to mesh

To assess the behavior of the scheme in the presence of steep gradients, and inflow and outflow boundary conditions, the advection of a discontinuity on the unit-square $\Omega := [0, 1]^2$ is considered for different transport velocities $\mathbf{a} = [\cos \alpha, \sin \alpha]^T$ with characteristic directions α of $15^\circ, 30^\circ, 45^\circ, 60^\circ$. A regular triangular mesh is used with uniform cell size $h_K = 1/25$, and each cell contains on average approximately 20 particles. Except for the case $\alpha = 45^\circ$, the flow directions are not aligned with the mesh. Dirichlet boundary conditions are prescribed at the inflow boundaries, and the specific

Table 3.6: Rigid body rotation: mass conservation errors (defined by Eqs. (3.49) and (3.50)) for different mesh configurations after a half rotation ($T = 1$) and a full rotation ($T = 2$).

		Mesh 1		Mesh 2	
		$T = 1$	$T = 2$	$T = 1$	$T = 2$
Case 1	$\epsilon_{\Delta\psi_{\Omega}}$	2.4e-5	1.6e-5	4.3e-6	2.0e-6
	$\epsilon_{\Delta\psi_K}$	-	-	-	-
Case 2	$\epsilon_{\Delta\psi_{\Omega}}$	1.3e-4	1.47e-4	1.45e-4	1.5e-4
	$\epsilon_{\Delta\psi_K}$	-	-	-	-
Case 3	$\epsilon_{\Delta\psi_{\Omega}}$	2.0e-16	2.0e-16	3.1e-15	5.9e-16
	$\epsilon_{\Delta\psi_K}$	1.7e-16	1.5e-16	1.3e-16	1.3e-16

mass ψ_p carried by the particles flowing into the domain is set accordingly. On the interior, particle specific masses need not updated for this pure advection problem. In this numerical example, as well as the last example of this chapter, the more complicated PDE-constrained projection is considered only. Furthermore, in the particle-mesh setting of this study, the test case can be used to assess the global conservation statement in the presence of inflow and outflow boundary conditions.

The fields ψ_h at $t = 2.0$ are plotted in Fig. 3.2. For a characteristic direction of 45° , the discontinuity is captured exactly at the mesh level. For the other characteristic directions, an overshoot is observed near the discontinuity. However, this overshoot remains strictly localized to one mesh cell upstream of the discontinuity. This behavior can be expected since no attempts are presently made to preserve monotonicity at the mesh, while the discontinuity at the particle level is inherently maintained without any diffusion. We leave the introduction of limiters as a fruitful area for future research, and refer to the work of Ten Eikelder on entropy-based discontinuity capturing [94], and the work of Bochev and coworkers on bound-preserving remaps as promising starting points [92].

The global mass conservation property of the scheme is verified by virtue of the time-discrete equivalent of Eq. (3.22). The global mass conservation error, denoted by $\epsilon_{\Delta\psi_{\Omega}}$, is the residual after subtracting the right-hand side from the left-hand side in this equation. The values for $\epsilon_{\Delta\psi_{\Omega}}$ thus obtained at $t = 2$ are tabulated in Table 3.7 for the four characteristic angles, confirming that global mass conservation is satisfied to machine precision.

Table 3.7: Advection skew to mesh: global mass conservation errors for different characteristic directions.

α	15°	30°	45°	60°
$\epsilon_{\Delta\psi_{\Omega}}$	-1.25e-15	-2.70e-15	-4.43e-15	-4.75e-15

3.8.4. Advection on a moving mesh

As a final example for this chapter, the formulation for the PDE-constrained projection on a moving mesh is assessed. As argued in Section 3.7, the PDE-constrained formulation allows a straightforward extension to moving meshes, despite the fact that the

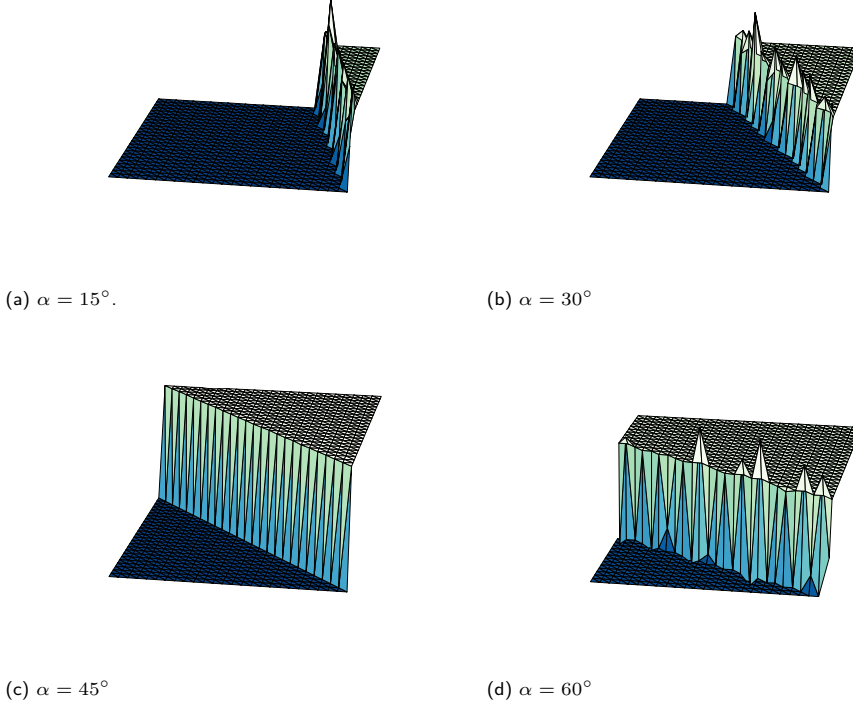


Figure 3.2: Advection skew to mesh: numerical solution ϕ_h on a unit square domain at $t = 2.0$ for different characteristic flow directions using polynomial orders $(k, l) = (1, 0)$.

mesh field needs to be stored at the moving mesh between consecutive time steps. Conservation properties were shown to be retained in the moving-mesh formulation and, for vanishing β , the DGCL condition is circumvented so that uniform fields remain uniform. To illustrate the latter assertion, this section particularly focuses on the influence of β .

To this end, let the unit square $\Omega := [0, 1] \times [0, 1]$ with an initially uniform mesh field $\psi_h^0 = 1$ and particle field $\psi_p^0 = 1$ be deformed by the time-dependent mesh velocity field

$$\hat{\mathbf{a}} = [g(t) \sin^2 \pi x \sin 2\pi y, -g(t) \sin^2 \pi y \sin 2\pi x]^\top, \quad (3.51)$$

which yields $\hat{\mathbf{a}} = \mathbf{0}$ on the boundary $\partial\Omega$ by construction. Furthermore, $g(t)$ is a time-periodic function, defined as

$$g(t) = 0.5 \cos \pi t. \quad (3.52)$$

This mesh velocity results in a time-periodic motion of the mesh with a period $T = 2$. For the time being, the Eulerian velocity \mathbf{a} is set to $\mathbf{0}$. Furthermore, the domain is partitioned into $32 \times 32 \times 2$ regular triangles, and the time step is set to 0.025. The mesh motion is determined by integrating Eq. (3.51) in time using a first-order Euler forward method.

To verify whether uniform fields remain uniform on the moving mesh, the mesh solution at maximum deformation is assessed for a range of β -values, listed in Table 3.8.

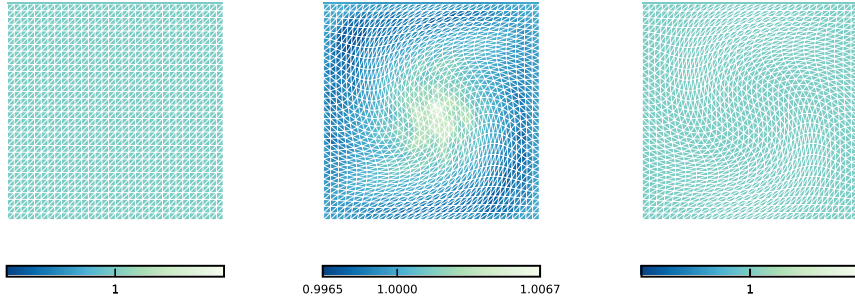


Figure 3.3: Advection on moving mesh: initially constant field at $t = 0$ (left), solution at maximum mesh deformation at $t = 0.5$ using $\beta = 1e0$ (middle) and $1e-12$ (right). The bounds of the color bar indicate the minimum and maximum value.

Table 3.8: L^2 -error at maximum deformation ($t = 0.5$) for different values of β .

β	$\ \psi_h^0 - \psi_h^{n+1}\ $
1e-0	1.37e-3
1e-2	1.91e-5
1e-4	1.93e-7
1e-6	1.94e-9
1e-8	1.96e-11
1e-10	1.96e-13
1e-12	4.27e-15

The results are visualized in Fig. 3.3 for a β value of 1 and a β value of $1e-12$. For the high value for β the field deviates from the uniform and constant initial condition, whereas for the small β value there is no visual difference between the field at maximum deformation and the initial condition, thus indicating that uniform fields remain uniform on moving meshes. To back this observation quantitatively, the L^2 -errors for different values of β are tabulated in Table 3.8. For the larger values of β , this error is relatively large, whereas this error approximates machine zero for the smaller β values. This provides an additional motivation of keeping the β value low, in line with the observation made in Section 3.8.2.

The scheme's ability to capture combined mesh motion and Eulerian advection is finally illustrated in Fig. 3.4. The mesh motion $\hat{\mathbf{a}}$ is given by Eq. (3.51), whereas the Eulerian advection \mathbf{a} is given by $-\hat{\mathbf{a}}$. The advected field is the cosine shaped pulse

$$\psi_h(0) = \begin{cases} 1 + \cos \pi r & \text{if } r \leq R, \\ 0 & \text{otherwise,} \end{cases} \quad (3.53)$$

in which the radius $R = 0.25$ and the radial coordinate $r := \sqrt{(x - x_c)^2 + (y - y_c)^2}$,

with the pulse initially being centered at $(x_c, y_c) = (0.25, 0.5)$. Furthermore, the amplitude of the pulse equals 2, i.e. $0 \leq \psi_h(0) \leq 2$. The domain is triangulated using $32 \times 32 \times 2$ regular triangular cells, on which a safe number of 50 particles per cell are seeded in order to avoid cells becoming empty as a result of the deforming cells. Furthermore, $(k, l) = (1, 0)$ and $\beta = 1e-8$ and the time step is set to $\Delta t = 0.025$.

At maximum mesh deformation and maximum distortion of the pulse ($t = 0.5$), the values of the reconstructed field ψ_h stay closely within this range, and visually, the initial shape of the pulse appears to be accurately recovered at $t = 1.0$. From the solution at $t = 1.0$ it can also be seen that initial mesh configuration is not exactly recovered. This is attributed to the first-order accurate mesh-motion, using the explicit Euler forward method.

Table 3.9 confirms global and local momentum conservation at the time of maximum distortion, $t = 0.5$, and at $t = 1$. In addition, the minimum and maximum values for $\bar{\psi}_h$ and ψ_h are shown, with these values staying relatively close to the expected range $0 \leq \psi \leq 2$, particularly so for the maximum values. The somewhat larger deviations for the minimum values are attributed to the cut-off in the cosine hump at $r = R$, see Eq. (3.53).

Table 3.9: Advection on moving mesh: global- ($\epsilon_{\Delta\psi_\Omega}$) and local mass conservation error ($\epsilon_{\Delta\psi_K}$) at $t = 0.5$ and $t = 1$. Also shown are $\bar{\psi}_{h,\min}$, $\bar{\psi}_{h,\max}$ and $\psi_{h,\min}$, $\psi_{h,\max}$.

	$\epsilon_{\Delta\psi_\Omega}$	$\epsilon_{\Delta\psi_K}$	$\bar{\psi}_{h,\min}$	$\bar{\psi}_{h,\max}$	$\psi_{h,\min}$	$\psi_{h,\max}$
$t = 0.5$	2.8e-17	1.0e-16	-0.09	2.04	-0.14	2.07
$t = 1.0$	1.8e-17	1.1e-16	-0.13	2.02	-0.04	2.03

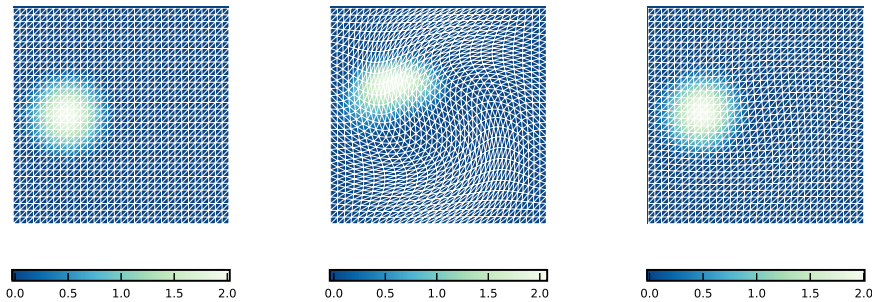


Figure 3.4: Advection on moving mesh: initial solution at $t = 0$ (left), solution at maximum mesh deformation at $t = 0.5$ (middle), solution at $t = 1.0$ (right). The bounds of the colorbar are given by the minimum and maximum value.

3.9. Conclusion

In this chapter a hybrid particle-mesh method has been formulated for the scalar advection-diffusion problem. The developed method is cast into an operator splitting scheme in

which Lagrangian particles are used to discretize an advection operator and an Eulerian HDG approach is employed for the diffusive part. The coupling between the Lagrangian particles and the Eulerian mesh is established by means of two auxiliary projection steps.

Variational formulations for these projection steps have been presented. Exploiting a (hybridized) discontinuous Galerkin framework, two variants for the particle-mesh projection operator were derived. One variant is to project the scalar particle property onto a mesh field via local (i.e. cellwise) ℓ^2 projections. An extension of this approach with bound constraints was illustrated, thus enabling a bound preserving particle-mesh projection.

To overcome the compromise on exact conservation of the local ℓ^2 particle-mesh projections, a PDE-constrained particle-mesh projection has been proposed. The key idea in formulating the constraint is that from a mesh-perspective the particle motion must satisfy an advection operator. By expressing the control variable of the resulting optimality system in terms of single-valued functions at cell interfaces, the hybridized discontinuous Galerkin method naturally provides the necessary ingredients for formulating the optimality system. Consistency and conservation of the constrained interaction were proven.

A fully-discrete particle-mesh operator splitting was proposed by presenting a specific combination of the four components constituting the particle-mesh scheme. Second-order overall time accuracy can be obtained using the presented time stepping scheme. To obtain this, consistency requirements were derived and verified, paying particular attention to the PDE-constrained particle-mesh projection. An analysis of the resulting discrete optimality system revealed that a specific choice for the Lagrange multiplier field renders the PDE-constrained optimization procedure independent of the time integration method, and results in a particularly attractive and robust scheme. In addition it was shown that the PDE-constrained particle-mesh projections admits a straightforward formulation on moving meshes.

A range of numerical experiments demonstrates the potential of the presented approach. Standard convergence rates in space and second-order convergence rates in time were confirmed for a range of examples for the linear advection-diffusion problem. Most notably, the results for a pure advection problem showed that the developed hybrid particle-mesh strategy eliminates numerical diffusion, without compromising accuracy and exact conservation. The method turns out to be promising, albeit ensuring monotonicity in the conservative PDE-constrained projection remains an open challenge.

4

Incompressible flow problems

This chapter extends the particle-mesh concepts explored in the previous chapter to the incompressible Navier-Stokes equations. In order to do so, specific attention is paid to the non-linearity of the advection term and it is shown that the proposed method circumvents stabilization of this term. In addition, it is demonstrated that the discretization of the incompressibility constraint is of utmost importance for maintaining a correct particle distribution.

A range of numerical examples shows that optimal convergence rates are obtained in space, and second-order accuracy is obtained in time given the particular time stepping strategy. The model capabilities are further demonstrated by presenting results for a Taylor-Green vortex instability in the high Reynolds number regime, the flow over a backward facing step and for the flow around a cylinder, which can be considered challenging benchmarks for particle-based schemes.

4.1. Model problem

As for the advection-diffusion equation in the previous chapter, consider the domain $\Omega \subset \mathbb{R}^d$, with $d = 2, 3$ the spatial dimension. The boundary $\Gamma = \partial\Omega$ is partitioned into complementary Dirichlet (Γ_D) and Neumann (Γ_N) parts, and \mathbf{n} is the outward pointing unit normal to Γ . Furthermore, $I = (t^0, t^N]$ is the time interval of interest, with t^0 and t^N the start and end time of the simulation.

Material presented in this chapter is partly based on:

[95]: A hybridized discontinuous Galerkin framework for high-order particle-mesh operator splitting of the incompressible Navier-Stokes equations, J.M. Maljaars et al. In: *Journal of Computational Physics* **358**, 150-172, 2018.

[77]: Conservative, high-order particle-mesh scheme with applications to advection-dominated flows, J.M. Maljaars et al. In: *Computer Methods in Applied Mechanics and Engineering*, **348**, 443-465, 2019.

4.1.1. Governing equations

On the space-time domain $\Omega \times I$ the incompressible Navier-Stokes equations are stated as follows: given the kinematic viscosity ν , the forcing term $\mathbf{f} : \Omega \times I \rightarrow \mathbb{R}^d$, the boundary conditions $\mathbf{g} : \Gamma_D \times I \rightarrow \mathbb{R}^d$ and $\mathbf{h} : \Gamma_N \times I \rightarrow \mathbb{R}^d$, and a solenoidal initial condition $\mathbf{u}_0 : \Omega \rightarrow \mathbb{R}^d$, find the velocity field $\mathbf{u} : \Omega \times I \rightarrow \mathbb{R}^d$ and the pressure field $p : \Omega \times I \rightarrow \mathbb{R}$ such that

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega \times I, \quad (4.1a)$$

$$\boldsymbol{\sigma} = \mathbf{u} \otimes \mathbf{u} + p\mathbf{I} - 2\nu \nabla^s \mathbf{u} \quad \text{in } \Omega \times I, \quad (4.1b)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times I, \quad (4.1c)$$

$$\boldsymbol{\sigma} \mathbf{n} = (1 - \gamma)(\mathbf{u} \otimes \mathbf{u}) \mathbf{n} + \mathbf{h} \quad \text{on } \Gamma_N \times I, \quad (4.1d)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_D \times I, \quad (4.1e)$$

$$\mathbf{u}(\mathbf{x}, t^0) = \mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega, \quad (4.1f)$$

where $\boldsymbol{\sigma}$ is the diffusive momentum flux tensor, \mathbf{I} is the identity tensor, and $\nabla^s \mathbf{u} = \frac{1}{2} \nabla(\mathbf{u}) + \frac{1}{2} \nabla(\mathbf{u})^\top$ is the symmetric velocity-gradient tensor. The factor γ is equal to one on inflow parts of Γ_N (that is, where $\mathbf{u} \cdot \mathbf{n} < 0$), imposing in this way the total momentum flux. On the outflow parts of Γ_N (where $\mathbf{u} \cdot \mathbf{n} \geq 0$), the factor γ equals zero which prescribes the diffusive part of the momentum flux only.

4.1.2. Operator-splitting

As in the previous chapter, the particle-mesh method is conceived as an operator splitting procedure on the sequence of $N+1$ discrete time levels $\{t^0, t^1, \dots, t^{N-1}, t^N\}$, where the half-open subinterval between time level t^n and t^{n+1} is indicated with $I_n = (t^n, t^{n+1}]$, and $\mathcal{I} := \{I_n\}$ is the ordered sequence of subintervals. Furthermore, let the total flux - given by Eq. (4.1b) - be decomposed into advective parts $\boldsymbol{\sigma}_a$ and diffusive parts $\boldsymbol{\sigma}_d$.

A spatiotemporal operator splitting procedure for the Navier-Stokes reads: find a vector-valued field $\mathbf{v} : \Omega \times I_n \rightarrow \mathbb{R}^d$ satisfying an advection problem,

$$\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot \boldsymbol{\sigma}_a = \mathbf{0} \quad \text{in } \Omega \times I_n, \quad (4.2a)$$

$$\boldsymbol{\sigma}_a = \mathbf{v} \otimes \mathbf{U} \quad \text{in } \Omega \times I_n, \quad (4.2b)$$

$$\boldsymbol{\sigma}_a \mathbf{n} = (1 - \gamma)(\mathbf{v} \otimes \mathbf{U}) \mathbf{n} + \gamma \mathbf{h}_a \quad \text{on } \Gamma_N \times I_n, \quad (4.2c)$$

$$\mathbf{v} = \mathbf{g} \quad \text{on } \Gamma_D^- \times I_n, \quad (4.2d)$$

$$\mathbf{v}(\mathbf{x}, t^n) = \mathcal{P}_L(\mathbf{u}(\mathbf{x}, t^n)) \quad \text{in } \Omega, \quad (4.2e)$$

in which Γ_D^- indicates inflow parts on the Dirichlet boundary, and a velocity field $\mathbf{u} :$

$\Omega \times I_n \rightarrow \mathbb{R}^d$ satisfying an incompressible Stokes problem,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\sigma}_d = \mathbf{f} \quad \text{in } \Omega \times I_n, \quad (4.3a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times I_n, \quad (4.3b)$$

$$\boldsymbol{\sigma}_d = p\mathbf{I} - 2\nu\nabla^s \mathbf{u} \quad \text{in } \Omega \times I_n, \quad (4.3c)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_D \times I_n, \quad (4.3d)$$

$$\boldsymbol{\sigma}_d \mathbf{n} = \mathbf{h}_d \quad \text{on } \Gamma_N \times I_n, \quad (4.3e)$$

$$\mathbf{u}(\mathbf{x}, t^n) = \mathcal{P}_E(\mathbf{v}(\mathbf{x}, t^{n+1})) \quad \text{in } \Omega, \quad (4.3f)$$

with notation similar to Eqs. (3.2-3.3). In particular, since the advection problem Eq. (4.2) is solved using Lagrangian particles, the mesh-particle projection \mathcal{P}_L provides the condition to advance the advection problem from t^n to t^{n+1} . The particle-mesh projection \mathcal{P}_E reconstructs a specific momentum field at the mesh from the scattered particle data, thereby providing the initial condition for advancing the incompressible Stokes equations in time. Furthermore, the advective field \mathbf{U} in Eq. (4.2b) is not yet specified, other than to require this field to be a consistent approximation to \mathbf{u} , which is piecewise constant on every subinterval $I_n \in \mathcal{I}$.

This chapter extends the strategies for the particle-mesh projections from Chapter 3 to the vector-valued setting, and pays specific attention to the relation between the discrete incompressibility constraint and the particle advection.

4.1.3. Auxiliary definitions

As in Chapter 3, the spatial domain Ω is triangulated into a set $\mathcal{T} := \{K\}$ of open, non-overlapping cells K . A measure of the cell size is denoted by h_K , and the outward pointing unit normal vector on the boundary ∂K of each cell is denoted by \mathbf{n} . The closure of a cell is denoted by $\bar{K} = K \cup \partial K$. Adjacent cells K_i and K_j ($i \neq j$) share a common facet $F = \partial K_i \cap \partial K_j$. The set of all facets (including the exterior boundary facets $F = \partial K \cap \partial \Omega$) is denoted by \mathcal{F} .

Function spaces

Throughout this chapter, the vector-valued equivalents of Eq. (3.4 - 3.6) are used, that is

$$\mathbf{W}_h := \left\{ \mathbf{w}_h \in [L^2(\mathcal{T})]^d, \mathbf{w}_h|_K \in [P_k(K)]^d \forall K \in \mathcal{T} \right\}, \quad (4.4)$$

$$\mathbf{T}_h := \left\{ \boldsymbol{\tau}_h \in [L^2(\mathcal{T})]^d, \boldsymbol{\tau}_h|_K \in [P_l(K)]^d \forall K \in \mathcal{T} \right\}, \quad (4.5)$$

$$\bar{\mathbf{W}}_{h,g} := \left\{ \bar{\mathbf{w}}_h \in [L^2(\mathcal{F})]^d, \bar{\mathbf{w}}_h|_F \in [P_k(F)]^d \forall F \in \mathcal{F}, \bar{\mathbf{w}}_h = \mathbf{g} \text{ on } \Gamma_D \right\}, \quad (4.6)$$

in which $P_{k,l}(K)$ denotes the set of polynomials of degree $(k, l) \geq (1, 0)$ on $K \in \mathcal{T}$, and $P_k(F)$ the set of polynomials of degree k on $F \in \mathcal{F}$. The function spaces \mathbf{W}_h and $\bar{\mathbf{W}}_{h,g}$ are used for the discretization of the velocity in the incompressible Navier-Stokes equations, Eq. (4.1). On top of that, these function spaces will be used in conjunction with the function space \mathbf{T}_h to formulate the PDE-constrained particle-mesh projection.

Function spaces required for the discretization of the pressure are defined as

$$Q_h := \{q_h \in L^2(\mathcal{T}), q_h|_K \in P_{k-1}(K) \forall K \in \mathcal{T}\}, \quad (4.7)$$

$$\bar{Q}_h := \{\bar{q}_h \in L^2(\mathcal{F}), \bar{q}_h|_F \in P_k(F) \forall F \in \mathcal{F}\}, \quad (4.8)$$

where it is emphasized that the function space Q_h is one order lower than the facet function space \bar{Q}_h . Furthermore, the extension of Q_h to cell boundaries ∂K is assumed implicit in the definition.

To enforce the incompressibility constraint in the Navier-Stokes equations (Eq. (4.1)), the following combinations of velocity and pressure spaces can be considered for the discretization of the unsteady Stokes equations [72]:

$$M_h := \begin{cases} \mathbf{W}_h \times \bar{\mathbf{W}}_h \times Q_h \times \bar{Q}_h & \text{HDG} & (4.9a) \\ \mathbf{W}_h \times (\bar{\mathbf{W}}_h \cap C^0(\mathcal{F})) \times Q_h \times \bar{Q}_h & \text{EDG-HDG} & (4.9b) \\ \mathbf{W}_h \times (\bar{\mathbf{W}}_h \cap C^0(\mathcal{F})) \times Q_h \times (\bar{Q}_h \cap C^0(\mathcal{F})) & \text{EDG} & (4.9c) \end{cases}$$

Historically, the embedded discontinuous Galerkin (EDG) is the oldest of the three, dating back to work by Labeur and Wells in 2007 [3] and further specialized by these authors in [6]. This variant employs velocity and pressure facet functions that are continuous, and yields velocity fields that are pointwise divergence-free within cells. The HDG variant was presented in a 2018 paper by Rhebergen and Wells [8], and uses discontinuous facet function spaces. This gives the advantage over the EDG formulation from [6] in that computed velocity fields are not only pointwise divergence-free within cells, but also possess continuity of the normal component across cell facets, which renders the method $H(\text{div})$ -conforming. This $H(\text{div})$ -conformity is retained in the EDG-HDG formulation, for which the facet velocity field is continuous and the facet pressure field is discontinuous, thereby reducing the number of dofs compared to the HDG variant [72].

An E/HDG framework is adopted to allow a unified formulation for the discretization of the unsteady Stokes equations and the particle-mesh projections. In the scope of the incompressible Navier-Stokes equations, the excellent (EDG), or exact (HDG and EDG-HDG) divergence properties of the resulting velocity fields further motivate the choice for such a framework. Since the formulations presented in the sequel are generic for all function space combinations from Eq. (4.9), no choice is made for either one of the three function space combinations at this stage.

Particle definitions

The Lagrangian particle configuration in the domain Ω at a fixed time instant t is denoted by χ_t , as specified in Eq. (3.7). The particle index set \mathcal{S}_t , and the index set \mathcal{S}_t^K of particles hosted by cell K , at a fixed time instant t are defined by Eq. (3.8) and Eq. (3.9), respectively. Furthermore, a Lagrangian vector field on the particles is defined as

$$\mathcal{V}_t := \{\mathbf{v}_p(t) \in \mathbb{R}^d, \forall p \in \mathcal{S}_t\}, \quad (4.10)$$

where $\mathbf{v}_p(t)$ denotes the vector quantity associated with particle p .

Importantly, subscripts p and h are used consistently throughout to distinguish between Lagrangian particle data and Eulerian mesh fields.

4.2. Semi-discrete formulations

A particle-mesh operator splitting procedure for the incompressible Navier-Stokes equations involves a similar sequence of steps as for the advection-diffusion equation:

1. *Lagrangian discretization of the advection problem*, in order to solve Eqs. (4.2a - 4.2d) at the particle level;
2. *particle-mesh projection*, in order to project the particle velocities onto a velocity field on the Eulerian mesh using the operator $\mathcal{P}_E : \mathcal{V}_t \rightarrow \mathbf{W}_h$ (see Eq. (4.3f));
3. *Eulerian discretization of the unsteady Stokes equations*, in order to solve Eqs. (4.3a - 4.3e) on the mesh;
4. *mesh-particle projection*, in order to update the particle velocities from the velocity field on the Eulerian mesh using the operator $\mathcal{P}_L : \mathbf{W}_h \rightarrow \mathcal{V}_t$ (see Eq. (4.2e)).

4.2.1. Lagrangian discretization of the particle advection

The particle advection step for the incompressible Navier-Stokes equations proceeds along similar lines as for the linear advection-diffusion equation (Eq. (3.11)). Particle specific momenta and positions are updated by integrating

$$\dot{\mathbf{v}}_p(t) = \mathbf{0} \quad \forall p \in \mathcal{S}_t, \quad (4.11)$$

$$\dot{\mathbf{x}}_p(t) = \mathbf{U}_h(\mathbf{x}_p(t), t^n) \quad \forall p \in \mathcal{S}_t, \quad (4.12)$$

over the time interval I_n . In this equation $\mathbf{x}_p(t)$ denotes the particle position at time $t \in I_n$. We still leave the advective field $\mathbf{U}_h \in \mathbf{W}_h$ unspecified, but recall that it should be a consistent approximation of \mathbf{u}_h , which is piecewise constant in time and explicitly known on every $I_n \in \mathcal{I}$ so as to linearize the non-linear advection term. Similar to the scalar-valued case, \mathbf{v}_p stays constant throughout the advection stage by virtue of Eq. (4.11).

4.2.2. Particle-mesh projection

The formulation of the vector-valued particle-mesh projections $\mathcal{P}_E : \mathcal{V}_t \rightarrow \mathbf{W}_h$, proceeds analogous to the particle-mesh projections for scalar-valued fields, detailed in Section 3.3. The most notable difference relates to the advection term in the PDE-constrained projection.

Local ℓ^2 projection

The local ℓ^2 projection of the specific momenta carried by the particles onto a mesh field is based on the vector-valued counterpart of the functional in Eq. (3.13), and reads

$$J := \sum_{p \in \mathcal{S}_t} \frac{1}{2} \|\mathbf{v}_h(\mathbf{x}_p(t), t) - \mathbf{v}_p(t)\|^2, \quad (4.13)$$

with $\|\cdot\|^2$ denoting the square of the Euclidean norm, i.e. $\|\mathbf{v}\|^2 = \mathbf{v} \cdot \mathbf{v}$.

Taking variations and replacing the variations by test functions test functions $\mathbf{w}_h \in \mathbf{W}_h$ results in

$$\sum_K \sum_{p \in \mathcal{S}_t^K} (\mathbf{v}_h(\mathbf{x}_p(t), t) - \mathbf{v}_p(t)) \cdot \mathbf{w}_h(\mathbf{x}_p(t)) = 0 \quad \forall \mathbf{w}_h \in \mathbf{W}_h, \quad (4.14)$$

which can be solved in a cellwise fashion since $(\mathbf{v}_h, \mathbf{w}_h) \in \mathbf{W}_h$ are double-valued at facets. Hence, the mapping of the particle data to the mesh basis functions only involves the inversion of small, local matrices, where this mapping has a regular solution if and only if the unsolvency criterion is met, see Section 3.3.1.

Conservative PDE-constrained projection

In order to ensure momentum conservation over the particle-mesh projection, a vector-valued PDE-constrained projection operator $\mathcal{P}_E : \mathcal{V}_t \rightarrow \mathbf{W}_h$ is formulated. The formulation of this operator proceeds along similar lines as for the linear advection-diffusion equation. That is, the objective functional constituting the least-squares minimization problem is augmented with Lagrange multiplier terms that enforce weak satisfaction of the momentum advection equation by the projected field \mathbf{v}_h . Integration by parts of the constraint introduces a facet-based variable $\bar{\mathbf{v}}_h$ controlling the flux across cell interfaces. This results in the following Lagrangian functional describing the PDE-constrained optimization problem for the (specific) momentum projection:

$$\begin{aligned} \mathcal{L}(\mathbf{v}_h, \bar{\mathbf{v}}_h, \boldsymbol{\lambda}_h) &= \sum_{p \in \mathcal{S}_t} \frac{1}{2} \|\mathbf{v}_h(\mathbf{x}_p(t), t) - \mathbf{v}_p(t)\|^2 + \sum_K \oint_{\partial K} \frac{1}{2} \beta \|\bar{\mathbf{v}}_h - \mathbf{v}_h\|^2 d\Gamma \\ &+ \int_{\Omega} \frac{\partial \mathbf{v}_h}{\partial t} \cdot \boldsymbol{\lambda}_h d\Omega - \sum_K \int_K \mathbf{v}_h \otimes \mathbf{U}_h \cdot \nabla \boldsymbol{\lambda}_h d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} (\bar{\mathbf{v}}_h \otimes \bar{\mathbf{U}}_h) \mathbf{n} \cdot \boldsymbol{\lambda}_h d\Gamma \\ &\quad + \oint_{\Gamma_N} (1 - \gamma) (\mathbf{v}_h \otimes \mathbf{U}_h) \mathbf{n} \cdot \boldsymbol{\lambda}_h d\Gamma + \oint_{\Gamma_N} \gamma \mathbf{h}_a \cdot \boldsymbol{\lambda}_h d\Gamma \quad (4.15) \end{aligned}$$

By taking variations with respect to $(\mathbf{v}_h, \boldsymbol{\lambda}_h, \bar{\mathbf{v}}_h) \in (\mathbf{W}_h, \mathbf{T}_h, \bar{\mathbf{W}}_h)$, the resulting optimality system becomes: at time $t \in I_n$, given the initial condition $\mathbf{v}_h^n \in \mathbf{W}_h$, the particle field $\mathbf{v}_p(t) \in \mathcal{V}_t$, the advective field $(\mathbf{U}_h, \bar{\mathbf{U}}_h) \in (\mathbf{W}_h, \bar{\mathbf{W}}_h)$ and the advective Neumann boundary condition $\mathbf{h}_a : \Gamma_N \rightarrow \mathbb{R}^d$, find $(\mathbf{v}_h, \boldsymbol{\lambda}_h, \bar{\mathbf{v}}_h) \in (\mathbf{W}_h, \mathbf{T}_h, \bar{\mathbf{W}}_{h,g})$ such that

$$\begin{aligned} &\sum_{p \in \mathcal{S}_t} (\mathbf{v}_h(\mathbf{x}_p(t), t) - \mathbf{v}_p) \cdot \delta \mathbf{v}_h(\mathbf{x}_p(t)) - \sum_K \oint_{\partial K} \beta (\bar{\mathbf{v}}_h - \mathbf{v}_h) \cdot \delta \mathbf{v}_h d\Gamma + \int_{\Omega} \frac{\partial \delta \mathbf{v}_h}{\partial t} \cdot \boldsymbol{\lambda}_h d\Omega \\ &- \sum_K \int_K (\delta \mathbf{v}_h \otimes \mathbf{U}_h) : \nabla \boldsymbol{\lambda}_h d\Omega + \oint_{\Gamma_N} (1 - \gamma) (\delta \mathbf{v}_h \otimes \mathbf{U}_h) \mathbf{n} \cdot \boldsymbol{\lambda}_h d\Gamma = 0 \quad \forall \delta \mathbf{v}_h \in \mathbf{W}_h, \end{aligned} \quad (4.16a)$$

$$\begin{aligned} &\int_{\Omega} \frac{\partial \delta \mathbf{v}_h}{\partial t} \cdot \delta \boldsymbol{\lambda}_h d\Omega - \sum_K \int_K (\mathbf{v}_h \otimes \mathbf{U}_h) : \nabla \delta \boldsymbol{\lambda}_h d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} (\bar{\mathbf{v}}_h \otimes \bar{\mathbf{U}}_h) \mathbf{n} \cdot \delta \boldsymbol{\lambda}_h d\Gamma \\ &+ \oint_{\Gamma_N} (1 - \gamma) (\mathbf{v}_h \otimes \mathbf{U}_h) \mathbf{n} \cdot \delta \boldsymbol{\lambda}_h d\Gamma + \oint_{\Gamma_N} \gamma \mathbf{h}_a \cdot \delta \boldsymbol{\lambda}_h d\Gamma = 0 \quad \forall \delta \boldsymbol{\lambda}_h \in \mathbf{T}_h, \end{aligned} \quad (4.16b)$$

$$\sum_K \oint_{\partial K \setminus \Gamma_N} (\delta \bar{\mathbf{v}}_h \otimes \bar{\mathbf{U}}_h) \mathbf{n} \cdot \boldsymbol{\lambda}_h \, d\Gamma + \sum_K \oint_{\partial K} \beta (\bar{\mathbf{v}}_h - \mathbf{v}_h) \cdot \delta \bar{\mathbf{v}}_h \, d\Gamma = 0 \quad \forall \delta \bar{\mathbf{v}}_h \in \bar{\mathbf{W}}_{h,0}. \quad (4.16c)$$

Worth mentioning is the similarity between Eq. (4.16) with Eq. (3.17) for the linear scalar advection problem. This owes to the use of the advective fields \mathbf{U}_h and $\bar{\mathbf{U}}_h$ which linearizes the momentum fluxes, thereby simplifying the optimality system considerably. It is also pointed out that the advective fluxes are discretized without any stabilization mechanisms.

The PDE-constrained projection step leads to a field $\mathbf{v}_h \in \mathbf{W}_h$ to be used for advancing the subsequent unsteady Stokes problem in time.

4.2.3. Eulerian discretization of the Stokes problem

Of the various HDG schemes for the (Navier-)Stokes equations presented in literature (see, e.g., [4, 6, 96]), this study employs the HDG method proposed by Labeur and Wells [6] and further specialized by Rhebergen and Wells [72] for the spatial discretization of the unsteady Stokes problem (Eqs. (4.3a-4.3e)). A brief summary of this method is provided below. For an in-depth description and analysis of the method, reference is made to the aforementioned papers as well as [97].

Splitting the equations into local (i.e. cellwise) and global balances is central to the HDG method: the local equations represent cellwise balances and the global equations are formulated by enforcing weakly the continuity of the mass and momentum fluxes across cell facets. The local balances on a cell are linked to the local balances on neighboring cells only via globally defined facet functions, thus circumventing a direct interaction between neighboring cells.

Starting with the mass balance equations, a Galerkin approximation of the local mass balance (i.e. the incompressibility constraint, given by Eq. (4.3b)), requires the local velocity field $\mathbf{u}_h \in \mathbf{W}_h$ to satisfy

$$\sum_K \int_K \mathbf{u}_h \cdot \nabla q_h \, d\Omega - \sum_K \oint_{\partial K} \mathbf{u}_h \cdot \mathbf{n} \, q_h \, d\Gamma = 0 \quad \forall q_h \in Q_h, \quad (4.17)$$

in which $p_h \in Q_h$ is the local pressure field.

The global mass balance enforces the flux $\mathbf{u}_h \cdot \mathbf{n}$ to be at least weakly continuous across cell facets, i.e.

$$\sum_K \oint_{\partial K} \mathbf{u}_h \cdot \mathbf{n} \, \bar{q}_h \, d\Gamma - \oint_{\partial \Omega} \bar{\mathbf{u}}_h \cdot \mathbf{n} \, \bar{q}_h \, d\Gamma = 0 \quad \forall \bar{q}_h \in \bar{Q}_h, \quad (4.18)$$

where $\bar{\mathbf{u}}_h \in \bar{\mathbf{W}}_{h,g}$ is the single-valued velocity defined at the cell facets.

A local Galerkin approximation to the momentum balance, Eq. (4.3a), can be formu-

lated as follows: find $\mathbf{u}_h \in \mathbf{W}_h$ such that

$$\begin{aligned} \int_{\Omega} \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{w}_h \, d\Omega - \sum_K \int_K \boldsymbol{\sigma}_h : \nabla \mathbf{w}_h \, d\Omega + \sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_h \mathbf{n} \cdot \mathbf{w}_h \, d\Gamma \\ + \sum_K \oint_{\partial K} 2\nu (\bar{\mathbf{u}}_h - \mathbf{u}_h) \cdot \nabla^s \mathbf{w}_h \mathbf{n} \, d\Gamma = \int_{\Omega} \mathbf{f} \cdot \mathbf{w}_h \, d\Omega \quad \forall \mathbf{w}_h \in \mathbf{W}_h, \end{aligned} \quad (4.19)$$

given the external forcing \mathbf{f} , the kinematic viscosity ν , the facet velocity $\bar{\mathbf{u}}_h \in \bar{\mathbf{W}}_{h,g}$ and the pressures $p_h \in Q_h$ and $\bar{p}_h \in \bar{Q}_h$. Furthermore, the momentum flux $\boldsymbol{\sigma}_h$ is given by

$$\boldsymbol{\sigma}_h = p_h \mathbf{I} - 2\nu \nabla^s \mathbf{u}_h, \quad (4.20)$$

and a numerical momentum flux $\hat{\boldsymbol{\sigma}}_h$ on cell facets is defined as

$$\hat{\boldsymbol{\sigma}}_h = p_h \mathbf{I} - 2\nu \nabla^s \mathbf{u}_h - 2 \frac{\alpha}{h_K} \nu (\bar{\mathbf{u}}_h - \mathbf{u}_h) \otimes \mathbf{n}, \quad (4.21)$$

where α is a stability parameter, as is typical of interior penalty methods [68]. As with the pressure-stabilizing term, the penalty term is local in that neighboring cells do not directly interact with each other, but only via the facet functions.

Similar to the global continuity equation, Eq. (4.18), a global momentum equation is furnished, requiring the numerical momentum flux $\hat{\boldsymbol{\sigma}}_h$ to be weakly continuous across cell facets. This results in the constraint

$$\sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_h \mathbf{n} \cdot \bar{\mathbf{w}}_h \, d\Gamma = \oint_{\Gamma_N} \mathbf{h}_d \cdot \bar{\mathbf{w}}_h \, d\Gamma \quad \forall \bar{\mathbf{w}}_h \in \bar{W}_{h,0}, \quad (4.22)$$

with \mathbf{h}_d a given diffusive momentum flux at the Neumann boundary Γ_N .

Given the context of particle-mesh methods, it is important to elaborate upon the local and global mass conservation statement. Integrating the local mass conservation statement, Eq. (4.17), by parts yields for all cells $K \in \mathcal{T}$

$$\int_K \nabla \cdot \mathbf{u}_h \, q_h \, d\Omega = 0 \quad \forall q_h \in Q_h. \quad (4.23)$$

Since by virtue of the function space definitions $q_h \in Q_h$, and $\nabla \cdot \mathbf{u}_h \in Q_h$, it follows that the discrete velocity field \mathbf{u}_h is pointwise divergence-free within each cell for either one of the function space combinations, Eq. (4.9a), (4.9b) or (4.9c).

The global mass conservation equation, Eq. (4.18), can be written as (see also [8])

$$\sum_{F \in \mathcal{F}_I} \oint_F \llbracket \mathbf{u}_h \rrbracket \bar{q}_h \, d\Gamma + \sum_{F \in \mathcal{F}_B} \oint_F (\mathbf{u}_h - \bar{\mathbf{u}}_h) \cdot \mathbf{n} \bar{q}_h \, d\Gamma = 0 \quad \forall \bar{q}_h \in \bar{Q}_h, \quad (4.24)$$

in which \mathcal{F}_I the set of interior facets, \mathcal{F}_B the set of exterior facets, and the jump $\llbracket \mathbf{u}_h \rrbracket := \mathbf{u}_h^+ \cdot \mathbf{n}^+ + \mathbf{u}_h^- \cdot \mathbf{n}^-$. Employing the HDG or the EDG-HDG function space

combinations, Eq. (4.9a) or Eq. (4.9b), it can be seen that $\bar{q}_h \in \bar{Q}_h$ and $[[\mathbf{u}_h]] \in \bar{Q}_h$. Hence, it follows from Eq. (4.24) that for HDG and EDG-HDG:

$$[[\mathbf{u}_h]] = 0 \quad \forall \mathbf{x} \in F, \forall F \in \mathcal{F}_I, \quad (4.25)$$

$$(\mathbf{u}_h - \bar{\mathbf{u}}_h) \cdot \mathbf{n} = 0 \quad \forall \mathbf{x} \in F, \forall F \in \mathcal{F}_B, \quad (4.26)$$

which implies that the discrete velocity field \mathbf{u}_h has continuous normal component across cell facets. Together with the pointwise divergence-free property inside cells, this proves that the velocity field $\mathbf{u}_h \in \mathbf{W}_h$ is $H(\text{div})$ -conforming.

This $H(\text{div})$ conformity of the velocity field is not obtained for the EDG function space combination, Eq. (4.9c), for the obvious reason that $[[\mathbf{u}_h]] \notin (\bar{Q}_h \cap C^0(\Gamma_0))$. Hence, Eq. (4.24) only holds in a weak sense for the EDG formulation.

A last important difference with the fully Eulerian formulation from [6, 8] is the absence of the advective flux terms since advection of the material quantities is done at the particle level. Hence, the resulting set of equations at the mesh becomes linear.

4.2.4. Mesh-particle projection

Finally, the vector-valued counterpart of Eq. (3.18) is used for the mesh-particle projection $\mathcal{P}_L: \mathbf{W}_h \rightarrow \mathcal{V}_t$. Thus, the specific momentum \mathbf{v}_p associated with particle p is updated via

$$\mathbf{v}_p(t) = \mathbf{u}_h(\mathbf{x}_p(t), t) \quad \forall p \in \mathcal{S}_t. \quad (4.27)$$

This mesh-particle projection concludes the sequence of steps for a particle-mesh operator splitting of the incompressible Navier-Stokes equations.

4.2.5. Properties of the semi-discrete formulation

In this section, consistency and global and local momentum conservation of the semi-discrete particle-mesh operator splitting method for the incompressible Navier-Stokes equations are demonstrated.

Consistency

Consistency requires first of all that the semi-discrete formulation for the unsteady Stokes equations from Section 4.2.3 is consistent with the continuous problem in Eq. (4.3). This was proven in [6, 8, 97]. Secondly, consistency of the mutual projections implies the vector-valued counterpart of Eq. (3.20).

Consistency of the PDE-constrained projections is proven by assuming sufficiently smooth vector fields \mathbf{v} and \mathbf{U} . Upon substitution into Eq. (4.16b) and after integration by parts, this equation is restated as

$$\begin{aligned} \int_{\Omega} \left(\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \mathbf{U}) \right) \cdot \delta \boldsymbol{\lambda}_h \, d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} (\bar{\mathbf{v}} \otimes \bar{\mathbf{U}} - \mathbf{v} \otimes \mathbf{U}) \mathbf{n} \cdot \delta \boldsymbol{\lambda}_h \, d\Gamma \\ - \oint_{\Gamma_N} \gamma (\mathbf{v} \otimes \mathbf{U}) \mathbf{n} \cdot \delta \boldsymbol{\lambda}_h \, d\Gamma + \oint_{\Gamma_N} \gamma \mathbf{h}_a \cdot \delta \boldsymbol{\lambda}_h \, d\Gamma = 0 \quad \forall \delta \boldsymbol{\lambda}_h \in \mathbf{T}_h, \end{aligned} \quad (4.28)$$

which demonstrates consistency with the strong form of the advection problem, Eq. (4.2a-4.2c), with the enforcement of flux continuity $(\bar{\mathbf{v}} \otimes \bar{\mathbf{U}}) \mathbf{n} = (\mathbf{v} \otimes \mathbf{U}) \mathbf{n}$ on interior facets and the Dirichlet boundary on Γ_D .

Conservation

To obey discrete momentum conservation, it is required that the relevant steps in the particle-mesh splitting of the incompressible Navier-Stokes equations conserve momentum in a discrete sense. For a proof of global and local momentum conservation properties of the Stokes solver, reference is made to [6, 8, 97]. Employing the local ℓ^2 -projection to reconstruct specific momentum fields at the background mesh from the cloud of moving particles, compromises on discrete conservation properties. As for the scalar valued case, discrete global and local conservation properties on the mesh level are only obtained when carrying out the particle-mesh projection via the PDE-constrained approach. To prove this, a similar approach is adopted as for the scalar valued case, albeit the advective velocity field at the facets need specific attention.

Global conservation is demonstrated by setting $\delta \boldsymbol{\lambda}_h = \mathbf{e}_j$ in Eq. (4.16b), where \mathbf{e}_j a canonical unit vector in direction j . After rearranging, this yields

$$\int_{\Omega} \frac{\partial \mathbf{v}_h}{\partial t} \cdot \mathbf{e}_j \, d\Omega = - \sum_K \oint_{\partial K \setminus \Gamma_N} (\bar{\mathbf{v}}_h \otimes \bar{\mathbf{U}}_h) \mathbf{n} \cdot \mathbf{e}_j \, d\Gamma - \oint_{\Gamma_N} (1 - \gamma) (\mathbf{v}_h \otimes \mathbf{U}_h) \mathbf{n} \cdot \mathbf{e}_j \, d\Gamma - \oint_{\Gamma_N} \gamma \mathbf{h}_a \cdot \mathbf{e}_j \, d\Gamma. \quad (4.29)$$

As long as the flux on interior facets is at least weakly continuous, the boundary integral on the union of interior cell facets vanishes. With $\bar{\mathbf{v}}_h$ single-valued on facets, it follows that for global conservation the normal component of the advective velocity, i.e. $\bar{\mathbf{U}}_h \cdot \mathbf{n}$ should be at least weakly continuous across cell facets. This criterion is inherently met, provided that $\bar{\mathbf{U}}_h \in \bar{\mathbf{W}}_h$. Choosing $\bar{\mathbf{U}}_h \in \mathbf{W}_h$ would be equally valid as long as the velocity field $\bar{\mathbf{U}}_h$ possesses (weakly) continuous normal components across cell facets. On this premise, the right-hand side in Eq. (4.29) equals the net ingoing flux at the exterior boundary, which demonstrates global mass conservation.

On a similar note, a local conservation statement is obtained by setting $\delta \boldsymbol{\lambda}_h = \mathbf{e}_j$ on a cell K , and $\delta \boldsymbol{\lambda}_h = \mathbf{0}$ on $\mathcal{T} \setminus K$ in Eq. (4.16b):

$$\int_K \frac{\partial \mathbf{v}_h}{\partial t} \cdot \mathbf{e}_j \, d\Omega = - \oint_{\partial K \setminus \Gamma_N} (\bar{\mathbf{v}}_h \otimes \bar{\mathbf{U}}_h) \mathbf{n} \cdot \mathbf{e}_j \, d\Gamma - \oint_{\Gamma_N} (1 - \gamma) (\mathbf{v}_h \otimes \mathbf{U}_h) \mathbf{n} \cdot \mathbf{e}_j \, d\Gamma - \oint_{\Gamma_N} \gamma \mathbf{h}_a \cdot \mathbf{e}_j \, d\Gamma. \quad (4.30)$$

This shows that the change in momentum local to a cell is induced by the net ingoing flux on the cell boundaries, where for local conservation (weak) continuity of the advective flux on interior facets is again a prerequisite.

4.3. Fully-discrete formulations

This section presents fully-discrete formulations for the four steps comprising the particle-mesh operator splitting of the incompressible Navier-Stokes equations.

4.3.1. Particle advection

To advect the particle positions, the ODE

$$\dot{\mathbf{x}}_p(t) = \mathbf{U}_h(\mathbf{x}_p(t), t^n) \quad \forall p \in \mathcal{S}_t \quad (4.31)$$

has to be solved for every discrete time interval $I_n \in \mathcal{I}$. This first of all requires a judicious choice for the advective field $\mathbf{U}_h(\mathbf{x}, t)$, for which we use the mesh related velocity field $\mathbf{u}_h \in \mathbf{W}_h$ obtained in the Stokes step and frozen at the old time level t^n ,

$$\mathbf{U}_h(\mathbf{x}_p(t), t) = \mathbf{u}_h(\mathbf{x}_p(t), t^n) \quad t \in I_n. \quad (4.32)$$

Important to realize is that by doing so, the advective velocity field \mathbf{U}_h inherits the divergence properties of the field $\mathbf{u}_h \in \mathbf{W}_h$ obtained in the Stokes step. Hence, opting for the EDG function spaces, see Eq. (4.9), \mathbf{U}_h will be pointwise divergence-free within each cell K , but the normal component of the velocity is only weakly continuous across cell facets. If, instead, the HDG or EDG-HDG function spaces are employed, the $H(\text{div})$ -conformity of the computed velocity field $\mathbf{u}_h \in \mathbf{W}_h$ guarantees that the advective field \mathbf{U}_h is pointwise divergence-free, and has continuous normal components across facets [8].

To obtain a fully-discrete implementation of Eq. (4.31), explicit integration schemes can be used, since the advective velocity \mathbf{U}_h is explicitly known in each time interval $I_n \in \mathcal{I}$. To this end, a second-order two-step Adams-Bashforth scheme is employed. This choice is motivated by considering the time accuracy of the discretization of the Stokes equation on the mesh in conjunction with the projection steps, see Sections 4.3.2 - 4.3.4. The combination of these steps can be rendered second-order accurate in time, as will be argued in Section 4.3.5.

4.3.2. Particle-mesh projection

Local ℓ^2 projection

In a time-discrete setting, the local minimization problem of Eq. (4.14) reads as follows: given the particle velocities $\mathbf{v}_p^n \in \mathcal{V}_t$ and positions $\mathbf{x}_p^{n+1} \in \mathcal{X}_t$, find the velocity field $\mathbf{v}_h^{n+1} \in \mathbf{W}_h$ such that

$$\sum_K \sum_{\mathcal{S}_t^K} (\mathbf{v}_h^{n+1}(\mathbf{x}_p^{n+1}) - \mathbf{v}_p^n) \cdot \mathbf{w}_h(\mathbf{x}_p^{n+1}) = 0 \quad \forall \mathbf{w}_h \in \mathbf{W}_h. \quad (4.33)$$

This fully defines a discretization of the operator \mathcal{P}_E in Eq. (4.3f) by virtue of a local least squares projection.

PDE-constrained particle-mesh projection

Analogous to Eq. (3.27), an approximation of the time derivative

$$\left. \frac{\partial \mathbf{v}_h}{\partial t} \right|_{t^{n+\theta}} \approx \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^{*,n}}{\Delta t_n}, \quad (4.34)$$

with $\mathbf{v}_h^{*,n}$ defined as

$$\mathbf{v}_h^{*,n} := \mathbf{v}_h^n + \Delta t_n \left((1 - \theta_L) \dot{\mathbf{u}}_h^{n-1} + \theta_L \dot{\mathbf{u}}_h^n \right), \quad (4.35)$$

is used. Furthermore, the advective fields are given explicitly by the corresponding velocity fields \mathbf{u}_h and $\bar{\mathbf{u}}_h$ from the previous Stokes step, i.e. for $t \in I_n$ we make use of

$$\mathbf{U}_h(\mathbf{x}, t) := \mathbf{u}_h^n(\mathbf{x}), \quad \text{and} \quad \bar{\mathbf{U}}_h(\mathbf{x}, t) := \bar{\mathbf{u}}_h^n(\mathbf{x}), \quad (4.36)$$

which *de facto* linearizes the PDE-constrained particle-mesh projection problem.

With these definitions, a fully-discrete implementation of the optimality system for the constrained projection of the specific momentum reads: given the vector-valued particle field $\mathbf{v}_p^n \in \mathcal{V}_t$, the particle positions $\mathbf{x}_p^{n+1} \in \mathcal{X}_t$, the field $\mathbf{v}_h^{*,n} \in \mathbf{W}_h$, and the advective Neumann boundary condition \mathbf{h}_a , find $(\mathbf{v}_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}, \bar{\mathbf{v}}_h^{n+1}) \in (\mathbf{W}_h, \mathbf{T}_h, \bar{\mathbf{W}}_{h,g})$ such that

$$\begin{aligned} & \sum_{p \in \mathcal{S}_t} (\mathbf{v}_h^{n+1}(\mathbf{x}_p^{n+1}) - \mathbf{v}_p^n) \cdot \mathbf{w}_h(\mathbf{x}_p^{n+1}) - \sum_K \oint_{\partial K} \beta (\bar{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) \cdot \mathbf{w}_h \, d\Gamma + \int_{\Omega} \frac{\mathbf{w}_h}{\Delta t_n} \cdot \boldsymbol{\lambda}_h^{n+1} \, d\Omega \\ & - \theta \sum_K \int_K (\mathbf{w}_h \otimes \mathbf{u}_h^n) : \nabla \boldsymbol{\lambda}^{n+1} \, d\Omega + \theta \oint_{\Gamma_N} (1 - \gamma) (\mathbf{w}_h \otimes \mathbf{u}_h^n) \mathbf{n} \cdot \boldsymbol{\lambda}_h^{n+1} \, d\Gamma = 0 \quad \forall \mathbf{w}_h \in \mathbf{W}_h. \end{aligned} \quad (4.37a)$$

$$\begin{aligned} & \int_{\Omega} \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^{*,n}}{\Delta t_n} \cdot \boldsymbol{\tau}_h \, d\Omega - \theta \sum_K \int_K (\mathbf{v}_h^{n+1} \otimes \mathbf{u}_h^n) : \nabla \boldsymbol{\tau}_h \, d\Omega + \sum_K \oint_{\partial K \setminus \Gamma_N} (\bar{\mathbf{v}}_h^{n+1} \otimes \bar{\mathbf{u}}_h^n) \mathbf{n} \cdot \boldsymbol{\tau}_h \, d\Gamma \\ & + \theta \oint_{\Gamma_N} (1 - \gamma) (\mathbf{v}_h^{n+1} \otimes \mathbf{u}_h^n) \mathbf{n} \cdot \boldsymbol{\tau}_h \, d\Gamma + \oint_{\Gamma_N} \gamma \mathbf{h}_a^{n+\theta} \cdot \boldsymbol{\tau}_h \, d\Gamma \\ & = (1 - \theta) \sum_K \int_K (\mathbf{v}_h^{*,n} \otimes \mathbf{u}_h^n) : \nabla \boldsymbol{\tau}_h \, d\Omega - (1 - \theta) \oint_{\Gamma_N} (1 - \gamma) (\mathbf{v}_h^{*,n} \otimes \mathbf{u}_h^n) \mathbf{n} \cdot \boldsymbol{\tau}_h \, d\Gamma \quad \forall \boldsymbol{\tau}_h \in \mathbf{T}_h, \end{aligned} \quad (4.37b)$$

$$\sum_K \oint_{\partial K \setminus \Gamma_N} (\boldsymbol{\lambda}_h^{n+1} \otimes \mathbf{u}_h^n) \mathbf{n} \cdot \bar{\mathbf{w}}_h \, d\Gamma + \sum_K \oint_{\partial K} \beta (\bar{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) \cdot \bar{\mathbf{w}}_h \, d\Gamma = 0 \quad \forall \bar{\mathbf{w}}_h \in \bar{\mathbf{W}}_{h,0}. \quad (4.37c)$$

Solving Eqs. (4.37) for $(\mathbf{v}_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}, \bar{\mathbf{v}}_h^{n+1})$ gives the reconstructed field \mathbf{v}_h^{n+1} , serving as an input to the Stokes solver.

4.3.3. Stokes problem

The backward Euler method is used for the time discretization of the semi-discrete unsteady Stokes equations, so that the fully-discrete problem reads: given the intermediate field $\mathbf{u}_h^{*,n} = \mathbf{v}_h^{n+1} \in \mathbf{W}_h$, the diffusive Neumann boundary condition \mathbf{h}_d^{n+1} ,

the forcing term \mathbf{f}^{n+1} , and the kinematic viscosity ν , find $(\mathbf{u}_h^{n+1}, \bar{\mathbf{u}}_h^{n+1}, p_h^{n+1}, \bar{p}_h^{n+1}) \in (\mathbf{W}_h, \bar{\mathbf{W}}_{h,g}, Q_h, \bar{Q}_h)$ such that (local and global momentum balances),

$$\begin{aligned} & \int_{\Omega} \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^{*,n}}{\Delta t_n} \cdot \mathbf{w}_h \, d\Omega - \sum_K \int_K \boldsymbol{\sigma}_{d,h}^{n+1} : \nabla \mathbf{w}_h \, d\Omega + \sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h}^{n+1} \mathbf{n} \cdot \mathbf{w}_h \, d\Gamma \\ & + \sum_K \oint_{\partial K} 2\nu (\bar{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^{n+1}) \cdot \nabla^s \mathbf{w}_h \mathbf{n} \, d\Gamma = \int_{\Omega} \mathbf{f}^{n+1} \cdot \mathbf{w}_h \, d\Omega \quad \forall \mathbf{w}_h \in \mathbf{W}_h, \end{aligned} \quad (4.38a)$$

$$\sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h}^{n+1} \mathbf{n} \cdot \bar{\mathbf{w}}_h \, d\Gamma = \oint_{\Gamma_N} \mathbf{h}_d^{n+1} \cdot \bar{\mathbf{w}}_h \, d\Gamma \quad \forall \bar{\mathbf{w}}_h \in \bar{\mathbf{W}}_{h,0}, \quad (4.38b)$$

and (local and global mass balances),

$$\sum_K \int_K \mathbf{u}_h^{n+1} \cdot \nabla q_h \, d\Omega - \sum_K \oint_{\partial K} \mathbf{u}_h^{n+1} \cdot \mathbf{n} q_h \, d\Gamma = 0 \quad \forall q_h \in Q_h, \quad (4.38c)$$

$$\sum_K \oint_{\partial K} \mathbf{u}_h^{n+1} \cdot \mathbf{n} \bar{q}_h \, d\Gamma - \oint_{\partial\Omega} \bar{\mathbf{u}}_h^{n+1} \cdot \mathbf{n} \bar{q}_h \, d\Gamma = 0 \quad \forall \bar{q}_h \in \bar{Q}_h, \quad (4.38d)$$

are satisfied. The diffusive fluxes $\boldsymbol{\sigma}_{d,h}^{n+1}$ and $\hat{\boldsymbol{\sigma}}_{d,h}^{n+1}$ in these equations are given by

$$\boldsymbol{\sigma}_{d,h}^{n+1} = p_h^{n+1} \mathbf{I} - 2\nu \nabla^s \mathbf{u}_h^{n+1}, \quad (4.39)$$

$$\hat{\boldsymbol{\sigma}}_{d,h}^{n+1} = \bar{p}_h^{n+1} \mathbf{I} - 2\nu \nabla^s \mathbf{u}_h^{n+1} - 2\nu \frac{\alpha}{h_K} (\bar{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^{n+1}) \otimes \mathbf{n}. \quad (4.40)$$

Solving the unsteady Stokes system, Eq. (4.38), yields a specific momentum field $\mathbf{u}_h^{n+1} \in \mathbf{W}_h$ which is used in the subsequent mesh-particle projection to update the specific momentum carried by the particles. In addition, this field serves as the advective velocity field in the time interval I_{n+1} in the next time step.

4.3.4. Mesh-particle projection

For the incompressible Navier-Stokes equations, the mesh-particle projection is the vector-valued counterpart of Eq. (3.33), i.e. the momentum field on particles is updated according to

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t_n \left((1 - \theta_L) \dot{\mathbf{u}}_h^n(\mathbf{x}_p^n) + \theta_L \dot{\mathbf{u}}_h^{n+1}(\mathbf{x}_p^{n+1}) \right) \quad \forall p \in \mathcal{S}_t, \quad (4.41)$$

with $\dot{\mathbf{u}}_h^n(\mathbf{x}_p^n)$ and $\dot{\mathbf{u}}_h^{n+1}(\mathbf{x}_p^{n+1})$ denoting the mesh-based accelerations at the respective time levels n and $n+1$, evaluated at the individual particle positions.

4.3.5. Algorithmic aspects

Time stepping accuracy

In essence, the same time stepping strategy is adopted as for the scalar-valued linear advection-diffusion equations. Hence, the fully-discrete time-stepping scheme provides a

second-order accurate approximation to the problem from Eqs. (4.2, 4.3) for $\theta_L = 1/2$, and $l = 0$ in case the PDE-constrained particle-mesh projection is used. However, two remarks are to be made:

Remark 1: Formal second-order accuracy is retained when using the local ℓ^2 particle-mesh projection as long as a second-order accurate scheme for the particle advection is used, e.g. the two-step Adams-Bashforth scheme.

Remark 2: For the PDE-constrained particle-mesh projection the advective velocity is set to $\mathbf{U}_h(\mathbf{x}, t) = \mathbf{u}_h^n$ in the time interval $t \in I_n$. This in fact linearizes the non-linear advection term in the incompressible Navier-Stokes equations by means of a Picard linearization. Hence, this choice might compromise on formal second-order time accuracy.

Static condensation

As demonstrated for the scalar-valued case, the PDE-constrained particle-mesh projection can be implemented efficiently using *static condensation*, see Section 3.6.3. This implementation strategy can also be applied to implement the HDG Stokes equations. To this end, the following block-structure at element level is recognized in Eq. (4.38):

$$\begin{bmatrix} \mathcal{A} & \mathcal{G} \\ \mathcal{G}^\top & \mathcal{B} \end{bmatrix} \begin{bmatrix} \mathcal{U} \\ \bar{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} \mathcal{Q} \\ \mathcal{S} \end{bmatrix} \quad (4.42)$$

in which $\mathcal{U} = [\mathbf{u}, \mathbf{p}]^\top$ a vector of the local velocity and pressure dofs, and $\bar{\mathcal{U}} = [\bar{\mathbf{u}}, \bar{\mathbf{p}}]^\top$ a vector containing the facet velocity and pressure dofs. The block-matrix \mathcal{A} results from the discretization of the terms involving the local velocity and pressure variables, \mathbf{u}_h^{n+1} and p_h^{n+1} , and \mathcal{G} results from the discretization of the terms involving the interface variables $\bar{\mathbf{u}}_h^{n+1}$ and \bar{p}_h^{n+1} in the local momentum and mass balance, Eq. (4.38a) and Eq. (4.38c). The vector \mathcal{Q} contains the discrete body force, and the terms involving the explicitly known intermediate solution $\mathbf{u}_h^{*,n}$. Furthermore, \mathcal{B} is a block matrix involving the interface variable terms $\bar{\mathbf{u}}_h^{n+1}$ and \bar{p}_h^{n+1} in the global momentum and mass balance, Eq. (4.38b) and Eq. (4.38d), and \mathcal{S} is vector containing the discretization of the inhomogeneous Neumann boundary.

Expressing the local variables in terms of the global variables in the system Eq. (4.42) yields:

$$\mathcal{U} = \mathcal{A}^{-1} (\mathcal{Q} - \mathcal{G}\bar{\mathcal{U}}) \quad (4.43)$$

Substituting this expression in the equations for the global variables collected in $\bar{\mathcal{U}}$, leads to

$$(\mathcal{G}^\top \mathcal{A}^{-1} \mathcal{G} - \mathcal{B}) \bar{\mathcal{U}} = \mathcal{G}^\top \mathcal{A}^{-1} \mathcal{Q} - \mathcal{S} \quad (4.44)$$

Assembling the local contributions Eq. (4.44) results in the global matrix

$$\bigwedge_K (\mathcal{G}^\top \mathcal{A}^{-1} \mathcal{G} - \mathcal{B}) \bar{\mathcal{U}} = \bigwedge_K (\mathcal{G}^\top \mathcal{A}^{-1} \mathcal{Q} - \mathcal{S}), \quad (4.45)$$

where \bigwedge_K implies assembly of the element contributions. After solving this system for the global solution $\bar{\mathcal{U}}$, the local solution \mathcal{U} is obtained after a straightforward backsubstitution via Eq. (4.43).

4.4. Numerical examples

A range of numerical examples illustrates the particle-mesh operator splitting for the incompressible Navier-Stokes equations. Particular attention is paid to the accuracy and conservation properties as well as the necessary conditions to maintain a uniform particle distribution throughout the computations.

All numerical examples consider domains $\Omega \in \mathbb{R}^2$, and constant time step sizes Δt are used to partition the time interval. Furthermore, the regularization term β in the PDE-constrained projection is set to a fixed value of 10^{-6} in all computations, and, unless otherwise specified, $l = 0$ so that the PDE-constrained particle-mesh projection is independent of θ . Moreover, the penalty parameter α in the definition of the diffusive interface flux in Eq. (4.40) is set to $6k^2$ as in [6, 8].

Unless otherwise specified, the $H(\text{div})$ conforming HDG variant, Eq. (4.9a), is used for the discretization of the unsteady Stokes equations. This choice is historically determined, since the bulk of the work presented in this chapter was conducted before the more efficient EDG-HDG variant was published. As for the advection-diffusion equation, the scheme is implemented using tools from the FEniCS project, see Appendix D for implementation details.

4.4.1. Poiseuille flow

Plane Poiseuille flow is considered to assess the spatial convergence behavior of the particle-mesh operator splitting scheme for the incompressible Navier-Stokes equations in the presence of no-slip boundaries and to elaborate upon the required minimum number of particles per cell.

Starting from rest, the flow gradually develops towards a steady state under the influence of a constant axial body force $\mathbf{f} = F\mathbf{e}_x$. The channel is modeled in the x - y plane, where the x -axis points in the direction of the flow and the plates are located at $y = \pm d$. The flow is periodic in the x -direction and at the location of the plates ($y = \pm 0.25$) no-flux and no-slip boundary conditions are used in the PDE-constrained particle-mesh projection and the Stokes step, respectively.

The transient solution of the axial velocity $u_x(y, t)$ is given by [98]

$$u_x(y, t) = \frac{F}{2\nu}(d^2 - y^2) - \sum_{n=0}^{\infty} \frac{(-1)^n 16d^2 F}{\nu\pi^3(2n+1)^3} \cos\left(\frac{(2n+1)\pi y}{2d}\right) \exp\left(-\frac{(2n+1)^2\pi^2\nu t}{4d^2}\right), \quad (4.46)$$

which for $t \rightarrow \infty$ tends to the stationary solution

$$u_x(y) = \frac{F}{2\nu}(d^2 - y^2) = U \left(1 - \frac{y^2}{d^2}\right), \quad (4.47)$$

where $U = \frac{1}{2}F d^2/\nu$ is the steady-state centerline velocity. The kinematic viscosity is set to $\nu = 1 \cdot 10^{-3}$, and the body force F is chosen such that the steady state Reynolds number $Re = 2Ud/\nu$ equals 200. The model domain is given by $\Omega := [0, 1] \times [-0.25, 0.25]$, furthermore, the CFL-number is defined as:

$$\text{CFL} = \frac{U\Delta t}{h_K}. \quad (4.48)$$

Convergence study

The domain of interest Ω is triangulated using a series of refined meshes, and on average, 30 particles are initially assigned per cell. This number is kept constant upon mesh refinement. Different test cases are considered, by varying the polynomial orders ($k = 1, 2$) and the particle-projection method, see Table 4.1. Specifically, Case 5 illustrates the impediments when introducing an inconsistency between the particle-mesh projection and the mesh-particle projection by using $\theta_L = 1/2$ and $\theta_L = 1$ in Eq. (4.41) and (4.35), respectively.

To illustrate the evolution of the transient solution, the discrete solutions of the velocity profile at various dimensionless time instants $t^* = tU/2d$ using a mesh of $40 \times 20 \times 2$ cells and a corresponding time step size $\Delta t = 0.125$ (CFL ≈ 1) are shown in Fig. 4.1 for Case 1 (piecewise-linear basis functions) and Case 2 (piecewise-quadratic basis functions). Both for the linear - and the quadratic basis functions the results match closely with the analytical solution. To further quantify and analyse the results, velocity

Table 4.1: Poiseuille flow: Overview of model settings. HDG function spaces are used.

	Projection Method	k	l	θ_L	Remark
Case 1	ℓ^2	1	-	1/2	-
Case 2	ℓ^2	2	-	1/2	-
Case 3	PDE	1	0	1/2	-
Case 4	PDE	2	0	1/2	-
Case 5	PDE	2	0	1/2	$\theta_L = 1/2$ in Eq. (4.41)
				1	$\theta_L = 1$ in Eq. (4.35)

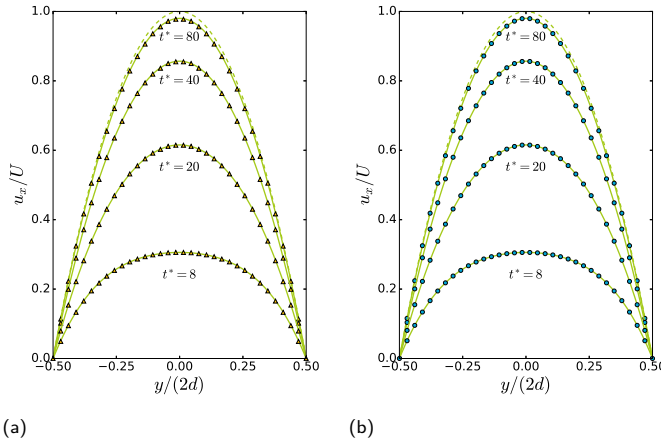


Figure 4.1: Poiseuille flow: non-dimensional flow velocity profiles at various dimensionless time instants $t^* = tU/2d$ for $Re = 200$ using polynomial orders (a) $k = 1$ (Δ) and (b) $k = 2$ (\circ); also shown are the analytical solutions (solid) including the steady-state solution for $t \rightarrow \infty$ (dashed).

and pressure errors at the dimensionless time instant $t^* = tU/2d = 100$ are presented

in Table 4.2 for the different test cases. The observed convergence is near-optimal for Cases 1-4, i.e. of order $k+1$ in the velocity and of order k in the pressure. This indicates that for these cases, the spatial convergence rate is not affected by the particle-mesh operator splitting. For Case 5, the convergence drops to first-order in the velocity as a result of the inaccuracy introduced in the definition of the stored mesh field $\mathbf{v}_h^{*,n}$ in Eq. (4.35).

Table 4.2: Poiseuille flow: convergence of the L^2 -error in the velocity and the pressure at dimensionless time $t^* = tU/2d = 100$ for different polynomial orders (k, l) .

	Cells	Δt	$\ \mathbf{u} - \mathbf{u}_h\ $	Rate	$\ p - p_h\ $	Rate
Case 1 ($k = 1$)	64	0.2	6.1e-3	-	1.5e-4	-
	256	0.1	1.6e-3	2.0	8.2e-5	0.9
	1024	0.05	3.9e-4	2.0	3.6e-5	1.2
	4096	0.025	9.7e-5	2.0	1.8e-5	1.0
Case 2 ($k = 2$)	64	0.2	3.9e-6	-	1.3e-7	-
	256	0.1	4.3e-7	3.2	1.8e-8	2.8
	1024	0.05	5.2e-8	3.1	3.7e-9	2.3
	4096	0.025	5.1e-9	3.3	8.0e-10	2.2
Case 3 ($k = 1$)	64	0.2	6.2e-3	-	1.4e-4	-
	256	0.1	1.6e-3	2.0	8.1e-5	0.7
	1024	0.05	3.9e-4	2.0	3.7e-5	1.2
	4096	0.025	9.7e-5	2.0	1.8e-5	1.0
Case 4 ($k = 2$)	64	0.2	3.9e-6	-	1.3e-7	-
	256	0.1	4.3e-7	3.2	1.8e-8	2.8
	1024	0.05	5.1e-8	3.1	3.7e-9	2.3
	4096	0.025	5.1e-9	3.3	8.0e-10	2.2
Case 5 ($k = 2$)	64	0.2	2.6e-5	-	1.4e-7	-
	256	0.1	1.4e-5	1.0	1.9e-8	2.9
	1024	0.05	7.2e-6	1.0	7.6e-9	1.4
	4096	0.025	3.6e-6	1.0	1.3e-9	2.6

Particle resolution

Evidently, it is desirable for efficiency to keep the number of particles as low as possible without compromising accuracy. We therefore investigate the influence of the particle resolution on the accuracy by using $l = 0$ and $k = 1, 2, 3, 4$, combined with a variable particle resolution. We restrict the discussion to the mesh containing 256 cells and a time step of $\Delta t = 0.1$ ($\text{CFL} \approx 0.5$) and vary the particle resolution for this configuration, so that the number of particles per cell (denoted with \bar{S}_0^K) is in the range 2-50, initially. In order to have full control over the initial particle configuration, particles are placed on a regular lattice.

In Fig. 4.2 results are visualized by plotting the L^2 -error of the velocity field against the average number of particles per cell for the different polynomial orders. Many of the low particle resolution tests fail prematurely before reaching the end time $t^* = tU/2d = 100$, and for plotting purposes an error-value of 1 is assigned in such cases.

Evident from the figure are the sharp transitions in the error levels once a critical value of \bar{S}_0^K is exceeded. That is, for low particle resolutions the computations are prone to fail

prematurely, whereas from a certain particle resolution threshold onward, accurate results are obtained with error values being independent of the particle resolution. Interestingly, this threshold particle resolution depends on the polynomial order of the basis functions and is approximately equal to $\bar{S}_0^K = 8, 8, 10, 15$ for $k = 1, 2, 3, 4$. By recognizing that the PDE-constrained particle-mesh projection roots in the local least-squares minimization problem (Eq. (4.13)), we note that unisolvency of the particle locations with respect to the discontinuous function space \mathbf{W}_h is a necessary condition for the particle-mesh projection to be accurate. Empirically, we thus observe this condition to be satisfied if the minimal number of particles in a cell is at least equal to the number of local basis functions, except for linear basis functions ($k = 1$) and to a lesser extent for quadratic basis functions ($k = 2$), which require a higher number of particles per cell, since in these cases the numerical results are more sensitive to fluctuations in the number of particles per cell.

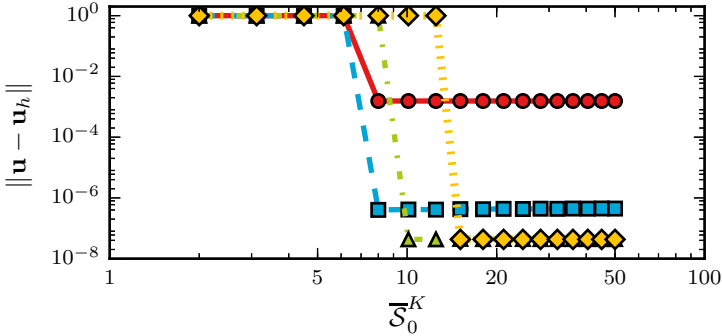


Figure 4.2: Poiseuille flow: L^2 -error at $t^* = 100$ as a function of the average number of particles initially assigned per cell (\bar{S}_0^K) for an Eulerian mesh with a fixed number of 256 cells and a time step of $\Delta t = 0.1$ for a linear (●), quadratic (■), cubic (▲) and quartic (◆) polynomial basis.

4.4.2. Taylor-Green flow

The Taylor-Green problem features a periodic sequence of decaying vortices. Since only an initial condition needs to be imposed, this test is particularly suited to determine the accuracy of the time integration method. The absence of body forces and boundary conditions render this test also useful for analyzing the momentum error. First and foremost, however, the Taylor-Green problem is used to assess the quality of the particle advection algorithm. In particular, the presence of stagnation points can lead to a poor particle distribution, as observed in other Lagrangian particle methods (see e.g. [42, 48]). Maintaining a high-quality particle distribution is imperative since the occurrence of clusters or voids would compromise the resolution of the specific momentum provided by the particles, leading to model inaccuracy or failure as illustrated in Fig. 4.2.

The test employs a bi-periodic square domain $\Omega = [-1, 1] \times [-1, 1]$, so that the side lengths are $L_x = L_y = 2$. Closed analytical expressions for the velocity and the pressure

are given by, respectively,

$$\mathbf{u}(\mathbf{x}, t) = U \exp(-2\nu\pi^2 t) (-\cos(k_x x) \sin(k_y y), \sin(k_x x) \cos(k_y y))^\top, \quad (4.49)$$

$$p(\mathbf{x}, t) = \frac{1}{4} \exp(-4\nu\pi^2 t) (\cos(2k_x x) + \cos(2k_y y)), \quad (4.50)$$

in which $U = 1$ is the initial velocity amplitude, ν is the kinematic viscosity, and $k_x = 2\pi/L_x$ and $k_y = 2\pi/L_y$ are wave numbers in the x - and y -direction, respectively. The Reynolds number is defined as $Re = UL/\nu$ with ν , and the CFL-number is defined as in Eq. (4.48), with U the initial velocity amplitude from Eq. (4.49).

Particle distribution

To assess the quality of the computed particle distribution, eight different function space combinations are considered for the Taylor-Green test, see Table 4.3. These combinations cover a number of HDG implementations, and for comparison two types of Taylor-Hood elements are considered as well. The ℓ^2 projections are employed for the particle-mesh projection, since this approach can be straightforwardly applied to the Taylor-Hood discretization. Furthermore, a coarse resolution mesh consisting of $8 \times 8 \times 2$ cells is employed, and initially 10000 particles are randomly seeded in the domain. The time step is set to $\Delta t = 8e-2$ ($CFL \approx 0.25$) and the end time of the simulations T_{end} is set to 10, unless the model run fails prematurely due to a lack of particles in which case results are given for the last time step before failure. Furthermore, the kinematic viscosity is set to $\nu = 2 \cdot 10^{-3}$ so that the Reynolds number is 1000. As yet, the only concern is the quality of the particle distribution. For an incompressible flow, this implies that an initially uniform particle configuration should remain uniform over the course of a simulation.

Table 4.3: Taylor-Green flow: overview of element types for assessing the particle distribution uniformity.

Name	Function Space	k	$k - 1$
EDG 1-0	Eq. (4.9c)	1	0
EDG 2-1		2	1
HDG 1-0	Eq. (4.9a)	1	0
HDG 2-1		2	1
EDG-HDG 1-0	Eq. (4.9b)	1	0
EDG-HDG 2-1		2	1
TH 2-1	Taylor-Hood quadratic velocity, linear pressure		
TH 3-2	Taylor-Hood cubic velocity, quadratic pressure		

Various metrics for investigating the particle distribution are available in literature, see e.g. the metrics proposed in [46, 99] and used in earlier work [95]. These metrics are based on the number of *particles per cell*, and therefore agnostic to the precise spatial distribution of the particles. To provide insight into the latter, the so-called radial distribution function (RDF) is used. The RDF quantitatively characterizes the particle distribution by describing how the number density of particles changes as a function of

distance from a reference particle, and is extensively used in statistical mechanics, e.g. [100], and for assessing the particle distribution in particle-based methods, e.g. [101].

For a uniform particle distribution, in which the particles are infinitesimal points rather than having a finite shape, the RDF $g(r)$ (see, e.g. [102], Eq. (3.21) for a 2 dimensional formulation) should evaluate to unity independent of the radial distance to the reference particle. This condition is accurately met using the HDG and the EDG-HDG function spaces, irrespective of the polynomial order, see Fig. 4.3b and Fig. 4.3c. For the EDG element, the RDF is close to unity for the EDG 2-1 element, whereas this is not so for the lower-order EDG 1-0 element where the RDF is much larger than unity for small radii r , see Fig. 4.3a. This implies that for the EDG 1-0 case, there is an increased likelihood of finding a particle close to the reference particle compared to finding a particle more distantly, thus indicating that particles tend to clump together. Similarly, the RDF is significantly larger than unity for the Taylor-Hood element, irrespective of the polynomial orders used, albeit TH 3-2 case appears to improve upon the TH 2-1 case.

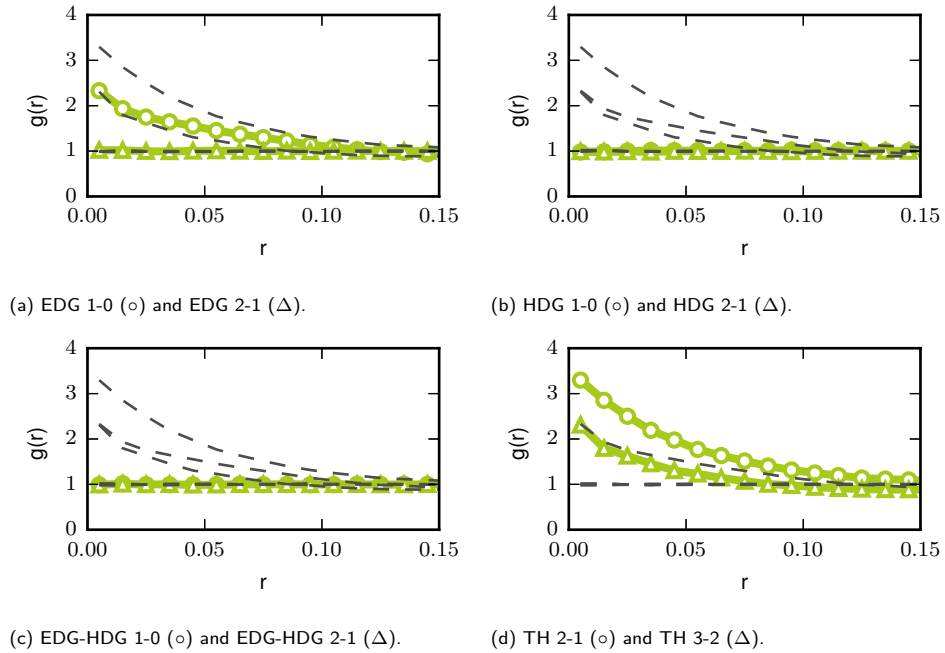


Figure 4.3: Taylor-Green flow: radial distribution function as a function of $r = \|\mathbf{x} - \mathbf{x}_p\|$ for different element types at $t = T_{\text{end}}$. Dashed grey lines indicate RD's from other function space combinations.

Before investigating the cause of these large differences in the RDF for the various element types, the practical implications on the particle distribution are visualized in Fig. 4.3. A non-uniform particle distribution is observed for the Taylor-Hood cases, especially so for the TH 2-1 case, which even fails prematurely at $T_{\text{end}} = 5.44$, see Fig. 4.4g. Also, the EDG 1-0 fails prematurely at $T_{\text{end}} = 1.2$ due to empty cells (i.e.

cells not containing particles), Fig. 4.4a. An improvement in the particle distribution is observed by increasing the polynomial orders to 2-1 for the EDG element, Fig. 4.4b.

To shed light on the underlying reason causing the difference in particle distributions for the distinct elements, the local divergence error

$$\epsilon_{\nabla} := \left(\sum_K \int_K (\nabla \cdot \mathbf{u}_h)^2 d\Omega \right)^{1/2}. \quad (4.51)$$

and the local normal velocity jump error over the internal facets \mathcal{F}_I

$$\epsilon_{[[\mathbf{u}]]} := \left(\sum_{F \in \mathcal{F}_I} \int_F [[\mathbf{u}]]^2 d\Gamma \right)^{1/2}, \quad (4.52)$$

with the jump operator $[[\mathbf{u}]] = \mathbf{u}^+ \cdot \mathbf{n}^+ + \mathbf{u}^- \cdot \mathbf{n}^-$, are further investigated at $t = T_{\text{end}}$.

Table 4.4 reports the results for these two error norms. The EDG, HDG and EDG-HDG approaches are all locally divergence free, with errors ϵ_{Δ} being of machine precision. More precisely, the EDG method is pointwise divergence free within a cell, but lacks continuity of the normal component of the velocity over a facet. The latter is apparent from the non-zero jump error $\epsilon_{[[\mathbf{u}]]}$. The sharp decrease in the jump error for the EDG 2-1 case compared to the EDG 1-0 case is remarkable. Using a quadratic basis for the facet pressures ($k = 2$) for the EDG 2-1 case, the pressure dofs placed at the facet midpoints offer improved control over the facet-normal velocity component compared to the case $k = 1$, leading to an improved inter-element continuity of the advective velocity field. This also explains the drastic improvement in maintaining a uniform particle distribution, see Fig. 4.4f.

With the velocity fields being continuous, inter-element continuity is inherently satisfied for the Taylor-Hood elements. The non-uniformity in the particle distribution for the TH 2-1 (Fig. 4.4g) and the TH 3-2 (Fig. 4.4h) case is therefore attributed to the local divergence-error in the Taylor-Hood method, see Table 4.4.

Finally, the HDG and EDG-HDG method are pointwise divergence-free within cells. In addition, the normal component of the velocity \mathbf{u}_h is continuous across cell facets, which renders these methods $H(\text{div})$ -conforming as discussed in Section 4.2.3. These properties are confirmed by the results in Table 4.4, showing that ϵ_{∇} and $\epsilon_{[[\mathbf{u}]]}$ are both zero to machine precision for the HDG and EDG-HDG case.

Contrasting these features of the HDG and the EDG-HDG method with the EDG and TH method, it is imperative to satisfy two criteria for maintaining a uniform particle distribution in incompressible flows. Firstly, the velocity field by which the particles are advected has to be locally (or pointwise) divergence-free, a condition which is met by the EDG and HDG methods, but not for the Taylor-Hood element. Secondly, the discrete velocity field should have sufficient inter-element continuity. This can be achieved by using a higher-order polynomial basis in the EDG method, but is inherently satisfied by using the $H(\text{div})$ -conforming HDG and EDG-HDG discretizations. By satisfying these two criteria, a uniform particle distribution is maintained over time, even near stagnation points, and no additional particle shifting techniques or velocity field reconstruction techniques are required, as for instance in [41–43, 48].

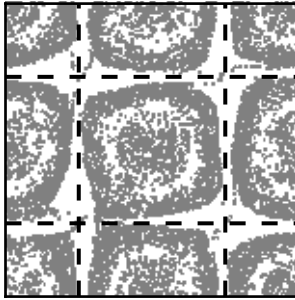
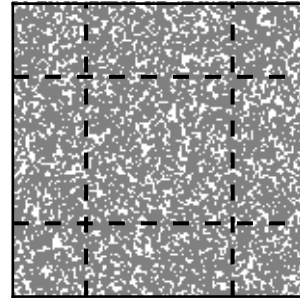
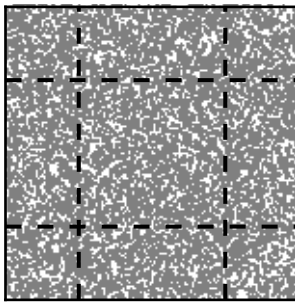
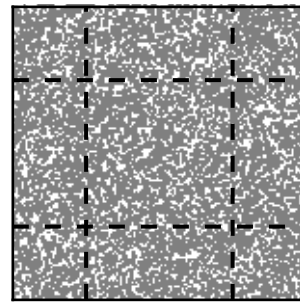
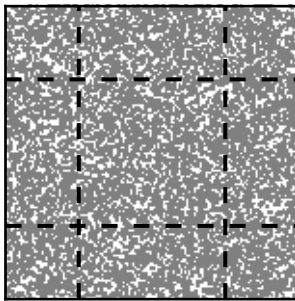
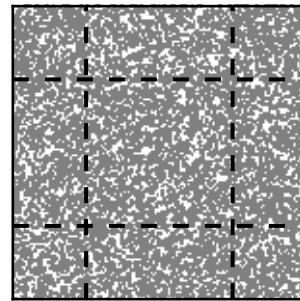
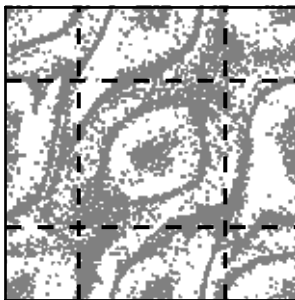
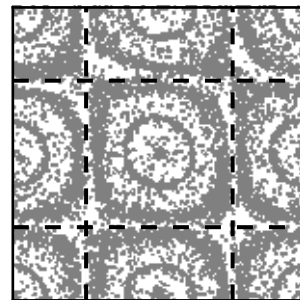
(a) EDG 1-0, $T_{\text{end}} = 1.2$ (b) EDG 2-1, $T_{\text{end}} = 10$ (c) HDG 1-0, $T_{\text{end}} = 10$ (d) HDG 2-1, $T_{\text{end}} = 10$ (e) EDG-HDG 1-0, $T_{\text{end}} = 10$ (f) EDG-HDG 2-1, $T_{\text{end}} = 10$ (g) TH 2-1, $T_{\text{end}} = 5.44$ (h) TH 3-2, $T_{\text{end}} = 10$

Figure 4.4: Taylor-Green flow: particle distributions on the biperiodic domain $\Omega := [-1, 1]^2$ at $t = T_{\text{end}}$ for different combinations of element type and polynomial orders, see Table 4.4. The intersections of the dashed lines indicate the locations of the stagnation points.

Table 4.4: Taylor-Green flow: local divergence error ϵ_{∇} and local jump error $\epsilon_{\llbracket \mathbf{u} \rrbracket}$ at $t = T_{\text{end}}$ for the different element configurations from Table 4.3.

	T_{end}	ϵ_{∇}	$\epsilon_{\llbracket \mathbf{u} \rrbracket}$
EDG 1-0	1.2	2.1e-15	1.9
EDG 2-1	10	3.0e-14	1.7e-2
HDG 1-0	10	2.4e-15	7.9e-16
HDG 2-1	10	3.3e-14	1.0e-15
EDG-HDG 1-0	10	2.3e-15	7.6e-16
EDG-HDG 2-1	10	3.3e-14	9.1e-16
TH 2-1	5.44	2.3	1.0e-16
TH 3-2	10	0.5	7.4e-16

Convergence study

To investigate the convergence properties of the scheme for constant CFL number, consider a time interval of interest $I = (0, 2]$. The initial peak velocity U in Eq. (4.49) is set to 1, the wave lengths $L_x = L_y = 2$, and the kinematic viscosity is either $\nu = 2 \cdot 10^{-2}$ or $\nu = 2 \cdot 10^{-3}$, with corresponding Reynolds numbers of $Re = UL/\nu = 100$ and $Re = 1000$. The chosen time step corresponds to a CFL-number of approximately 0.4. Different polynomial orders are considered for the HDG function space as defined by Eq. (4.9a), see Table 4.5. The PDE-constrained particle-mesh projection is employed in Cases 1,2, and for comparison, Case 3 and Case 4 make use of the non-conservative ℓ^2 particle-mesh projection.

Table 4.5: Taylor-Green flow: Overview of model settings. HDG function spaces are used.

	Projection Method	k	l	θ_L
Case 1	PDE	1	0	1/2
Case 2	PDE	2	0	1/2
Case 3	ℓ^2	2	-	1/2
Case 4	ℓ^2	2	-	1

In addition to velocity and pressure errors for the different model runs, Table 4.6 reports the global momentum conservation error ϵ_m , with this error being defined as

$$\epsilon_m = \left| \int_{\Omega} (\mathbf{v}_h(\mathbf{x}, t) - \mathbf{v}_h(\mathbf{x}, 0)) \cdot \mathbf{e}_x \, d\Omega \right| + \left| \int_{\Omega} (\mathbf{v}_h(\mathbf{x}, t) - \mathbf{v}_h(\mathbf{x}, 0)) \cdot \mathbf{e}_y \, d\Omega \right|, \quad (4.53)$$

in which \mathbf{e}_x and \mathbf{e}_y the unit vectors in the Cartesian x and y -direction, respectively. In all the runs, the average number of particles per cell is 28. For Case 1, consistent second-order convergence is observed for the velocity and first-order convergence is obtained for the pressure. Given the function spaces (piecewise linear velocity, and piecewise constant pressure), no better convergence rates could be expected.

For Case 2, approximately second-order convergence in the velocity and the pressure is obtained for the $Re = 100$ test case and the $Re = 1000$ test case. This indicates that

for $k = 2$ the time stepping error becomes dominant over the spatial error, where the former is expected to converge with second-order. Momentum is conserved globally up to machine precision for the PDE-constrained particle-mesh interaction (Case 1, Case 2), whereas this is clearly not so for the unconstrained projection method (Case 3), using local ℓ^2 -projections. Noteworthy to mention is that the errors in the velocity and pressure are almost identical for Cases 2 and 3.

Table 4.6: Taylor-Green flow: overview of model runs with the associated errors $\|\mathbf{u} - \mathbf{u}_h\|$, $\|p - p_h\|$ and ϵ_m at time $t = 2$.

		Cells	Δt	$\ \mathbf{u} - \mathbf{u}_h\ $	Rate	$\ p - p_h\ $	Rate	ϵ_m
Case 1 ($k = 1$)	$Re = 100$	128	0.1	1.2e-1	-	6.5e-2	-	1.7e-15
		512	0.05	2.4e-2	2.3	2.6e-2	1.3	8.9e-15
		2048	0.025	4.7e-3	2.4	1.2e-2	1.1	5.4e-15
		8192	0.0125	1.5e-3	1.6	6.0e-3	1.0	4.7e-15
	$Re = 1000$	128	0.1	2.4e-1	-	2.9e-1	-	2.3e-15
		512	0.05	4.8e-2	2.3	8.7e-2	1.7	1.8e-14
		2048	0.025	1.1e-2	2.1	4.0e-2	1.1	8.2e-15
		8192	0.0125	4.0e-3	1.5	2.0e-2	1.0	2.3e-14
Case 2 ($k = 2$)	$Re = 100$	128	0.1	6.5e-3	-	1.5e-2	-	8.8e-14
		512	0.05	1.9e-3	1.8	3.2e-3	2.2	1.6e-13
		2048	0.025	5.1e-4	1.9	8.4e-3	1.9	3.3e-13
		8192	0.0125	1.3e-4	2.0	2.1e-4	2.0	6.5e-13
	$Re = 1000$	128	0.1	7.6e-2	-	5.6e-2	-	1.4e-13
		512	0.05	1.2e-2	2.7	1.3e-2	2.1	3.1e-13
		2048	0.025	2.3e-3	2.4	3.2e-3	2.0	6.1e-13
		8192	0.0125	5.6e-4	2.0	7.8e-4	2.0	1.3e-12
Case 3 ($k = 2$)	$Re = 100$	128	0.1	6.6e-3	-	1.5e-2	-	3.0e-4
		512	0.05	1.9e-3	1.8	3.2e-3	2.2	1.2e-4
		2048	0.025	5.2e-4	1.9	8.5e-4	1.9	1.3e-5
		8192	0.0125	1.3e-4	2.0	2.1e-4	2.0	4.2e-7
	$Re = 1000$	128	0.1	7.6e-2	-	5.6e-2	-	3.6e-3
		512	0.05	1.2e-2	2.7	1.3e-2	2.1	6.4e-5
		2048	0.025	2.2e-3	2.4	3.1e-3	2.1	4.4e-6
		8192	0.0125	5.5e-4	2.0	7.6e-4	2.0	1.6e-6
Case 4 ($k = 2$)	$Re = 100$	128	0.1	9.5e-2	-	5.5e-2	-	1.0e-3
		512	0.05	5.2e-2	0.9	3.3e-2	0.7	6.4e-5
		2048	0.025	2.8e-2	0.9	1.8e-2	0.9	2.2e-5
		8192	0.0125	1.4e-2	1.0	9.6e-3	0.9	3.7e-7
	$Re = 1000$	128	0.1	7.6e-2	-	4.5e-1	-	8.6e-4
		512	0.05	1.2e-2	0.8	3.0e-1	0.6	9.4e-5
		2048	0.025	2.2e-3	0.8	1.8e-1	0.7	5.4e-6
		8192	0.0125	5.5e-4	0.9	1.0e-1	0.8	5.7e-7

High Reynolds number test

Finally, we show that the developed scheme is robust for high Reynolds numbers. To this end, we investigate an 8×8 lattice of Taylor-Green vortices (i.e. using $L_x = L_y = 0.5$). The initial velocity amplitude is $U = 1$ and the kinematic viscosity is chosen such that the Reynolds number is $Re = UL/\nu = 2.5 \cdot 10^5$. A $200 \times 200 \times 2$ regular triangular mesh is used, the polynomial orders are set to $k = 2$ and $l = 0$, and HDG function spaces are used for the discretization of the Stokes equations. Furthermore, $2.25 \cdot 10^6$ particles are seeded in the domain resulting in approximately 28 particles per cell, initially. Furthermore, the time step size is $\Delta t = 5 \cdot 10^{-3}$, resulting in a CFL-number of approximately 1.

The vorticity field at various time instants is plotted in Fig. 4.5. For this high Reynolds number, the regular initial lattice of vortices collapses into a chaotic, turbulent motion.

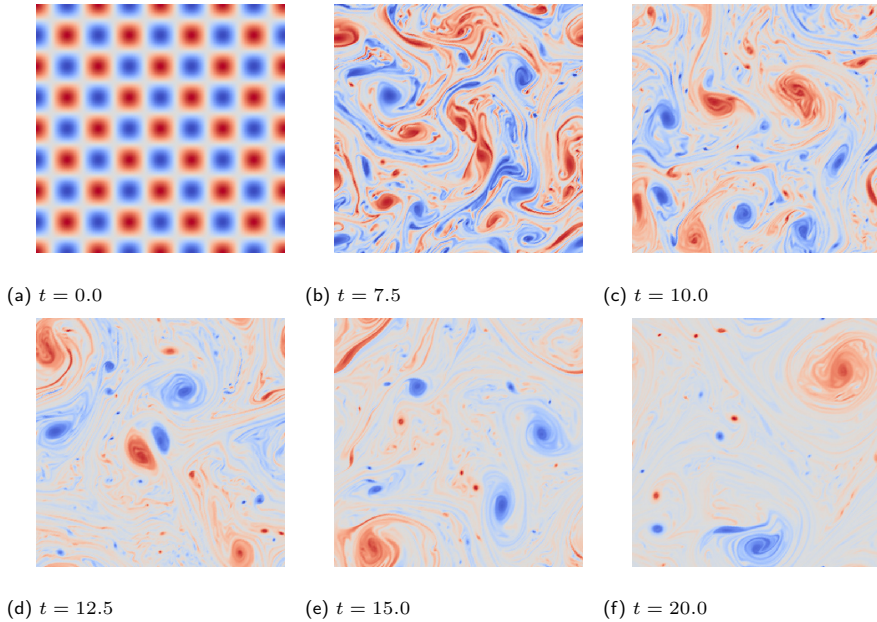


Figure 4.5: Taylor-Green flow: Vorticity fields for Taylor-Green flow with $Re = 2.5 \cdot 10^5$ at different time instances. Color scaling is between -25 (blue) and 25 (red).

Two features characteristic of 2D turbulence are qualitatively observed in Fig. 4.5. The emergence of large-scale coherent structures is clearly observed for longer simulation runtimes, compare for example the vorticity fields at $t = 7.5$ and $t = 20$. These large-scale vortices are only weakly dissipative and persist for many eddy-turnover times [103]. Secondly, the viscous dissipation takes place in the thin vortex filaments, clearly observed in Fig. 4.5. Somewhat remarkable are the small-scale, high-vorticity spots, best observed for the longer simulation runtimes (e.g. for $t = 15$ or $t = 20$). As reported in literature [104], such small-scale vortices can be generated by Kelvin-Helmholtz type instabilities of the vortex filaments for sufficiently high Reynolds numbers.

The energy density spectra for different time instants are plotted in Fig. 4.6. At time instant 0, the energy density is concentrated at one wave-number. At later time instances, an energy cascade develops with a slope of approximately k^{-3} covering almost two decades. This is in good correspondence with the k^{-3} slope for the direct enstrophy cascade as predicted by Kraichnan [105]. A slight steepening of the spectrum is observed at later time instances. Corroboration for this behavior is found in the literature, where the coherent vortices are reported to destroy scale invariance and produce steeper spectra than predicted by the k^{-3} direct enstrophy cascade [103]. The increase in the energy density contained in the low wave numbers at the expense of the energy density contained in the higher wave number range, further illustrates the formation of large-scale coherent structures. In short, it is concluded that the results for this high Reynolds number test are promising and demonstrate the robustness of the particle-mesh scheme.

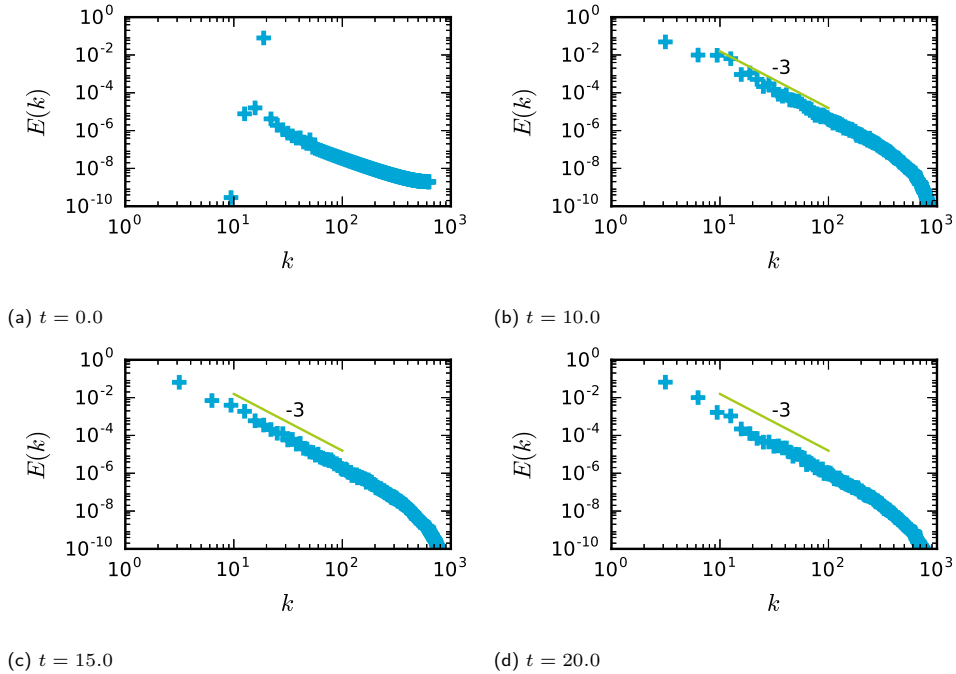


Figure 4.6: Taylor-Green flow: One-dimensional energy spectra for $\text{Re}=2.5 \cdot 10^5$ at different time instants (blue) with a k^{-3} slope (green).

4.4.3. Backward facing step

Flow over a backward facing step is considered for various Reynolds numbers. Due to flow separation at the location of the step, a stagnation point in the lower left corner, and the recirculation and reattachment of the flow past the step, this problem is a challenging test for particle(-mesh) methods, typically requiring heuristic particle shifting measures to maintain a uniform particle distribution [106].

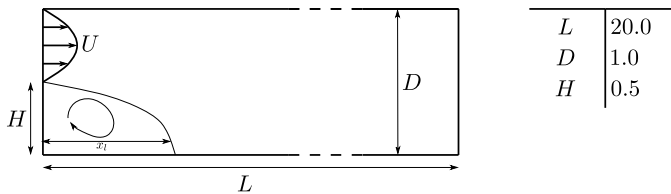


Figure 4.7: Backward facing step: model setup.

The problem setup is presented in Fig. 4.7. The step height H is equal to half the channel diameter D . At the inlet a parabolic velocity profile is prescribed with maximum velocity U_{\max} . Due to the abrupt expansion of the channel, a recirculation zone develops with a Reynolds number dependent reattachment length x_l . The Reynolds number is

defined as

$$Re = \frac{\bar{U}D}{\nu}, \quad (4.54)$$

with ν the kinematic viscosity and $\bar{U} = 2/3 U_{\max}$ denoting the mean inflow velocity - not to be confused with the bold-faced $\bar{\mathbf{U}}$.

A parabolic inflow velocity with $U_{\max} = 1$ is specified at the inflow boundary of the model domain $\Omega = [0, 20] \times [0, 1]$, that is, for $0.5 \leq y \leq 1$; see Fig. 4.7. At this boundary, particles are inserted, and the particle specific momentum is initialized based on an interpolation from the inflow boundary condition. At the outflow boundary at $x = 20$, a homogeneous Neumann boundary condition for the velocity is specified. At the top and the bottom boundary, the no-slip boundary condition $\mathbf{u} = \mathbf{0}$ is specified. The kinematic viscosity ν is adjusted in order to obtain Reynolds numbers ranging from 100 to 800. A regular triangular mesh with $(210 \times 16 \times 2)$ cells is used, and the polynomial order is set to $k = 2$, and HDG function spaces are used for the discretization of the Stokes equations. Initially, 200,000 particles are seeded in the domain, resulting in approximately 30 particles per cell on average. The time step size is set to $\Delta t = 0.05$ resulting in a CFL-number, based on the mean inflow velocity, of approximately 0.5 for all simulations. For a Reynolds number of 800, the stationary solution at time $t = 160$ is

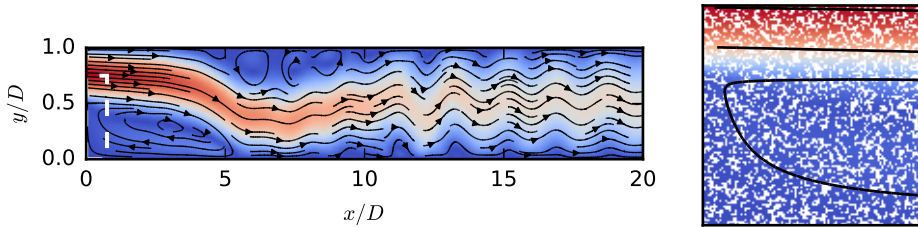


Figure 4.8: Backward facing step: particle velocities for $Re = 800$ at $t = 160$, and velocity streamlines obtained from particle velocities (black lines). A detailed view of the particle distribution on the square $[0, 0.75] \times [0, 0.75]$ near the lower-left stagnation point is shown at the right.

plotted in Fig. 4.8, showing the simulated particle velocity field with the corresponding streamlines. The computed streamlines involve a recirculation zone in the lower left corner and a secondary recirculation zone near the upper boundary. Of special interest are the particle distributions near the stagnation point in the lower left corner and in the recirculation zones. These zones pose difficulties to, e.g., the SPH method [106], where voids are observed. However, in our method the particle distribution remains uniform over time, even in these critical regions.

In Fig. 4.9, the computed dimensionless reattachment lengths obtained with the particle-mesh method are compared to values in literature obtained from measurements [107] and numerical experiments [6]. Overall, a very good agreement is found between the computed reattachment lengths and the values found in [6]. This result indicates that the artificial viscosity is low, since the reattachment length is strongly dependent on the Reynolds number. Similarly to the results obtained with the fully Eulerian approach from [6], the model results deviate from the experimental results for $Re > 400$. In var-

ious papers [6, 108] this is attributed to the three-dimensional structures present in the experimental setting, which cannot be captured with the two-dimensional computations.

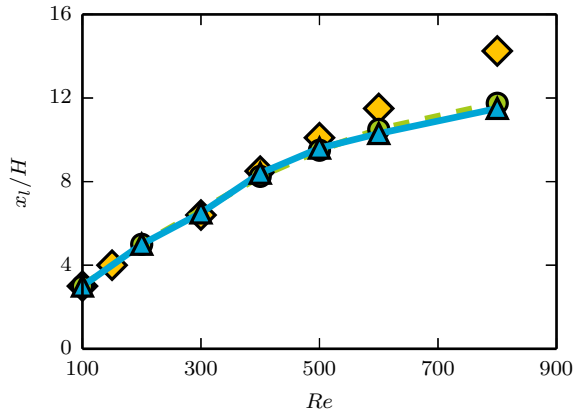


Figure 4.9: Backward facing step: computed reattachment lengths (\triangle) compared with experimental values (\diamond) from Armaly et al. [107], and numerical results (\circ) from Labeur and Wells [6].

4.4.4. Flow past a circular cylinder

The flow around a circular cylinder is considered as a final benchmark test. Due to the presence of a stagnation point at the upstream part of the cylinder and the occurrence of flow separation at the downstream part of the cylinder, this benchmark is a challenging test when it comes to maintaining a uniform particle distribution. Apart from assessing the quality of the particle distribution, this benchmark is also used to assess the accuracy of computed dynamic variables such as the drag and lift coefficient and the Strouhal number, where the latter indicates the frequency with which vortices are shedded from the cylinder at sufficiently high Reynolds numbers.

The setup from the benchmark paper by Schäfer and Turek [109] is used, see Fig. 4.10. A boundary-fitted, fully unstructured mesh is used to triangulate the domain of interest. At the inlet, a parabolic velocity profile is specified as follows,

$$U(0, y) = 4U_{\max}y(H - y)/H^2, \quad (4.55)$$

and the Reynolds number is defined by Eq. (4.54), using the mean inlet velocity $\bar{U} = 2/3U_{\max}$.

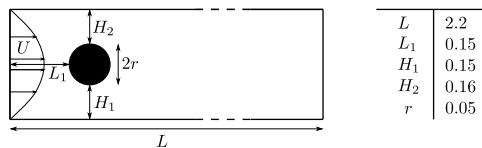


Figure 4.10: Flow past a circular cylinder: general setup (after [109]).

Following [109], the model is run for Reynolds numbers of 20 and 100 by setting U_{\max} equal to 0.3 and 1.5, respectively, while using a kinematic viscosity $\nu = 10^{-3}$ for both model runs. The polynomial order is set to $k = 2$, and HDG function spaces are used for the discretization of the Stokes equations. Furthermore, particles are inserted near the inflow boundaries. Other model parameters are listed in Table 4.7. The CFL-number

Table 4.7: Flow past a circular cylinder: model settings.

	$Re = 20$	$Re = 100$
U_{\max}	0.3	1.5
$h_{K,\min}$	0.011	0.011
Δt	$1e-2$	$5e-3$
$ \mathcal{T} $	8972	8972
$ \mathcal{S}_t $	418500	418500
CFL (approx.)	0.27	0.68

based on U_{\max} for the case $Re = 20$ is approximately 0.27, and for the case $Re = 100$ it is approximately 0.68. At the cylinder boundary the no-slip boundary condition is specified, and at the outflow boundary a homogeneous Neumann condition is imposed on the velocity. The particle distribution at $t = 8$ for the case $Re = 100$ is shown in

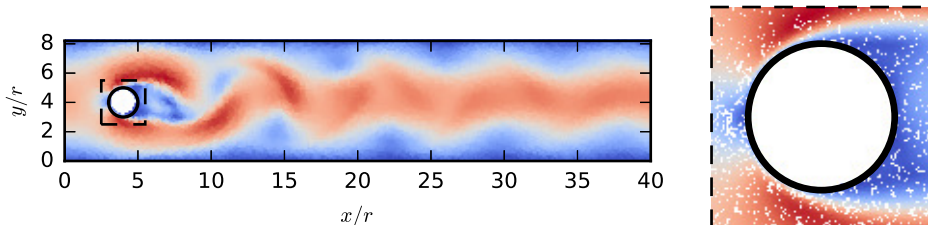


Figure 4.11: Flow past a circular cylinder: particle velocity field at $t = 8$ for $Re = 100$; a detailed view of the particle distribution near the circular cylinder is shown at the right.

Fig. 4.11. A Reynolds number of 100 is well above the threshold for the onset of vortex shedding, resulting in an unsteady flow characterized by vortices shed alternately from the cylinder. This process is also qualitatively observed in Fig. 4.11. This figure also shows a detailed view of the particle distribution in the vicinity of the cylinder. Clearly, the particle distribution remains uniform over time near the stagnation point and in the boundary layer. Moreover, neither unphysical voids nor clusters of particles are observed in the recirculation zone.

Various computed bulk quantities are compared with the values reported in [109] both for the low Reynolds number test case and the moderate Reynolds number test case. The drag and lift force are expressed in terms of coefficients C_D and C_L , respectively, which are defined as

$$C_D = \frac{F_x}{\rho \bar{U}^2 r}, \quad (4.56)$$

and

$$C_L = \frac{F_y}{\rho \bar{U}^2 r}, \quad (4.57)$$

where the force vector $\mathbf{F} = [F_x, F_y]^\top$ acting on the cylinder is computed with

$$\mathbf{F} = \oint_{\partial\Omega_C} \boldsymbol{\sigma}_h \mathbf{n} \, d\Gamma, \quad (4.58)$$

with $\boldsymbol{\sigma}_h$ given by Eq. (4.21) and $\partial\Omega_c$ denoting the outer boundary of the cylinder. Following [109] the pressure drop over the cylinder is computed as

$$\Delta P = P(L_1, H_1 + r) - P(L_1 + 2r, H_1 + r). \quad (4.59)$$

Finally, for the unsteady case ($Re = 100$), the Strouhal number is computed as

$$St = \frac{2rf}{\bar{U}}, \quad (4.60)$$

in which f is the frequency with which the flow separation occurs.

Results are tabulated in Table 4.8 in which also a comparison is made with the reference values from [109]. In general, a good correspondence is observed between the computed quantities and the reference values, the more so as the tabulated differences are well within the range of values presented in [109]. The results for this complex benchmark test demonstrate the accuracy and robustness of the proposed particle-mesh method.

Table 4.8: Flow past a circular cylinder: model results for different Reynolds numbers, compared to numerical values from Schäfer and Turek [109].

	$Re = 20$				$Re = 100$			
	C_D	C_L	ΔP	St	C_D	C_L	ΔP	St
Simulated	5.61	0.039	0.106	-	3.12	1.04	2.31	0.29
Reference	5.58	0.011	0.1174	-	3.23	1.00	2.48	0.30

4.5. Conclusion

This chapter proposed and validated a generic particle-mesh algorithm for the incompressible Navier-Stokes equations, based on a hybridized discontinuous Galerkin (HDG) framework. Analogous to the previous chapter, the developed method is cast into an operator splitting approach in which Lagrangian particles are used to discretize an advection operator and an Eulerian HDG approach is employed for the constitutive modeling. The coupling between the Lagrangian particles and the Eulerian mesh is established by means of two auxiliary projection steps: a particle-mesh projection and a mesh-particle projection. Similar to the scalar case, two approaches for the particle-mesh projection have been proposed. In one approach, the projections are formulated in terms of local (i.e. cellwise) ℓ^2 -minimization problems. To retain discrete convergence properties, a

second particle-mesh projection strategy extends the ℓ^2 minimization problem with a linearized PDE-constraint. Consistency and conservation of this particle-mesh projection strategy were proven. Moreover, both particle-mesh projection approaches allow for a straightforward extension to arbitrary-order spatial accuracy on general meshes.

The importance of accurate particle advection was emphasized, advocating the use of second-order accurate time integration methods to update the specific momentum carried by the particles and to update the particle positions by using velocity fields which are H(div)-conforming. These velocity fields are inherently obtained when using an HDG or EDG-HDG method to solve for the unsteady Stokes equations at the background mesh. Exploiting the facet variable, the PDE-constrained particle-mesh projection and the discrete Stokes system can be implemented efficiently via static condensation.

The developed scheme was validated for various single-phase problems. Optimal spatial convergence rates were obtained for laminar Poiseuille flow, indicating that the particle-mesh splitting strategy does not affect the spatial accuracy. For the presented time discretization method, second-order accuracy in time was demonstrated for the Taylor-Green test. A high Reynolds number Taylor-Green vortex instability demonstrated the robustness of the scheme, and illustrated that the developed method is virtually free of numerical diffusion.

Maintaining a uniform particle distribution over time was shown to be key to obtain accurate model results. This is achieved by advecting the particles by velocity fields being (nearly) locally divergence-free, and having sufficient inter-element continuity. When these conditions are met, the method is able to accurately resolve the flow near stagnation points, in recirculation zones, and near separation points without the need for additional quality control on the particle distribution. Furthermore, dynamic quantities, such as the reattachment lengths in the backward facing step benchmark and the forces acting on a cylinder, are accurately computed.

5

Application to multiphase flows

Flow phenomena involving the simultaneous transport of different fluids occur in a wide range of problems in industry and science. In environmental flows, this includes, for example, the mixing of fresh and salt water, as well as the violent breaking of waves on a coastal structure. Whereas in the former, the density differences are comparatively small, the breaking wave is an example in which the air and water density are of a vastly different order of magnitude. The presence of such strong density jumps in multifluid or multiphase problems poses challenges to numerical schemes, particularly so as the phase-separating interface typically undergoes complex topological changes, including coalescence and breakup.

A significant and ongoing research effort is therefore devoted to developing numerical schemes that can be used to solve flow problems involving complex interfaces. Existing solution strategies with this aim are broadly categorized in *surface tracking* techniques, and *surface capturing* strategies. In the former type of methods, the interface is explicitly tracked by either deforming the mesh, or by subdividing the mesh cells such that it conforms with the location of the interface. Mesh deforming strategies are typically based on an arbitrary Lagrangian Eulerian (ALE) formulation of the conservation laws, see e.g. [85], and require rezoning and remeshing strategies to accommodate complex interface topologies. Extended Finite Element (XFEM) [110, 111] and finite cell [112, 113] approaches subdivide the mesh cells such that the sub-cells match the interface. A particular challenge related to these methods is to cure the ill-conditioning of the resulting system matrix, due to the presence of cut cells [112, 113].

Surface capturing schemes rely on an auxiliary function for either, i) reconstructing the location of the interface at a fixed mesh, yielding so-called volume-of-fluid (VoF) approaches [114], or ii) to provide an implicit representation of the surface as is typical to level-set techniques [115]. These methods typically employ a one-fluid model approach, whereby the problem is posed as one set of equations for the entire domain. The auxiliary field - which provides the required information for determining the interface location - is

used to derive the material dependent quantities, such as density and viscosity. Propagating the location of the interface over time, requires the advection of the auxiliary field by solving a hyperbolic scalar conservation law. As such, surface capturing schemes typically inherit the challenges related to mesh-based advection schemes, most notably for level-set methods in which numerical diffusion may deteriorate the sharp interface between the separate phases, and satisfying discrete (mass) conservation is a non-trivial issue, see e.g. [116] and references. Fig. 5.1 illustrates different surface tracking and surface capturing schemes.

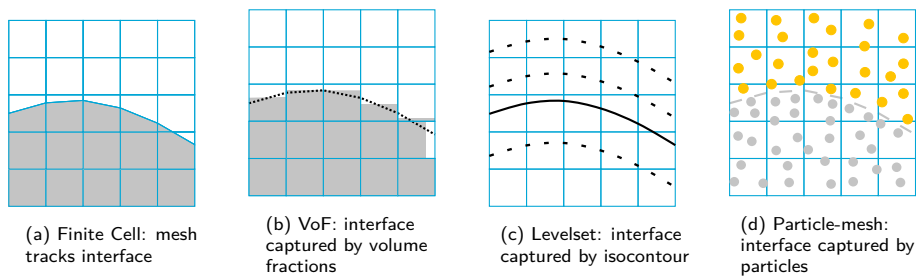


Figure 5.1: Sketch of different strategies to *track* (a) or *capture* (b - d) interfaces on a mesh.

This chapter proposes a surface capturing particle-mesh approach for simulating advection dominated, immiscible multiphase flows involving sharp jumps in the density. In the proposed approach, density and specific momentum fields are tracked at Lagrangian particles, see Fig. 5.1d, while the unsteady Stokes equations with variable density and viscosity are solved at the mesh level. Employing particles for tracking the density fields provides information on the location of the interface, hence obviating interface reconstruction or capturing techniques as in VoF and level-set methods. Furthermore, Chapter 3 showed that the combination of particles and a mesh can be rendered diffusion-free, high-order accurate, and conservative. This arguably is a powerful combination for tracking density fields in immiscible fluids, in which a sharp density jump between the phases is to be captured in the discrete solution, and where conservation properties are essential to avoid unphysical mass losses. Finally, attaching a specific momentum value to the particles in addition to a density value, primarily fits the purpose of including mesh motion in a straightforward manner as argued in Section 3.7. This feature can be exploited to impose conditions on moving boundaries.

The chapter proceeds as follows. The governing equations for immiscible multiphase flow are presented in Section 5.1. Section 5.2 highlights the formulation of a particle-mesh operator splitting for solving these equations. In particular, it focuses on the peculiarities related to the multiphase character of the flow problem. This includes the formulation of the constraint in the projection of the specific momentum, and the introduction of a penalty term to limit over- and undershoots in the reconstruction of the mesh density field from the particles. Various model properties are discussed in Section 5.3. Section 5.4 addresses some practical implementation issues in view of the numerical examples presented later on.

These numerical examples illustrate the potential and the challenges related to the

proposed particle-mesh scheme when applied to multiphase flows. Starting with the rising bubble example for relatively mild density jumps, Section 5.5 gradually builds towards problems involving large density jumps, typical to, e.g., air-water interfacial flows. The final examples in this chapter focus on the simulation of non-breaking and breaking waves, and a comparison is made between numerically and experimentally obtained impact forces for a breaking wave on a laboratory-scale coastal structure.

5.1. Model problem

The incompressible Navier-Stokes equations for immiscible fluids can be stated as: given a forcing term $\mathbf{f} : \Omega \times I \rightarrow \mathbb{R}^d$, comprising the body force \mathbf{f}_G and surface tension force \mathbf{f}_T , a dynamic viscosity $\mu > 0$, a solenoidal initial condition $\mathbf{u}^0 : \Omega \times I \rightarrow \mathbb{R}^d$, and initial density field $\rho^0 : \Omega \times I \rightarrow \mathbb{R}^d$ find the velocity \mathbf{u} , the pressure p and the density ρ such that

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla \cdot \boldsymbol{\sigma}_d = \mathbf{f} \quad \text{in } \Omega \times I, \quad (5.1a)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad \text{in } \Omega \times I, \quad (5.1b)$$

$$\boldsymbol{\sigma}_d = p\mathbf{I} - 2\mu\nabla^s \mathbf{u} \quad \text{in } \Omega \times I, \quad (5.1c)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times I, \quad (5.1d)$$

$$(\rho \mathbf{u} \otimes \mathbf{u}) \mathbf{n} = \mathbf{h}_a \quad \text{on } \Gamma_N^- \times I, \quad (5.1e)$$

$$(\rho \mathbf{u}) \cdot \mathbf{n} = h \quad \text{on } \Gamma^- \times I, \quad (5.1f)$$

$$\boldsymbol{\sigma}_d \mathbf{n} = \mathbf{h}_d \quad \text{on } \Gamma_N \times I, \quad (5.1g)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_D \times I, \quad (5.1h)$$

$$\mathbf{u}(\mathbf{x}, t^0) = \mathbf{u}^0(\mathbf{x}) \quad \text{in } \Omega, \quad (5.1i)$$

$$\rho(\mathbf{x}, t^0) = \rho^0(\mathbf{x}) \quad \text{in } \Omega, \quad (5.1j)$$

in which \mathbf{g} and \mathbf{h} values specified at the Dirichlet boundary or Neumann boundary, where \mathbf{h} is splitted into an advective part \mathbf{h}_a and a diffusive part \mathbf{h}_d . Note that the advective fluxes can only be prescribed at inflow parts of the boundaries (where $\mathbf{u} \cdot \mathbf{n} < 0$), which is indicated by the superscript $-$.

5.1.1. Operator splitting

Given the similarities between Eq. (5.1) and Eqs. (3.2, 4.1), it appears that the multiphase flow problem is amenable to a particle-mesh operator splitting using tools from the preceding chapters.

As before, the time interval of interest I is therefore partitioned into a sequence of half open subintervals $I_n = (t^n, t^{n+1}]$, with $\{t^0, t^1, \dots, t^{N-1}, t^N\}$ indicating the discrete time levels. In addition, we define a velocity field \mathbf{U} which is a consistent and solenoidal approximation to the advective velocity \mathbf{u} in Eq. (5.1a) and Eq. (5.1b), and piecewise constant over the time interval I_n . As in Chapter 4, this choice linearizes the non-linear advection term in Eq. (5.1a). Moreover, by introducing this field \mathbf{U} the momentum advection in Eq. (5.1a) is decoupled from the density advection, Eq. (5.1b).

Consistent with previous chapters, the following notations are adopted to anticipate a particle-mesh operator splitting of the multiphase flow problem. Let ψ denote the advected scalar-valued (density) field and let \mathbf{v} denote the advected vector-valued (specific) momentum field, whereas fields ρ and \mathbf{u} denote the density field and the specific momentum field throughout the constitutive part. Furthermore, let $\psi(\mathbf{x}, t^0) = \psi^0 = \rho^0$ be the initial condition on the scalar field.

After separating the advective terms from the constitutive terms, a spatiotemporal operator splitting of Eq. (5.1) involves: find a scalar field $\psi: \Omega \times I_n \rightarrow \mathbb{R}$ and a vector field $\mathbf{v}: \Omega \times I_n \rightarrow \mathbb{R}^d$ satisfying an advection problem

$$\frac{\partial \psi \mathbf{v}}{\partial t} + \nabla \cdot (\psi \mathbf{v} \otimes \mathbf{U}) = \mathbf{0} \quad \text{in } \Omega \times I_n, \quad (5.2a)$$

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \mathbf{U}) = 0 \quad \text{in } \Omega \times I, \quad (5.2b)$$

$$(\psi \mathbf{v} \otimes \mathbf{U}) \mathbf{n} = \mathbf{h}_a \quad \text{on } \Gamma_N^- \times I, \quad (5.2c)$$

$$(\psi \mathbf{U}) \cdot \mathbf{n} = h \quad \text{on } \Gamma^- \times I, \quad (5.2d)$$

$$\mathbf{v} = \mathbf{g} \quad \text{on } \Gamma_D^- \times I_n, \quad (5.2e)$$

$$\mathbf{v}(\mathbf{x}, t^n) = \mathcal{P}_L(\mathbf{u}(\mathbf{x}, t^n)) \quad \text{in } \Omega, \quad (5.2f)$$

$$\psi(\mathbf{x}, t^n) = \psi^n \quad \text{in } \Omega, \quad (5.2g)$$

and find a velocity field $\mathbf{u}: \Omega \times I_n \rightarrow \mathbb{R}^d$ satisfying an incompressible Stokes problem with variable density

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \boldsymbol{\sigma}_d = \mathbf{f} \quad \text{in } \Omega \times I_n, \quad (5.3a)$$

$$\frac{\partial \rho}{\partial t} = 0 \quad \text{in } \Omega \times I_n, \quad (5.3b)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times I_n, \quad (5.3c)$$

$$\boldsymbol{\sigma}_d = p \mathbf{I} - 2\mu \nabla^s \mathbf{u} \quad \text{in } \Omega \times I_n, \quad (5.3d)$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_D \times I_n, \quad (5.3e)$$

$$\boldsymbol{\sigma}_d \mathbf{n} = \mathbf{h}_d \quad \text{on } \Gamma_N \times I_n, \quad (5.3f)$$

$$\rho(\mathbf{x}, t^{n+1}) = \mathcal{P}_{E,I}(\psi(\mathbf{x}, t^{n+1})) \quad \text{in } \Omega, \quad (5.3g)$$

$$\mathbf{u}(\mathbf{x}, t^n) = \mathcal{P}_{E,II}(\rho(\mathbf{x}, t^{n+1}), \mathbf{v}(\mathbf{x}, t^{n+1})) \quad \text{in } \Omega. \quad (5.3h)$$

Eq. (5.2) and Eq. (5.3) are to be solved sequentially for every subinterval I_n , and notations for the Dirichlet Γ_D and the Neumann boundary conditions Γ_N are similar to the previous chapters. Furthermore, the projection operator $\mathcal{P}_{E,I}$ is introduced to couple the field ψ to ρ , and projection operators $\mathcal{P}_{E,II}$ and \mathcal{P}_L are required to couple fields \mathbf{v} and \mathbf{u} , where ρ and \mathbf{u} are defined on a mesh, while the fields ψ and \mathbf{v} are naturally defined on Lagrangian particles.

The latter assertion becomes clear by expanding the hyperbolic conservation law for the momentum advection, Eq. (5.2a), as

$$\mathbf{v} \left(\frac{\partial \psi}{\partial t} + \mathbf{U} \cdot \nabla \psi \right) + \psi \left(\frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \mathbf{U}) \right) = \mathbf{0}, \quad (5.4)$$

where it is recalled that the scalar valued field ψ represents the advected density field. With \mathbf{U} solenoidal and using Eq. (5.2b), it follows that the first term between brackets at the left-hand side is 0. Exploiting that \mathbf{U} is solenoidal once more, the remaining term in the momentum advection equation, Eq. (5.2a) is, together with the density advection equation, Eq. (5.2b), rewritten as

$$\frac{D\mathbf{v}}{Dt} := \frac{\partial\mathbf{v}}{\partial t} + \mathbf{U} \cdot \nabla\mathbf{v} = \mathbf{0}, \quad (5.5a)$$

$$\frac{D\psi}{Dt} := \frac{\partial\psi}{\partial t} + \mathbf{U} \cdot \nabla\psi = 0, \quad (5.5b)$$

where these equations are straightforwardly solved in a Lagrangian reference frame as will be shown below.

Analogous to Chapter 4, the incompressible Stokes problem Eq. (5.3) is solved at an Eulerian mesh. By virtue of Eq. (5.3b), the density ρ stays constant throughout this stage, and is prescribed by the particle-mesh projection operator $\mathcal{P}_{E,I}$, which reconstructs a density field ρ at the mesh from the scattered Lagrangian particles, Eq. (5.3g). A second particle-mesh operator, i.e. $\mathcal{P}_{E,II}$ in Eq. (5.3h), reconstructs a specific momentum field \mathbf{u} at the mesh. These two distinct particle-mesh projection operators are required to provide conditions for advancing the incompressible Stokes problem from t^n to t^{n+1} . Furthermore, the mesh-particle projection \mathcal{P}_L in Eq. (5.2f) is introduced to project the specific momentum from the mesh to the particles, thus providing the condition for advancing the advection problem in time over the next time interval. Similar techniques as derived in Chapter 3 and Chapter 4 are adopted to define these projection operators. Crucial, however, is that no mesh-particle projection operator is needed for advecting the scalar value field ψ , see Eq. (5.2g), for the obvious reason that the density evolution amounts to purely advective transport.

In view of the extension to moving meshes, use is made of the ‘convective velocity’

$$\mathbf{C} = \mathbf{U} - \hat{\mathbf{U}}, \quad (5.6)$$

when formulating the mesh-particle projections $\mathcal{P}_{E,I}$ and $\mathcal{P}_{E,II}$ as a PDE-constrained optimization problem. In this equation, \mathbf{U} is the physical transport velocity and $\hat{\mathbf{U}}$ is the velocity of the moving mesh, analogous Eq. (3.40). Note that mesh motion does not affect a Lagrangian, particle-based solution to Eq. (5.5).

5.1.2. Auxiliary definitions

To accommodate the PDE-constrained formulation for the particle-mesh projection of the density, $\mathcal{P}_{E,I}$, the function space definitions from Eqs. (3.4-3.6) are used for W_h , T_h , and the trace space \bar{W}_h , that is

$$W_h := \{w_h \in L^2(\mathcal{T}), w_h|_K \in P_k(K) \forall K \in \mathcal{T}\}, \quad (5.7)$$

$$T_h := \{\tau_h \in L^2(\mathcal{T}), \tau_h|_K \in P_l(K) \forall K \in \mathcal{T}\}, \quad (5.8)$$

$$\bar{W}_h := \{\bar{w}_h \in L^2(\mathcal{F}), \bar{w}_h|_F \in P_k(F) \forall F \in \mathcal{F}\}, \quad (5.9)$$

in which $P_{k,l}(K)$ denotes the set of polynomials of degree $(k, l) \geq (1, 0)$ on $K \in \mathcal{T}$, and $P_k(F)$ the set of polynomials of degree k on $F \in \mathcal{F}$. The PDE-constrained particle-

mesh projection $\mathcal{P}_{E,II}$ of the momentum is carried out in function spaces which are the vector equivalents of Eqs. (5.7-5.9), denoted by \mathbf{W}_h , \mathbf{T}_h and $\bar{\mathbf{W}}_h$, respectively.

For the discretization of the unsteady Stokes equations, EDG-HDG function space definitions are employed, see Eq. (4.9b). Hence, the facet function space for the velocity is continuous in the mesh vertices, i.e. $\bar{\mathbf{W}}_h \cap C^0(\mathcal{F})$, with $\bar{\mathbf{W}}_h$ the vector-valued equivalent of Eq. (5.9). The local function space for the velocity, \mathbf{W}_h , is defined as the vector-valued equivalent of Eq. (5.7). Analogous to Eq. (4.7) and Eq. (4.8), local and global pressure function spaces are defined by

$$Q_h := \{q_h \in L^2(\mathcal{T}), q_h|_K \in P_{k-1}(K) \forall K \in \mathcal{T}\}, \quad (5.10)$$

$$\bar{Q}_h := \{\bar{q}_h \in L^2(\mathcal{F}), \bar{q}_h|_F \in P_k(F) \forall F \in \mathcal{F}\}. \quad (5.11)$$

Since the focus in this chapter is on the applicability for practical cases rather than on detailed convergence and accuracy studies, the discussion is limited to $k = 1$, i.e. the class of linear polynomials for \mathbf{W}_h , $\bar{\mathbf{W}}_h$, \bar{Q}_h , and piecewise constants for the local pressure space Q_h . Function spaces for the Lagrange multiplier fields T_h and \mathbf{T}_h are also assumed piecewise constant, i.e. $l = 0$, which significantly simplifies the optimality system for the PDE-constrained density and momentum projection, see Section 3.6.2.

Particle notations follow the conventions adopted in the preceding chapters, and are recapitulated for completeness. The Lagrangian particle configuration in the domain Ω at a fixed time instant t is defined as

$$\mathcal{X}_t := \{\mathbf{x}_p(t) \in \Omega\}_{p=1}^{N_p}, \quad (5.12)$$

in which \mathbf{x}_p denotes the spatial coordinates of particle p and N_p is the number of particles. Furthermore, Lagrangian scalar and vector fields on the set of particles \mathcal{S}_t are defined as

$$\Psi_t := \{\psi_p(t) \in \mathbb{R}, \forall p \in \mathcal{S}_t\}, \quad (5.13)$$

$$\mathcal{V}_t := \{\mathbf{v}_p(t) \in \mathbb{R}^d, \forall p \in \mathcal{S}_t\}, \quad (5.14)$$

where ψ_p and $\mathbf{v}_p(t)$ denote the corresponding scalar (density) and vector-valued (specific momentum) quantities associated with particle p . Furthermore, \mathcal{S}_t is the index set of all particles, see Eq. (3.8).

5.2. Model formulation

Overall, the sequence of steps for the particle-mesh operator splitting of Eq. (5.1) resembles the steps for the incompressible Navier-Stokes equations in Chapter 4. However, the density projection has to be incorporated, and for the sake of generality we take mesh motion into account from the outset. Hence, the complete sequence of steps becomes:

- 1a. *Lagrangian discretization of the advection problems*, in order to solve Eq. (5.5) for the advection of the scalar field ψ and the vector field \mathbf{v} at the particle level;
- 1b. *(Optional) Advection of the mesh*, the mesh is advected using the mesh velocity $\hat{\mathbf{U}}$, see Eq. (5.6);

- 2a. *particle-mesh projection I*, a density field is reconstructed at the background mesh via the projection operator $\mathcal{P}_{E,I} : \Psi_t \rightarrow W_h$ in Eq. (5.3g);
- 2b. *particle-mesh projection II*, a momentum field at the background mesh is obtained via the projection $\mathcal{P}_{E,II} : \mathcal{V}_t \rightarrow \mathbf{W}_h$ in Eq. (5.3h);
4. *Eulerian discretization of the unsteady Stokes equations*, in order to solve Eqs. (5.3a)-(5.3f);
5. *mesh-particle projection*, in order to update the specific momentum at the particle level via the projection operator $\mathcal{P}_L : \mathbf{W}_h \rightarrow \mathcal{V}_t$ in Eq. (5.2f);

In-depth derivations for these steps were presented in preceding chapters, and we suffice to present the most important modifications needed in the multi-fluid setting. Furthermore, the formulations for the particle-mesh projections are presented for the PDE-constrained approach on moving meshes, from which the simpler local ℓ^2 -projections are straightforwardly obtained.

5.2.1. Lagrangian discretization of advection problems

In a Lagrangian, particle-based frame of reference, the advection problem for the scalar density field ψ and the vector field \mathbf{v} in Eq. (5.5) is written as a system of three ODEs:

$$\dot{\psi}_p(t) = 0 \quad \forall p \in \mathcal{S}_t, \quad (5.15a)$$

$$\dot{\mathbf{v}}_p(t) = \mathbf{0} \quad \forall p \in \mathcal{S}_t, \quad (5.15b)$$

$$\dot{\mathbf{x}}_p(t) = \mathbf{U}_h(\mathbf{x}_p(t), t^n) \quad \forall p \in \mathcal{S}_t, \quad (5.15c)$$

in which $\dot{\psi}_p$, $\dot{\mathbf{v}}_p$ and $\dot{\mathbf{x}}_p(t)$ the total time derivatives at time $t \in I_n$ of the scalar density, the vector valued specific momentum and the position of particle p . As in Chapter 4, the advective velocity field $\mathbf{U}_h(\mathbf{x}, t)$ for $t \in I_n$, uses the mesh related velocity field $\mathbf{u}_h \in \mathbf{W}_h$ from the Stokes step and frozen at the old time level t^n , i.e. (see Eq. (4.32))

$$\mathbf{U}_h(\mathbf{x}_p(t), t) = \mathbf{u}_h(\mathbf{x}_p(t), t^n), \quad (5.16)$$

thereby inheriting the H(div) conformity of the field $\mathbf{u}_h(\mathbf{x}_p(t), t^n) \in \mathbf{W}_h$.

From Eq. (5.15a) and Eq. (5.15b) it follows that the scalar value ψ_p and the vector value \mathbf{v}_p , carried by particle p , stay constant throughout the advection stage. Moreover, since the mesh-based steps do not affect the scalar particle field via a mesh-particle projection, ψ_p will stay constant throughout the computation.

5.2.2. PDE-constrained density map

The particle-mesh projection $\mathcal{P}_{E,I} : \Psi_t \rightarrow W_h$ for reconstructing a scalar density field at the mesh from the Lagrangian scalar values ψ_p is further specified in this section. In line with the adopted conventions, the reconstructed field is denoted by ψ_h during the advection stage. The projection is carried out via a PDE-constrained projection, in which the objective function minimizes the difference between the particle field and the unknown density field ψ_h in a least squares sense, subjected to the hyperbolic conservation law for the density advection. This constraint is given by Eq. (5.2b) complemented

with boundary and initial condition, Eq. (5.2d) and Eq. (5.2g), respectively. In order to formulate this constraint on a moving mesh, convective velocities

$$\bar{\mathbf{C}}_h := \bar{\mathbf{U}}_h - \hat{\mathbf{U}}_h, \quad \text{and} \quad \mathbf{C}_h := \mathbf{U}_h - \hat{\mathbf{U}}_h, \quad (5.17)$$

are introduced in which $\bar{\mathbf{U}}_h \in \bar{\mathbf{W}}_{h,g}$ the transport velocity on the mesh facets and $\hat{\mathbf{U}}_h$ the mesh velocity. Furthermore, use will be made of the deformation tensor $\mathbf{F} = \hat{\nabla} \mathbf{x}$ - with $\hat{\nabla}$ the gradient on the reference domain - and the determinant $J = \det \mathbf{F}$.

Following the procedure detailed in Section 3.3.2, and assuming $l = 0$ for the Lagrange multiplier space, the following Lagrangian functional is obtained:

$$\begin{aligned} \mathcal{L}_I(\psi_h, \bar{\psi}_h, \lambda_h) = & \sum_{p \in \mathcal{S}_t} \frac{1}{2} (\psi_h(\mathbf{x}_p(t), t) - \psi_p)^2 + \sum_K \oint_{\partial K} \frac{1}{2} \beta (\bar{\psi}_h - \psi_h)^2 d\Gamma \\ & + \sum_K \int_K \frac{1}{2} \zeta \|\nabla \psi_h\|^2 d\Omega + \frac{d}{dt} \int_{\hat{\Omega}} J \psi_h \lambda_h d\hat{\Omega} + \sum_K \oint_{\partial K \setminus \hat{\Gamma}} J (\mathbf{F}^{-\mathbf{T}} \bar{\mathbf{C}}) \cdot \hat{\mathbf{n}} \bar{\psi}_h \lambda_h d\hat{\Gamma} \\ & + \oint_{\hat{\Gamma}^+} J (\mathbf{F}^{-\mathbf{T}} \mathbf{C}) \cdot \hat{\mathbf{n}} \psi_h \lambda_h d\hat{\Gamma} + \oint_{\hat{\Gamma}^-} \hat{h} \lambda_h d\hat{\Gamma}, \quad (5.18) \end{aligned}$$

with $\psi_p(t) \in \Psi_t$ given, the constraint formulated on the reference domain $\hat{\Omega}$, see Eq. (3.41), and $\hat{\Gamma}^+$ and $\hat{\Gamma}^-$ the out- and inflow parts of the boundary, respectively. The inclusion of a penalty term on density gradients, governed by ζ , is the most important difference compared to Eq. (3.41). This term is introduced to penalize over- and undershoot in the vicinity of density jumps, and is further elaborated in Section 5.4.

The derivation of the semi-discrete optimality system and their fully-discrete counterparts was extensively discussed in Chapter 3, and in particular Section 3.7 discussed a strategy to formulate the constraint on the spatial domain. Hence, the fully-discrete optimality system for Eq. (5.18) reads in analogy to Eq. (3.44): given the particle field $\psi_p^n \in \Psi_t$, the particle positions $\mathbf{x}_p^{n+1} \in \mathcal{X}_t$, and the density field at the old time level $\psi_h^n \in W_h$, find $(\psi_h^{n+1}, \lambda_h^{n+1}, \bar{\psi}_h^{n+1}) \in (W_h, T_h, \bar{W}_{h,g})$ such that

$$\begin{aligned} & \sum_{p \in \mathcal{S}_t} (\psi_h^{n+1}(\mathbf{x}_p^{n+1}) - \psi_p^n) w_h(\mathbf{x}_p^{n+1}) \\ & - \sum_K \oint_{\partial K} \beta (\bar{\psi}_h^{n+1} - \psi_h^{n+1}) w_h d\Gamma + \sum_K \int_K \zeta \nabla \psi_h^{n+1} \cdot \nabla w_h d\Omega \\ & + \frac{1}{\Delta t} \int_{\Omega} w_h \lambda_h^{n+1} d\Omega + \oint_{\Gamma^+} \mathbf{C}_h^n \cdot \mathbf{n} \lambda_h^{n+1} w_h d\Gamma = 0 \quad \forall w_h \in W_h. \quad (5.19a) \end{aligned}$$

$$\begin{aligned} \frac{1}{\Delta t} \left(\int_{\Omega} \psi_h^{n+1} \tau_h \, d\Omega - \int_{\hat{\Omega}} \psi_h^n \tau_h \, d\hat{\Omega} \right) + \sum_K \oint_{\partial K \setminus \Gamma} \mathbf{C}_h^n \cdot \mathbf{n} \bar{\psi}_h^{n+1} \tau_h \, d\Gamma \\ + \oint_{\Gamma^+} \mathbf{C}_h^n \cdot \mathbf{n} \psi_h^{n+1} \tau_h \, d\Gamma + \oint_{\Gamma^-} h^{n+1} \tau_h \, d\Gamma = 0 \quad \forall \tau_h \in T_h, \end{aligned} \quad (5.19b)$$

$$\sum_K \oint_{\partial K \setminus \Gamma} \mathbf{C}_h^n \cdot \mathbf{n} \lambda_h^{n+1} \bar{w}_h \, d\Gamma + \sum_K \oint_{\partial K} \beta (\bar{\psi}_h^{n+1} - \psi_h^{n+1}) \bar{w}_h \, d\Gamma = 0 \quad \forall \bar{w}_h \in \bar{W}_{h,0}. \quad (5.19c)$$

In these equations, Ω indicates the current spatial configuration of the mesh, and $\hat{\Omega}$ indicates the reference configuration, with the latter chosen equal to the domain at time level t^n , see Eq. (3.44b).

Solving the optimality system Eq. (5.19) for $(\psi_h^{n+1}, \lambda_h^{n+1}, \bar{\psi}_h^{n+1}) \in (W_h, T_h, \bar{W}_{h,g})$, yields the reconstructed density field ψ_h^{n+1} which is used in the momentum projection $\mathcal{P}_{E,II}$ and in the unsteady Stokes solver, with these steps being discussed next.

5.2.3. PDE-constrained momentum map

The momentum field at the mesh is reconstructed via the projection $\mathcal{P}_{E,II} : \mathcal{V}_t \rightarrow \mathbf{W}_h$. This is either done by projecting the particle momenta $\psi_p \mathbf{v}_p$, or by projecting the specific particle momenta \mathbf{v}_p . In the latter case, the momentum field at the mesh is obtained by taking the product $\psi_h \mathbf{v}_h$. In this thesis, the specific momentum field \mathbf{v}_p is projected, with this choice motivated by the expected smoothness of \mathbf{v}_p compared to $\psi_p \mathbf{v}_p$ which is discontinuous as the result of discontinuities in ψ_p .

Momentum conservation throughout this step is ensured by imposing the hyperbolic conservation law Eq. (5.2a) with boundary condition Eq. (5.2c) as a constraint to the ℓ^2 objective function. Hence, the PDE-constrained particle-mesh projection of the specific momentum is based on finding the stationary points of the Lagrangian

$$\begin{aligned} \mathcal{L}_{II}(\mathbf{v}_h, \bar{\mathbf{v}}_h, \boldsymbol{\lambda}_h) &= \sum_p \frac{1}{2} (\mathbf{v}_h(\mathbf{x}_p(t), t) - \mathbf{v}_p(t))^2 + \sum_K \oint_{\partial K} \frac{1}{2} \beta (\bar{\mathbf{v}}_h - \mathbf{v}_h)^2 \, d\Gamma \\ &+ \frac{d}{dt} \int_{\hat{\Omega}} J \psi_h \mathbf{v}_h \cdot \boldsymbol{\lambda}_h \, d\hat{\Omega} + \sum_K \oint_{\partial \hat{K} \setminus \hat{\Gamma}_N} J (\bar{\psi}_h \bar{\mathbf{v}}_h \otimes (\mathbf{F}^- \bar{\mathbf{C}}_h)) \hat{\mathbf{n}} \cdot \boldsymbol{\lambda}_h \, d\hat{\Gamma} \\ &+ \oint_{\hat{\Gamma}_N^+} (\psi_h \mathbf{v}_h \otimes (\mathbf{F}^- \bar{\mathbf{C}}_h)) \hat{\mathbf{n}} \cdot \boldsymbol{\lambda}_h \, d\hat{\Gamma} + \oint_{\hat{\Gamma}_N^-} \hat{\mathbf{h}}_a \cdot \boldsymbol{\lambda}_h \, d\hat{\Gamma}. \end{aligned} \quad (5.20)$$

In the discrete setting, $\psi_h \in W_h$ and $\bar{\psi}_h \in \bar{W}_h$ are known fields, obtained in the preceding PDE-constrained density projection. Furthermore, using convective velocity

fields \mathbf{C}_h and $\bar{\mathbf{C}}_h$ defined by Eq. (5.16) and Eq. (5.17) effectively linearize the non-linear advection term.

Performing the time discretization and carrying out the optimization results in the following fully-discrete optimality system: given the vector-valued particle field $\mathbf{v}_p^n \in \mathcal{V}_t$, the particle positions $\mathbf{x}_p^{n+1} \in \mathcal{X}_t$, the intermediate field $\mathbf{v}_h^{*,n} \in \mathbf{W}_h$ defined via Eq. (4.35), and density fields ψ_h^{n+1} and ψ_h^n , find $(\mathbf{v}_h^{n+1}, \boldsymbol{\lambda}_h^{n+1}, \bar{\mathbf{v}}_h^{n+1}) \in (\mathbf{W}_h, \mathbf{T}_h, \bar{\mathbf{W}}_{h,g})$ such that the time-discrete co-state equation

$$\begin{aligned} & \sum_{p \in \mathcal{S}_t} (\mathbf{v}_h^{n+1}(\mathbf{x}_p^{n+1}) - \mathbf{v}_p^n) \cdot \mathbf{w}_h(\mathbf{x}_p^{n+1}) - \sum_K \oint_{\partial K} \beta (\bar{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) \cdot \mathbf{w}_h \, d\Gamma \\ & + \frac{1}{\Delta t} \int_{\Omega} \psi_h^{n+1} \mathbf{w}_h \cdot \boldsymbol{\lambda}_h^{n+1} \, d\Omega + \oint_{\Gamma_N^+} (\psi_h^{n+1} \mathbf{w}_h \otimes \mathbf{C}_h^n) \mathbf{n} \cdot \boldsymbol{\lambda}_h^{n+1} \, d\Gamma = 0 \quad \forall \mathbf{w}_h \in \mathbf{W}_h, \end{aligned} \quad (5.21a)$$

is satisfied. Simultaneously the fully-discrete state equation

$$\begin{aligned} & \frac{1}{\Delta t} \left(\int_{\Omega} \psi_h^{n+1} \mathbf{v}_h^{n+1} \cdot \boldsymbol{\tau}_h \, d\Omega - \int_{\hat{\Omega}} \psi_h^n \mathbf{v}_h^{*,n} \cdot \boldsymbol{\tau}_h \, d\hat{\Omega} \right) + \sum_K \oint_{\partial K \setminus \Gamma_N} (\bar{\psi}_h^{n+1} \bar{\mathbf{v}}_h^{n+1} \otimes \bar{\mathbf{C}}_h^n) \mathbf{n} \cdot \boldsymbol{\tau}_h \, d\Gamma \\ & + \oint_{\Gamma_N^+} (\psi_h^{n+1} \mathbf{v}_h^{n+1} \otimes \mathbf{C}_h^n) \mathbf{n} \cdot \boldsymbol{\tau}_h \, d\Gamma + \oint_{\Gamma_N^-} \mathbf{h}_a^{n+1} \cdot \boldsymbol{\tau}_h \, d\Gamma = 0 \quad \forall \boldsymbol{\tau}_h \in \mathbf{T}_h, \end{aligned} \quad (5.21b)$$

and the fully-discrete optimality condition

$$\sum_K \oint_{\partial K \setminus \Gamma_N} (\boldsymbol{\lambda}_h^{n+1} \otimes \mathbf{u}_h^n) \mathbf{n} \cdot \psi_h^{n+1} \bar{\mathbf{w}}_h \, d\Gamma + \sum_K \oint_{\partial K} \beta (\bar{\mathbf{v}}_h^{n+1} - \mathbf{v}_h^{n+1}) \cdot \bar{\mathbf{w}}_h \, d\Gamma = 0 \quad \forall \bar{\mathbf{w}}_h \in \bar{\mathbf{W}}_{h,0}, \quad (5.21c)$$

are to be satisfied. In these equations, Ω indicates the current spatial configuration of the mesh, and $\hat{\Omega}$ indicates the reference configuration, chosen equal to the domain at time level t^n . Also, $\hat{\boldsymbol{\tau}}_h$ implies evaluation of the test function $\boldsymbol{\tau}_h$ on the reference domain.

Solving the optimality system, Eq. (5.21), yields a (specific) momentum field $\mathbf{v}_h^{n+1} \in \mathbf{W}_h$ which, together with the reconstructed density field $\psi_h^{n+1} \in W_h$ is used in the subsequent Stokes solve.

5.2.4. Multiphase Stokes solver

The Lagrangian particle advection in conjunction with the constrained particle-mesh projections yield a reconstructed density field $\rho_h^{n+1} = \psi_h^{n+1} \in W_h$, and an intermediate velocity field $\mathbf{u}_h^{*,n} := \mathbf{v}_h^{n+1} \in \mathbf{W}_h$. These fields serve as initial conditions for the unsteady, multiphase Stokes solver in order to account for the diffusive forces, and to enforce the incompressibility constraint. Employing a EDG-HDG discretization to fit

this purpose results in: find $(\mathbf{u}_h^{n+1}, \bar{\mathbf{u}}_h^{n+1}, p_h^{n+1}, \bar{p}_h^{n+1}) \in (\mathbf{W}_h, \bar{\mathbf{W}}_h \cap C^0(\mathcal{F}), Q_h, \bar{Q}_h)$ such that (local and global momentum balance)

$$\begin{aligned} & \int_{\Omega} \frac{\rho_h^{n+1} \mathbf{u}_h^{n+1} - \rho_h^{n+1} \mathbf{u}_h^{*,n}}{\Delta t} \cdot \mathbf{w}_h \, d\Omega - \sum_K \int_K \boldsymbol{\sigma}_{d,h}^{n+1} : \nabla \mathbf{w}_h \, d\Omega + \sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h}^{n+1} \mathbf{n} \cdot \mathbf{w}_h \, d\Gamma \\ & + \sum_K \oint_{\partial K} 2\mu (\bar{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^{n+1}) \cdot \nabla^s \mathbf{w}_h \mathbf{n} \, d\Gamma = \int_{\Omega} (\mathbf{f}_G^{n+1} + \mathbf{f}_T^{n+1}) \cdot \mathbf{w}_h \, d\Omega \quad \forall \mathbf{w}_h \in \mathbf{W}_h, \end{aligned} \quad (5.22a)$$

$$\sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h}^{n+1} \mathbf{n} \cdot \bar{\mathbf{w}}_h \, d\Gamma = \oint_{\Gamma_N} \mathbf{h}_d^{n+1} \cdot \bar{\mathbf{w}}_h \, d\Gamma \quad \forall \bar{\mathbf{w}}_h \in (\bar{\mathbf{W}}_h \cap C^0(\mathcal{F})), \quad (5.22b)$$

and (local and global mass balances),

$$\sum_K \int_K \mathbf{u}_h^{n+1} \cdot \nabla q_h \, d\Omega - \sum_K \oint_{\partial K} \mathbf{u}_h^{n+1} \cdot \mathbf{n} q_h \, d\Gamma = 0 \quad \forall q_h \in Q_h, \quad (5.22c)$$

$$\sum_K \oint_{\partial K} \mathbf{u}_h^{n+1} \cdot \mathbf{n} \bar{q}_h \, d\Gamma - \oint_{\partial\Omega} \bar{\mathbf{u}}_h^{n+1} \cdot \mathbf{n} \bar{q}_h \, d\Gamma = 0 \quad \forall \bar{q}_h \in \bar{Q}_h, \quad (5.22d)$$

are satisfied, on the current spatial configuration of the domain Ω . In these equations, the diffusive flux $\boldsymbol{\sigma}_{d,h}^{n+1}$ is given by

$$\boldsymbol{\sigma}_{d,h}^{n+1} = p_h^{n+1} \mathbf{I} - 2\mu \nabla^s \mathbf{u}_h^{n+1}, \quad (5.23)$$

and the diffusive flux at cell facets, $\hat{\boldsymbol{\sigma}}_{d,h}^{n+1}$, is defined as

$$\hat{\boldsymbol{\sigma}}_{d,h}^{n+1} = \bar{p}_h^{n+1} \mathbf{I} - 2\mu \nabla^s \mathbf{u}_h^{n+1} - 2\mu \frac{\alpha}{h_K} (\bar{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^{n+1}) \otimes \mathbf{n}, \quad (5.24)$$

in which μ the dynamic viscosity. Since the density stays constant during the Stokes step by virtue of Eq. (5.3b), the density field only needs evaluation at one time level, and we choose the time level $n+1$ to fit this purpose, i.e. ρ_h^{n+1} . Furthermore, \mathbf{f}_G^{n+1} is the body force, typically given by $\rho_h^{n+1} \mathbf{g}$ with \mathbf{g} the gravitational acceleration vector, and \mathbf{f}_T^{n+1} is a volumetric formulation of the surface tension force, for which a simple discretization via a continuum surface force (CSF) approach [117] is presented in Appendix C.

Mesh-particle projection

After solving Eq. (5.22), the particle specific momentum is updated via the projection operator $\mathcal{P}_L : \mathbf{W}_h \rightarrow \mathcal{V}_t$ similar to the procedure outlined in Section 4.3.4, that is

$$\mathbf{v}_p^{n+1} = \mathbf{v}_p^n + \Delta t_n ((1 - \theta_L) \dot{\mathbf{u}}_h^n(\mathbf{x}_p^n) + \theta_L \dot{\mathbf{u}}_h^{n+1}(\mathbf{x}_p^{n+1})) \quad \forall p \in \mathcal{S}_t, \quad (5.25)$$

with $1/2 \leq \theta_L \leq 1$, and $\dot{\mathbf{u}}_h^n(\mathbf{x}_p^n)$ and $\dot{\mathbf{u}}_h^{n+1}(\mathbf{x}_p^{n+1})$ denoting the mesh-based accelerations at the respective time levels n and $n+1$, evaluated at the individual particle positions. Since the density does not change over the unsteady Stokes step, the density value carried by the Lagrangian particles needs no updating, i.e. $\psi_p^{n+1} = \psi_p^n$.

5.3. Model properties

Conservation over the different model steps is demonstrated in the first part of this section. The second part of this section assesses the ability of the scheme to establish hydrostatic equilibrium.

5.3.1. Conservation

To ensure mass conservation, the only requirement is that the particle-mesh projection for the density is conservative. Global and local conservation properties of the PDE-constrained projection for a density field were proven in Chapter 3. That is, local and global mass conservation over the PDE-constrained particle-mesh projection $\mathcal{P}_{E,I}$ follows from the discrete state equation in the optimality system resulting from Eq. (5.18), and mass conservation properties are not affected by the density gradient penalty term governed by ζ .

To guarantee momentum conservation, it is required that both the particle-mesh projection for the momentum and the unsteady Stokes step are conservative. Starting with the latter, local momentum conservation follows by setting $\mathbf{w}_h = \mathbf{e}_j$ in Eq. (5.22a) on a cell K and $\mathbf{w}_h = \mathbf{0}$ on $\mathcal{T} \setminus K$, with \mathbf{e}_j the canonical unit vector:

$$\int_K \frac{\rho_h^{n+1} \mathbf{u}_h^{n+1} - \rho_h^{n+1} \mathbf{u}_h^{*,n}}{\Delta t} \cdot \mathbf{e}_j \, d\Omega = \int_K (\mathbf{f}_G^{n+1} + \mathbf{f}_T^{n+1}) \cdot \mathbf{e}_j \, d\Omega - \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h}^{n+1} \mathbf{n} \cdot \mathbf{e}_j \, d\Gamma. \quad (5.26)$$

Assuming that $\Gamma_D = \emptyset$, global momentum conservation over the Stokes step follows by setting $\mathbf{w}_h = \mathbf{e}_j$ in Eq. (5.22a) and $\bar{\mathbf{w}}_h = -\mathbf{e}_j$ in Eq. (5.22b), and summing the results:

$$\int_{\Omega} \frac{\rho_h^{n+1} \mathbf{u}_h^{n+1} - \rho_h^{n+1} \mathbf{u}_h^{*,n}}{\Delta t} \cdot \mathbf{e}_j \, d\Omega = \int_K (\mathbf{f}_G^{n+1} + \mathbf{f}_T^{n+1}) \cdot \mathbf{e}_j \, d\Omega - \oint_{\Gamma} \mathbf{h}_d^{n+1} \cdot \mathbf{e}_j \, d\Gamma. \quad (5.27)$$

Conservation of momentum over the projection $\mathcal{P}_{E,II}$ requires investigation of the state equation Eq. (5.21b). Substituting $\boldsymbol{\tau}_h = \mathbf{e}_j$, with \mathbf{e}_j a canonical unit basis vector, on cell K in the spatial domain and the corresponding cell \hat{K} in the reference domain, and setting $\boldsymbol{\tau}_h = \mathbf{0}$ on $\mathcal{T} \setminus K$ into this equation, leads after rearrangement to

$$\begin{aligned} \frac{1}{\Delta t} \left(\int_K \rho_h^{n+1} \mathbf{v}_h^{n+1} \cdot \mathbf{e}_j \, d\Omega - \int_{\hat{K}} \rho_h^n \mathbf{v}_h^{*,n} \cdot \mathbf{e}_j \, d\hat{\Omega} \right) \\ = - \oint_{\partial K} \left(\bar{\rho}_h^{n+1} \bar{\mathbf{v}}_h^{n+1} \otimes \bar{\mathbf{C}}_h^n \right) \mathbf{n} \cdot \mathbf{e}_j \, d\Gamma, \quad (5.28) \end{aligned}$$

where, for the sake of simplicity, the cell K is assumed interior to the domain, thus not sharing facets with the domain boundary $\partial\Omega$. The temporal change in the local storage over the deforming cell, i.e. the left-hand side in Eq. (5.28), is balanced by the momentum flux term at the right-hand side, which is single-valued at interfaces.

Hence, Eq. (5.28) is a local conservation statement if $\rho_h^n \mathbf{v}_h^{*,n}$ represents the momentum after the Stokes solve at the old time level, i.e. $\rho_h^n \mathbf{u}_h^n$ in the conservation statements Eq. (5.26) and Eq. (5.27). Recalling that $\mathbf{v}_h^{*,n}$ is defined via Eq. (4.35) as

$$\mathbf{v}_h^{*,n} = \mathbf{v}_h^n + \Delta t_n \left((1 - \theta_L) \dot{\mathbf{u}}_h^{n-1} + \theta_L \dot{\mathbf{u}}_h^n \right), \quad (5.29)$$

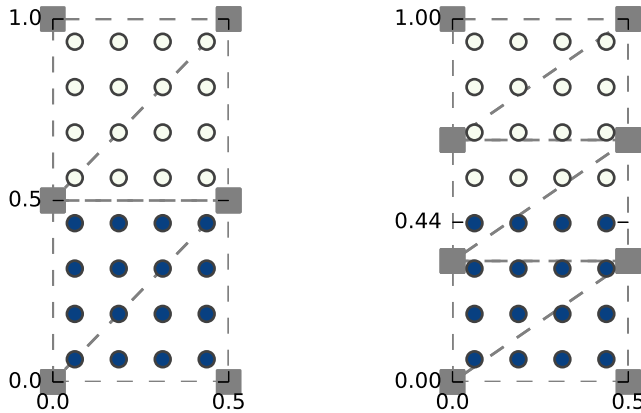
the product $\rho_h^n \mathbf{v}_h^{*,n}$ only equals the momentum at the old time level, $\rho_h^n \mathbf{u}_h^n$, when $\theta_L = 1$ in Eq. (5.29). This follows by rewriting Eq. (5.29) for $\theta_L = 1$ as

$$\mathbf{v}_h^{*,n} = \mathbf{u}_h^{*,n-1} + \Delta t_n \frac{\mathbf{u}_h^n - \mathbf{u}_h^{*,n-1}}{\Delta t_n} := \mathbf{u}_h^n. \quad (5.30)$$

Henceforth, the definition for $\mathbf{v}_h^{*,n}$ via Eq. (5.30) is used. The value of θ_L in the update of the particle specific momentum, is kept at a value of 1/2, at the expense of introducing a small abuse on consistency, see the discussion in Section 3.6.1. Other possible definitions for $\rho_h^n \mathbf{v}_h^{*,n}$ are not further considered.

5.3.2. Hydrostatic equilibrium balance

Since steep gradients in the density field are to be expected, the question arises to what extent the pressure gradient in the proposed scheme is able to balance the body force under hydrostatic equilibrium conditions. This is subject of ongoing research in, e.g., atmospheric and oceanographic sciences in which the dynamics of the problems of interest are typically governed by small deviations from the hydrostatic and geostrophic balance [118]. Also in the scope of air-water interfacial flows for which the diffusion is low, it is anticipated that accurately satisfying the hydrostatic balance is paramount to avoid interface artifacts.



(a) Case 1: mesh and interface aligned.

(b) Case 2: mesh and interface not aligned.

Figure 5.2: Hydrostatic balance: set-up of hydrostatic balance test with (a) aligned and (b) non-aligned interface. Particles representing the dense fluid are blue colored.

The model capabilities for establishing hydrostatic equilibrium are heuristically investigated by considering the following minimal example. Let the domain $\Omega := [0, 0.5] \times [0, 1]$

be triangulated into 4 (henceforth labeled ‘Case 1’), respectively 6 (‘Case 2’) equally-sized cells, on which 4×8 particles are placed in a regular lattice. Furthermore, let the interface between Fluid 1 (with $\rho_1 = 10 \text{ kg.m}^{-3}$, $\mu_1 = 1\text{e-}5 \text{ kg.m}^{-1}.\text{s}^{-1}$) and Fluid 2 (with $\rho_2 = 1 \text{ kg.m}^{-3}$, $\mu_2 = 1\text{e-}6 \text{ kg.m}^{-1}.\text{s}^{-1}$) be located at $y = 0.5$, so that for Case 1 the fluid interface and the mesh interface are aligned, Fig. 5.2a, whereas this is not so for Case 2, Fig. 5.2b. Initially, the fluid is at rest, and hydrostatic equilibrium should be maintained, i.e. the pressure gradient should balance the body force. To assess whether or not this is the case, the particle-mesh scheme is performed for one time step $\Delta t = 2.5\text{e-}2$. Two different reconstructions of the density field are compared:

- *FEM*: the discontinuous initial density is projected onto $\rho_h \in W_h$, as if there are no particles involved.
- *PDE*: particle density values are initialized from the discontinuous density field, after which the particle densities are projected onto $\rho_h \in W_h$ using the PDE-constrained particle-mesh projection, with $\beta = 1\text{e-}6$ and $\zeta = 20$, where this choice for ζ will be motivated in Section 5.4.

Since the initial velocity field equals $\mathbf{0}$ everywhere, and only one time step is performed, the particle advection and momentum projection step are omitted.

A first prerequisite to maintain hydrostatic equilibrium is that the density gradients in x -direction are zero, and velocities remain zero. To elucidate this, consider the L^2 -norm of the local density gradient in x -direction

$$\epsilon_{\nabla_x \rho} = \left(\int_{\Omega} \left| \frac{\partial \rho_h}{\partial x} \right|^2 d\Omega \right)^{1/2}, \quad (5.31)$$

and the L^2 -norm of the velocity:

$$\epsilon_{\mathbf{u}} = \left(\int_{\Omega} |\mathbf{u}_h|^2 d\Omega \right)^{1/2}. \quad (5.32)$$

The computed values for these metrics are tabulated in Table 5.1, where - in the case of hydrostatic equilibrium - deviations from machine zero can be considered as errors. For Case 1, the different projections indeed return machine zero for $\epsilon_{\nabla_x \rho}$, and $\epsilon_{\mathbf{u}}$ also approximates machine zero as a result. However, using the PDE-constrained particle-mesh projection to reconstruct a density field at the mesh for Case 2, yields non-zero density gradients in x -direction. This inevitably induces a spurious current at the interface. Hence, even though the maximum (ρ_{\max}) and the minimum (ρ_{\min}) density value at the dof-locations stay within the physical bounds, this ‘monotonicity’ is clearly not sufficient for maintaining hydrostatic equilibrium. Instead, a principal cause is that the ℓ^2 -projection introduces spurious density gradients when projecting a discontinuous particle field onto a polynomial space at the mesh.

Eliminating the interface issues for arbitrary locations of the interface with respect to the mesh is a study in itself, beyond the scope of this research. Neither do we resort to other *ad hoc* mitigating measures such as smearing of the sharp interface over multiple cells, or adding diffusion in a narrow band around the interface. It is believed that

Table 5.1: Hydrostatic balance: density gradient error and velocity error after one time step for the hydrostatic balance cases sketched in Fig. 5.2.

		$\epsilon_{\nabla_x \rho}$	$\epsilon_{\mathbf{u}}$	ρ_{\min}	ρ_{\max}
Case 1	FEM	7.5e-15	2.4e-11	1.	10.
	PDE	3.5e-8	5.7e-10	1.	10.
Case 2	FEM	7.5e-15	2.1e-11	1.	10.
	PDE	2.0e-1	1.8e-2	1.	10.

such measures only would obfuscate the formulations, instead of contributing to the main purpose of this chapter to illustrate the capabilities and challenges of the proposed particle-mesh scheme for multiphase flows. Anticipating the numerical examples in Section 5.5 involving air-water density jumps, the only stabilizing measure is to use an artificial value for the viscosity μ , which is assumed uniform throughout the domain. The resulting artificial dissipation mechanism primarily dissipates the velocity gradients in the fluid fraction with the highest relative increase in the kinematic viscosity, i.e. the air fraction. As a result, spurious currents - which mainly pertain to the lighter air fraction - are oppressed. When applicable, explicit notice is made of this artificial viscosity fix.

5.4. Implementation considerations

5.4.1. ℓ^2 -projection or conservative PDE projection?

A valid question at this point is whether or not to employ the local ℓ^2 - or the more involved PDE-constrained particle-mesh projection in the formulations for $\mathcal{P}_{E,I}$ and $\mathcal{P}_{E,II}$. The advantages of the former are clear, as it yields an efficient and easy-to-parallelize implementation, for which a bound-preserving density projection is readily available, see Section 3.5.2. These features render the ℓ^2 -projection an attractive option for practical applications, even though exact conservation properties are lost.

For the PDE-constrained projections, discrete conservation of mass and momentum is ensured, although this comes at the expense of solving global systems for the two particle-mesh projection operators. Moreover, monotonicity of the density field cannot be guaranteed as for the local ℓ^2 projections augmented with bound constraints. Therefore, over- and undershoot in the reconstructed density fields is mitigated by using the simple gradient penalization, to be introduced next.

From a scientific perspective, however, the conservative and more complicated PDE-constrained particle-mesh projections are undoubtedly much more interesting. The more so as particle-based methods have been somewhat suspected by mesh-based practitioners for the reason that these particle-based methods typically fall short in matching the accuracy and (local) conservation properties of state-of-the-art mesh-based schemes. The presented PDE-constrained projection entails the promise of bridging this gap, and overcomes some of the challenges in existing *surface tracking* schemes in that local conservation properties and maintaining sharp density jumps are inherent. Hence, focus will be on the PDE-constrained projections.

5.4.2. Density gradient penalization

Many interface capturing schemes rely on a way to smooth the sharp jumps in material properties, e.g. the density, over several mesh cells. In level-set methods, for instance, the interface reconstruction is typically done using a smoothed Heaviside function to ‘achieve numerical robustness’ [116]. With the ability of tracking sharp interfaces on particles as one of the strong assets of the proposed scheme, the aim is to stay away from such smeared interface approaches. The only measure for accommodating steep density gradients at the mesh, is by means of the gradient penalty term, governed by the parameter ζ in Eq. (5.18). The idea behind this penalty term is obvious: gradients inside elements are penalized, so that the density field is forced to become piecewise constant on cells, thereby alleviating over- or undershoot.

To make this penalty term mesh- and particle resolution independent, the choice of the parameter ζ needs further consideration. Inspecting the Lagrangian which underpins the PDE-constrained particle-mesh projection, Eq. (5.18), it is clear that the ℓ^2 -term is the primary agent by which over- and undershoot is introduced. An order of magnitude estimate for this term is given by the number of particles within a cell, i.e.

$$\sum_{\mathcal{S}_t^K} \rho_h(\mathbf{x}_p) w_h(\mathbf{x}_p) \propto |\mathcal{S}_t^K| \rho_h. \quad (5.33)$$

On the other hand, the gradient penalty term scales in two spatial dimensions as

$$\int_K \zeta \nabla \rho_h \cdot \nabla w_h \, d\Omega \propto \zeta \frac{\Delta \rho_h}{h_K^2} h_K^2, \quad (5.34)$$

which, upon the assumption $\Delta \rho_h \propto \rho$ shows that the ℓ^2 -term and the penalty term are of the same order of magnitude provided that

$$\zeta \propto |\mathcal{S}_t^K|. \quad (5.35)$$

The effectiveness of this term is further investigated for some numerical examples in Section 5.5. It nevertheless is evident that no strong claims on boundedness of the reconstructed density field can be made by adopting this heuristic approach. Deriving a physics-based formulation for ζ can use the work by Ten Eikelder as a starting point [94].

5.4.3. No-flux boundary condition

In practical applications, it usually is required to specify the boundary conditions at complex geometries. Strong imposition of no-slip condition is straightforward in such situations by requiring the velocity to be zero at such boundaries. However, when viscous effects near the walls can be ignored, the ‘no-flux’ condition needs to be imposed at the boundary. This boundary condition is a combination of the ‘no-penetration’ boundary [54], i.e. $\mathbf{u} \cdot \mathbf{n} = 0$, and free-slip, i.e. $(\hat{\boldsymbol{\sigma}}_d \mathbf{n}) \cdot \mathbf{t} = 0$, in the tangential direction(s) \mathbf{t} .

For conveniently handling complex boundary topologies, this no-flux condition is enforced via a penalty term on the global momentum equations in the Stokes step, by

modifying Eq. (5.22b) into

$$\sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_{d,h}^{n+1} \mathbf{n} \cdot \bar{\mathbf{w}}_h \, d\Gamma + \oint_{\Gamma_F} \epsilon \mathbf{n} \cdot (\bar{\mathbf{u}}_h^{n+1} \otimes \bar{\mathbf{w}}_h) \mathbf{n} \, d\Gamma = \oint_{\Gamma_N} \mathbf{h}_d^{n+1} \cdot \bar{\mathbf{w}}_h \, d\Gamma \quad \forall \bar{\mathbf{w}}_h \in \bar{\mathbf{W}}_{h,0}, \quad (5.36)$$

thus enforcing $\bar{\mathbf{u}}_h^{n+1} \cdot \mathbf{n} \rightarrow 0$ for a penalty parameter $\epsilon \gg 1$ on the boundary portion Γ_F , where $\Gamma_F \cap \Gamma_N = \emptyset$ the portion of the boundary on which a no-flux conditions is prescribed. The condition $(\hat{\boldsymbol{\sigma}}_d \mathbf{n}) \cdot \mathbf{t} = 0$ follows by substituting $\bar{\mathbf{w}}_h = \mathbf{t}$, with \mathbf{t} the unit tangent, and using the orthogonality of \mathbf{t} and \mathbf{n} , so that $\mathbf{n} \cdot (\bar{\mathbf{u}}_h^{n+1} \otimes \mathbf{t}) \mathbf{n} = 0$. In practice, the penalty parameter ϵ is set to the large value of 1e12. This term needs further scrutiny when iterative solution strategies for the unsteady Stokes system are used, as it might affect the conditioning of the unsteady Stokes system.

5.4.4. Particle insertion and deletion

To use the particle-mesh scheme on more complex geometries, an unstructured mesh can be used for the triangulation of the domain. Cells of different sizes typically pertain to such meshes, and controlling the amount of particles per cell therefore becomes inevitable. At first sight, this might contrast the claim from Chapter 4 that no *ad hoc* particle shifting algorithm is required. However, the particle control mechanism needed on unstructured meshes in our particle-mesh scheme, is a *practical* issue rather than a *fundamental* issue. More importantly, creating or deleting particles will not affect the conservation properties when using the PDE-constrained particle-mesh projections, since the particles only carry point samples of the underlying mesh fields, rather than mass and momentum. This feature contrasts with other particle methods such as SPH and MPM, in which particle insertion or deletion violates conservation claims. In Section D.2.4 of Appendix D the implementation of a particle initialization and deletion procedure is briefly outlined. For all the examples in Section 5.5, explicit notice will be made whether or not particle insertion/deletion is activated.

5.5. Numerical examples

A range of two-dimensional numerical examples is presented to demonstrate the potential of the scheme for practical multi-fluid and multi-phase problems. The first two examples consider problems with a relatively small density jump between the phases. The second half of this section demonstrates some examples in which the density jump spans several orders of magnitude, and specific emphasis is on problems relevant to civil engineering hydraulics. Finally, some laboratory experiments for free surface flows are reproduced numerically, and detailed comparisons are made. The examples in the second part thus demonstrate the applicability and/or the challenges related to simulating practical situations.

Concerning the particle-mesh projections for the density and the specific momentum, the focus is on the PDE-constrained approach. Typically, around 20 particles per cell are used and henceforth the penalty parameter ζ from Eq. 5.18 is set to 20, such that it approximately conforms the average number of particles per cell. Furthermore, $\mathbf{v}_h^{*,n} = \mathbf{u}_h^n$ in Eq. (5.21), see Eq. (5.30), whereas the particle specific momentum is

updated via Eq. (5.25) with $\theta_L = 1/2$. Except for the first example in this section (the rising bubble), surface tension forces are ignored.

5.5.1. Rising bubble

The rising of a low-density fluid bubble in a liquid column is a well-known benchmark for two-fluid numerical codes. Quantitative benchmark results for the two-dimensional problem were presented by Hysing et al. [119], and are used for validation purposes in this section. Apart from an accurate representation of the bubble in terms of shape, position and location, the quality of the reconstructed density field in terms of mass conservation and over- and undershoot is particularly relevant. For this test, the inclusion of surface tension forces is required, and the continuum surface force (CSF) discretization presented in Appendix C is used to this end.

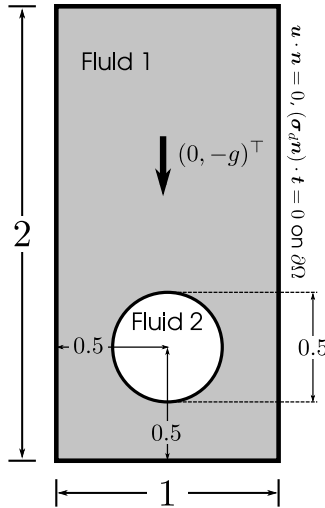


Figure 5.3: Rising bubble: initial configuration and boundary conditions.

The problem configuration is relatively simple: a circular, low-density bubble with radius $r = 0.25$, and initially centered at $\mathbf{x}_c = [0.5, 0.5]^T$, rises under the influence of buoyancy - induced by the gravity $\mathbf{g} = [0, -0.98]^T$ - in a surrounding heavier fluid. The two fluids occupy the domain $\Omega := [0, 1] \times [0, 2]$, and at the top and bottom of this container, no-flux conditions are imposed, i.e. $\mathbf{u} \cdot \mathbf{n} = 0$ and $(\boldsymbol{\sigma}_d \mathbf{n}) \cdot \mathbf{t} = 0$, with \mathbf{t} the unit tangent vector. A sketch of the benchmark setup is presented in Fig. 5.3.

The domain Ω is discretized into $160 \times 320 \times 2$ regular triangular elements, and approximately $2.4e6$ particles are randomly seeded in the domain so that the average number of particles per cell amounts to 24. No particles are added nor deleted, and the number of particles stays fixed over the course of the computation as a result. The time step is set to 0.02, which is approximately $h_K/4$, and thus relatively large compared to the $h_K/16$ time step used in [119] for comparable mesh resolutions. Furthermore, the considered time interval is $I \in (0, 3]$.

Two cases are studied, characterized by different density ρ_1/ρ_2 and viscosity ratios μ_1/μ_2 - with the subscripts 1 and 2 referring to the heavier and lighter fluid fractions, respectively - in addition to different surface tension coefficients, see Table 5.2. Using a characteristic length scale $L = 2r_0$, and velocity scale $U_g = \sqrt{2r_0g}$, the test cases are further classified by the Reynolds number and the Eotvös number,

$$Re = \frac{\rho_1 U_g L}{\mu_1}, \quad \text{and} \quad Eo = \frac{\rho_1 U_g^2 L}{\sigma},$$

where the latter indicates the ratio of the gravitational over the surface tension forces. The Reynolds number is equal for both cases, whereas the surface tension forces in Case 1 are much more dominant compared to Case 2.

Table 5.2: Rising bubble: overview of physical model parameters and dimensionless values after the benchmarks proposed in [119].

	ρ_1	ρ_2	μ_1	μ_2	g	σ	Re	Eo
Case 1	1000	100	10	1	0.98	24.5	35	10
Case 2	1000	1	10	0.1	0.98	1.96	35	125

The computed evolution of the bubble for the two cases is depicted in Fig. 5.4 at different time instants. A visual comparison is made with the available data from [119], showing excellent agreement both in shape and position of the bubble. Apart from demonstrating the feasibility to use Lagrangian particles for the tracking of density and momentum fields in a multiphase setting, this result also shows that the simple surface tension implementation from Section 5.4 performs well. For the surface tension dominated case (Case 1) the bubble attains an ellipsoidal shape, whereas for the more challenging Case 2, small secondary trailing bubbles are observed. Worth mentioning, is the shape of the bubble at $t = 3$ for Case 2 as shown in Fig. 5.4f. The simulated results suggest that two thin filaments connect the bulk of the bubble with the small secondary bubbles which broke off in an earlier stage. The different methods tested in [119] didn't reach consensus whether these thin strands should break or not. Our particle-mesh results suggest this breaking should not occur. This observation is supported by recently presented XFEM results in which strand breaking occurs for low-resolution computations, whereas thin strands pertain for high-resolution computations [111].

Fig. 5.5 compares the mean rising velocity for the two different cases to the values reported in literature, where the mean velocity is obtained by averaging the velocities of the light-fraction particles. Excellent agreement is obtained between the simulated velocities and the literature values in terms of maximum velocities, and the steady cruising speed attained around $t = 2$ in Case 1. Note also that the two local velocity maxima for the more complicated Case 2 are well captured by the particle-mesh scheme.

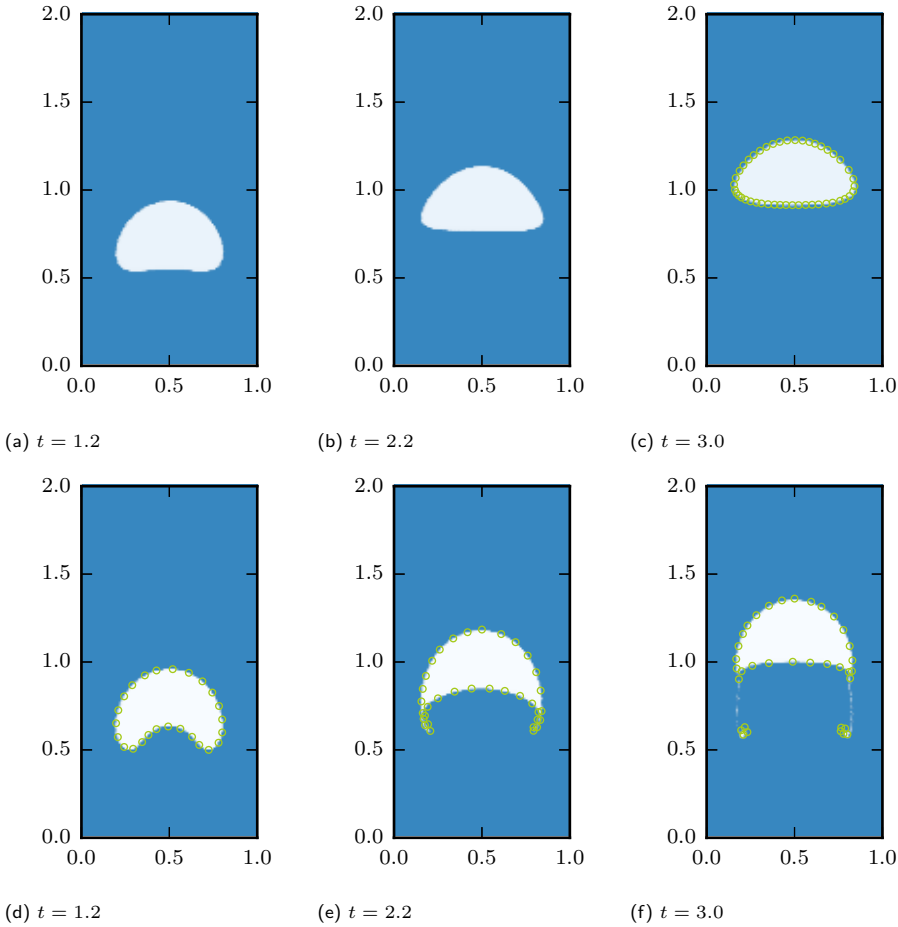


Figure 5.4: Rising bubble: time evolution at different snapshots in time for Case 1 (a-c) and Case 2 (d-f) compared with available data (green \circ) from [119].

Besides the shape, position and velocity of the bubble, the quality of the reconstructed density field in terms of over- and undershoot in the reconstructed density field is particularly good, and mass and momentum conservation over the particle-mesh projection steps are confirmed. Values for these quantities are listed in Table 5.3. The maximum dof value for the reconstructed density field over all time steps - indicated by $\rho_{h,\max}$ - amounts to 1001, thus closely matching the maximum physically admissible value of 1000. On a similar note, the minimum value, $\rho_{h,\min}$, is 92.8 for Case 1, which is slightly lower than the minimum physically admissible value of 100. The undershoot is more pronounced for Case 2, where the minimum density at the mesh over the course of the simulation amounts to -11.1. This deviates significantly from the physically admissible minimum value of 1, but does not affect the overall behavior. Table 5.3 confirms mass and momentum conservation over the PDE-constrained projections, in which the errors

over a time step are computed analogous to Eq. (3.49) and Eq. (4.53), using time levels $n + 1$ and n .

Table 5.3: Rising bubble: minimum ($\rho_{h,\min}$) and maximum ($\rho_{h,\max}$) value in reconstructed density field over the course of the computations. Also tabulated are the mass error ($\epsilon_{\Delta\rho}$) and the momentum error ($\epsilon_{\Delta(\rho u)}$) over particle-mesh projection at $t = 3$.

	$\epsilon_{\Delta\rho}$	$\epsilon_{\Delta(\rho u)}$	$\rho_{h,\min}$	$\rho_{h,\max}$
Case 1	-3.8e-15	-2.1e-13	92.8	1001
Case 2	-1.1e-15	6.1e-14	-11.1	1001

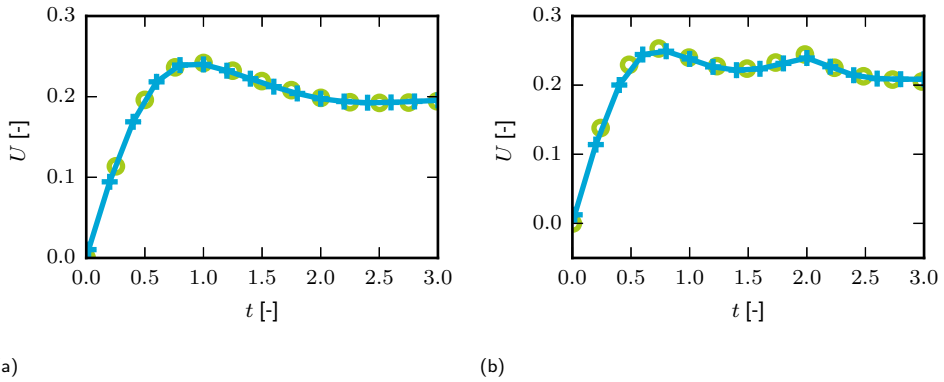


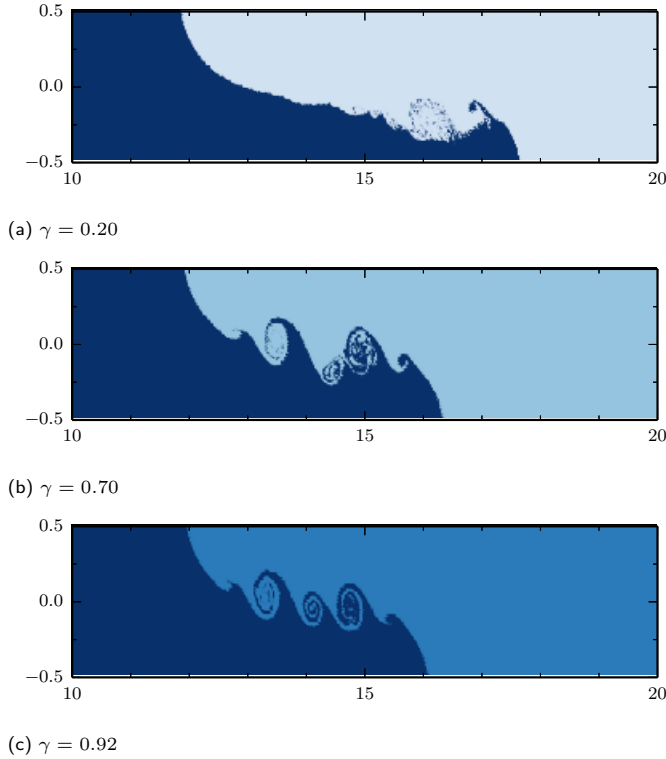
Figure 5.5: Rising bubble: center-of-gravity velocity ('+' marked solid line) for Case 1 (a) and Case 2 (b) compared with literature values (green 'o') from [119].

5.5.2. Lock exchange

The lock exchange problem is a classical model problem for the evolution of gravity currents induced by, for instance, gradients in salinity, temperature, or turbidity. The setup of the presented test is as follows: a rectangular channel with length L and height H is partitioned into two compartments, filled with fluids of different density and viscosity. These fluids are initially separated by a vertical wall, located at $x = L_1$. Under the influence of the gravity $\mathbf{g} = [0, -9.81]^\top$, the denser fluid propagates along the bottom of the channel, and the lighter fluid moves along the top of the channel after releasing the wall at time $t = 0$.

The lock exchange test serves a two-fold purpose. First of all, it demonstrates the conservation properties of the PDE-constrained projections for the mass and the momentum. In addition, it is used to investigate the performance of the scheme for a range of density ratios, with $\gamma := \rho_2/\rho_1 \in \{0.2, 0.26, 0.3, 0.4, 0.7, 0.92\}$ the density ratio, and ρ_1 the density of the heavier fluid. In general, the lower this ratio, the more challenging the problem is to simulate numerically. As such, the test serves as a stepping stone towards problems characterized by even lower density ratios, which will be considered in subsequent examples. Obtained results for the lock exchange test are compared with theoretical results presented in [120] and numerical results presented in its twin paper

[121].

Figure 5.6: Lock exchange: density profiles at dimensionless time $t^* = 4.0$.

The domain of interest is $\Omega := [0, 30] \times [-0.5, 0.5]$, and triangulated into $1200 \times 40 \times 2$ regular rectangular cells, on which $2.352e6$ particles are randomly seeded, resulting in 33 particles per cell on average. The number of particles is kept constant throughout the computation. No-flux conditions are imposed on the walls of the channel, using the penalty approach discussed in Section 5.4. A ‘virtual wall’, located at $x = 14$, initially separates the heavy fluid at the left, from the lighter fluid at the right. As in [121], the computations are run for a Reynolds number

$$Re = \frac{UH}{\nu} = 4000, \quad (5.37)$$

in which H the height of the channel, ν a kinematic viscosity which is set constant in the two fluids, and U a characteristic velocity based on the buoyancy velocity, i.e. $U = \sqrt{g'H}$, with the reduced gravity $g' = g(\rho_1 - \rho_2)/\rho_1 = g(1 - \gamma)$. To keep the Reynolds number at the same value of 4000, the kinematic viscosity is adapted for varying γ . Fluids are assumed immiscible, so that no diffusive mixing takes place through the interface, whereas in [121] a molecular diffusivity of $K = \nu$ is included in

the transport equations for the density field in view of numerical stability. The time is made dimensionless using a time scale $\sqrt{H/g'}$, and the non-dimensional end-time of the computations is set to

$$T^* = T \cdot \sqrt{g'/H} = 16. \quad (5.38)$$

The time step size equals $\Delta t^* = \Delta t \sqrt{H/g'} = 2.5e-2$, so that the total number of time steps amounts to 640. The total computational time for this setup is approximately 6200s using 14 Intel Xeon CPU E5-2690 v4 cores. A detailed analysis of the computational times for the lock exchange test on a high-resolution mesh, is found in Appendix D.4.2.

Particle fields at dimensionless time $t^* = 4.0$ are depicted in Fig. 5.6 around the location of the interface for $\gamma = (0.2, 0.7, 0.92)$. Paramount difference between the low-density ratio run $\gamma = 0.2$ and the high-density ratio runs $\gamma = (0.7, 0.92)$, is the emergence of Kelvin-Helmholtz instabilities at the interface for the high-density ratio cases, where these instabilities are less pronounced for the case with $\gamma = 0.2$ in which the buoyancy can be expected to be dominant over the viscous forces. Noteworthy is that the leftward moving front appears to be in approximately the same position at dimensionless time $t^* = 4$, thus indicating a constant dimensionless velocity of this front for the different density ratios.

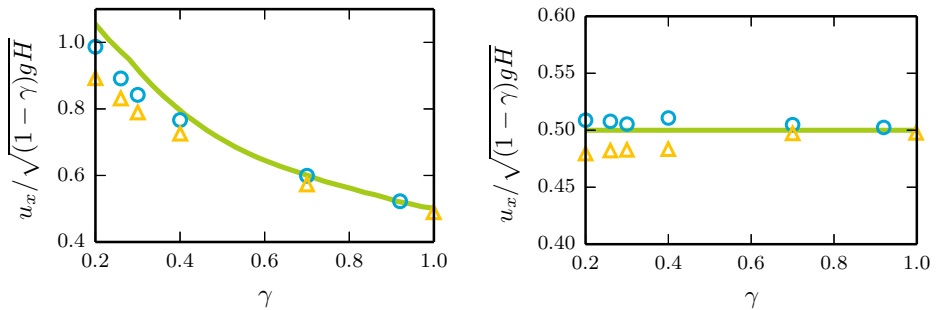


Figure 5.7: Lock exchange: dimensionless velocity of the heavy (left) and the light front (right), simulated values (o), numerical values from Birman et al. [121] (Δ) and theoretical values reported in Lowe et al. [120] (solid green line).

This is further illustrated by assessing the propagation velocities of the heavy and the light front, Fig. 5.7. Overall, the simulated results are in good agreement with the theoretical results from [120] and the numerical results from [121] for all values of γ . The simulated front positions are in excellent agreement with the theoretical values reported in [120], but slightly higher than the numerical results from [121]. Following the same explanation as in [120], it is conjectured that this discrepancy is due to diffusive mixing. This process is included in the results from [121], but absent in our immiscible simulations.

Finally, the conservation of mass and momentum to machine precision over the PDE-constrained particle mesh projections is demonstrated in Fig. 5.8, in which the conser-

vation error $\epsilon_{\Delta\rho}$ over the projection step is defined analogous to Eq. (3.49) as

$$\epsilon_{\Delta\rho} = \left| \frac{\int_{\Omega} (\rho_h^{n+1} - \rho_h^n) d\Omega}{\int_{\Omega} \rho_h^n d\Omega} \right|, \quad (5.39)$$

and the global momentum conservation error over the PDE-constrained momentum projection is given by

$$\epsilon_{\Delta(\rho\mathbf{u})} = \frac{\left| \int_{\Omega} (\rho_h^{n+1} \mathbf{v}_h^{n+1} - \rho_h^n \mathbf{v}_h^{*,n}) \cdot \mathbf{e}_x d\Omega \right| + \left| \int_{\Omega} (\rho_h^{n+1} \mathbf{v}_h^{n+1} - \rho_h^n \mathbf{v}_h^{*,n}) \cdot \mathbf{e}_y d\Omega \right|}{\left| \int_{\Omega} (\rho_h^n \mathbf{v}_h^{*,n} \cdot \mathbf{e}_x + \rho_h^n \mathbf{v}_h^{*,n} \cdot \mathbf{e}_y) d\Omega \right|}, \quad (5.40)$$

with \mathbf{e}_x and \mathbf{e}_y the unit vectors in the Cartesian x and y -direction, respectively.

The time series for these errors are depicted in Fig. 5.8, demonstrating that mass and momentum conservation are satisfied to machine precision.

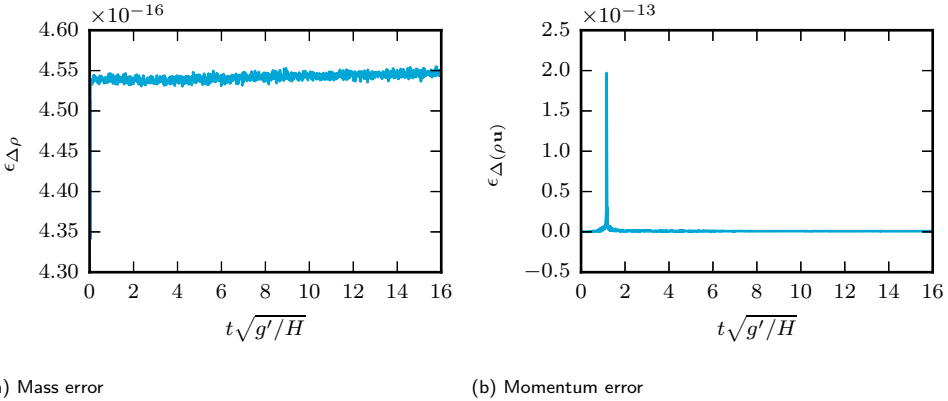


Figure 5.8: Lock exchange: mass error (Eq. (5.39)) and momentum error ((Eq. (5.40))) as a function of the dimensionless time.

5.5.3. Hydrostatic balance

From this example onward, examples involving large density jumps are considered. This is relevant for, e.g., problems involving air-water interfaces, where the ratio between the air phase and the water phase amounts to an order of 1000. In combination with the low physical viscosities, such problems typically pose challenges to numerical schemes, and the particle-mesh scheme forms no exception as illustrated in Section 5.4

To assess the influence of the viscosity on maintaining hydrostatic equilibrium, the hydrostatic balance is revisited. Though seemingly trivial, it is a particularly demanding test case in the numerical realms, as it requires a numerical scheme to accurately capture the subtle balance between gravity and the pressure gradient.

Consider the domain $\Omega := [0, 1] \times [0, 2]$ where the lower half is filled with fluid, having the density of water, i.e. $\rho_1 = 1000 \text{ kg.m}^{-3}$. The upper half is filled with a liquid having a density of $\rho_2 = 1 \text{ kg.m}^{-3}$, matching the air density. The top of the domain is assumed open by imposing a zero traction condition on this part of the boundary, and on the other boundaries the no-flux condition is imposed. The domain is partitioned into $10 \times 21 \times 2$ cells, so that mesh facets and liquid interface do not conform. On the domain, 11,250 particles are seeded. The time step is set to $\Delta t = 2.5e-2 \text{ s}$, and the time interval of interest is $I \in (0, 20] \text{ s}$. Different viscosities and/or density projections are considered, resulting in the four test cases listed in Table 5.4. The viscosity in Case 1 is close to the true dynamic viscosities for air and water, whereas in all other cases the dynamic viscosity is assumed equal in water and air, and larger than the physical values.

Table 5.4: Hydrostatic balance: overview of the different test cases by varying the viscosity and/or the density projection.

	Density projection	μ [kg.m ⁻¹ .s ⁻¹]	ν [m ² .s ⁻¹]
Case 1	PDE	$\rho_h \nu$	$1e-6$
Case 2	PDE	1	-
Case 3	PDE	5	-
Case 4	ℓ^2	5	-

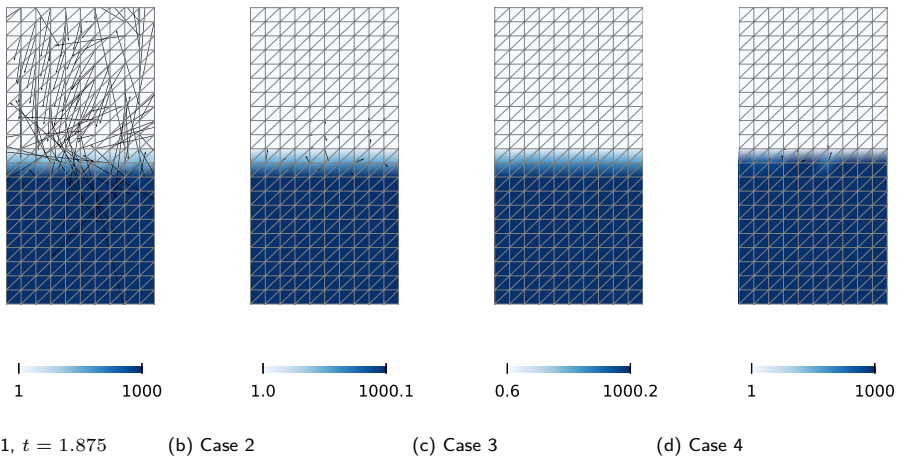


Figure 5.9: Hydrostatic balance: testing hydrostatic equilibrium for different projections and viscosities at $t = 20$ or at time of failure. For description of cases, refer to Table 5.4.

Results for the different cases are visualized in Fig. 5.9. Case 1 fails already in an early stage as a result of spurious velocities, originating from the interface and migrating into the lighter air fraction, since the local imbalance in the pressure gradient term has much more effect on the accelerations in the lighter fluid fraction. Increasing the dynamic viscosity artificially, mitigates this issue and spurious velocities are hardly visible (Case 2)

or virtually absent (Case 3 and 4) after 800 time steps.

Increasing the viscosity artificially is a fix, rather than a fundamental solution to the problem. The interface issue therefore deserves scrutiny in future research, where this open challenge is related to an incomplete discretization of the governing physical processes. Arguably, any numerical scheme which not provably obeys the underlying physics, should reveal its shortcomings in the returned results one way or the other.

The interface issue plays a role in all the examples presented in the sequel of this chapter, most prominently so for the examples in which the dynamics are the result of small deviations from the hydrostatic balance, i.e. the standing wave test in Section 5.5.4 and the propagating wave train in Section 5.5.5. As a result, the artificial dynamic viscosity is explicitly reported and chosen comparatively high in these examples. The interface artifacts are expected to be less pronounced for the dam break test Section 5.5.6 and the breaking of a solitary wave on a slope Section 5.5.7, both characterized by violent interface motion that overrules the delicate hydrostatic balance.

5.5.4. Standing wave

To assess whether a high dynamic viscosity, needed in order to alleviate interface artifacts, does not oppress the relevant dynamics altogether, a standing wave in a square box $\Omega := [0, 8]^2$ is considered next. The still water depth d is 4 m, and the initial configuration of the air-water interface is given by

$$\eta(x) = A \cos \frac{2\pi x}{L}, \quad (5.41)$$

with the amplitude $A = 0.05 d = 0.2$ m, and the wave length $L = 8$ m so that the wave length equals the width of the box. No-flux conditions are imposed at the boundaries, and the initial velocity is set to zero everywhere. The square box is triangulated into $80 \times 80 \times 2$ regular triangular cells on which 320,000 particles are randomly seeded, so that the average number of particles amounts to 25 particles per cell. The time step is set to $\Delta t = 1e-2$ s, and $I \in (0, 20]$ s is the time interval of interest. Furthermore, the body force is set to $\mathbf{F}_G = [0, -g]^\top$, where $g = 9.81$ m.s⁻² is the gravitational acceleration. The density in the top-half of the domain is set to the air density, i.e. $\rho_2 = 1$ kg.m⁻³, and in the lower-half to the water density, i.e. $\rho_1 = 1000$ kg.m⁻³. The dynamic viscosity is set to $\mu = 5.0$ kg.m⁻¹.s⁻¹. for both fluids.

The particle positions are visualized in Fig. 5.10 at different time instants, chosen such that they correspond to the maximum (Fig. 5.10b) and minimum (Fig. 5.10c) surface elevation at $x = 4$. Qualitatively, the air-water interface is well-maintained over the course of the computation. This observation is further supported by comparing the time series of the simulated elevations at the center of the tank, with theoretical values. Since the amplitude-depth ratio amounts to 0.05, non-linear effects are anticipated non-negligible, and a comparison is made with the second-order solution from Wu & Taylor [122], which reads

$$\eta(t) = -A \cos \omega_2 t + \frac{A^2}{8g\omega_2^2} (2\omega_2^4 \cos 2\omega_2 t + (k_2^2 g^2 + \omega_2^4 - (k_2^2 g^2 + 3\omega_2^4) \cos \omega_4 t)) \quad (5.42)$$

in which the wavenumber k_m and the eigen frequency ω_m are given by:

$$k_m = \frac{m\pi}{L}, \quad \text{and} \quad \omega_m = \sqrt{k_m g \tanh k_m d}.$$

Simulated and theoretical wave elevations at the center of the box are presented in Fig. 5.11, showing an excellent match between theoretical and simulated results in terms of amplitudes as well as phases. Second-order effects are clearly reflected in the simulated results, leading to, for instance, a high wave crest around $t = 10$ s and a deep wave trough around $t = 16$ s.

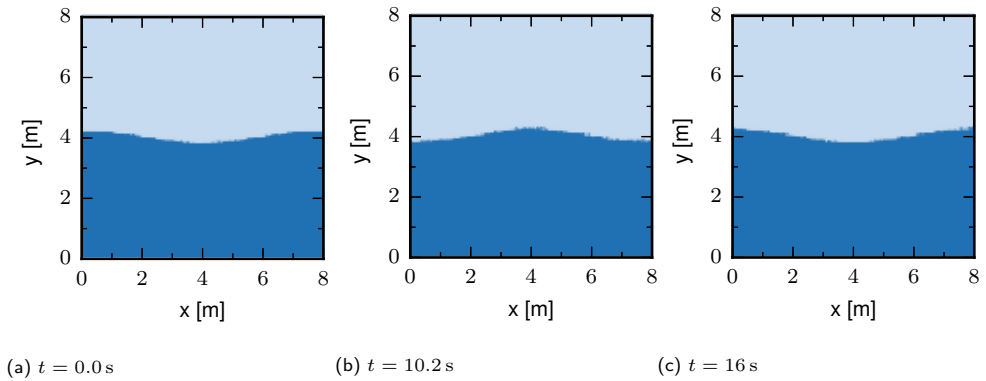


Figure 5.10: Standing wave: particles with associated density at different time instants.

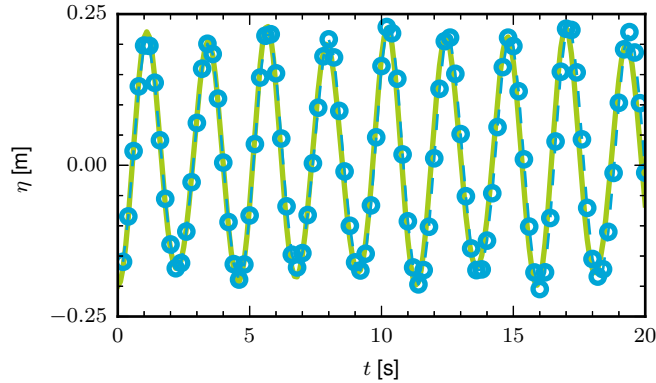


Figure 5.11: Standing wave: simulated results (dashed line with \circ markers) compared with second-order theoretical solution from [122] (solid line).

Results for this test show that the uniform dynamic viscosity $\mu = 5.0 \text{ kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$, introduced to alleviate spurious interface currents, has little impact on the bulk motion of the standing wave for the simulated time interval. This is explained by the notion that

the kinematic viscosity ν , which is frequently called the *momentum diffusivity*, remains relatively small in the water phase, i.e. $5 \cdot 10^{-3} \text{ m}^2 \cdot \text{s}^{-1}$, hence not deteriorating the bulk motion. In the air phase, however, the momentum diffusivity ν is 1000 times larger, thereby effectively oppressing spurious velocity currents in the air phase.

5.5.5. Wave train

The ability of the scheme to generate a regular progressive wave train in a numerical wave tank is assessed next. This feature is essential to render the scheme applicable to, e.g., reproduce wave experiments carried out in laboratory wave flumes. Our ‘numerical wave flume’ initially comprises the rectangular domain $\Omega_0 = [0, 40] \times [0, 4] \text{ m}$, in which the still water depth d equals 2.5 m. To pave the way for more complex geometries, the domain of interest is partitioned using i) a regular triangular mesh consisting of $800 \times 160 \times 2$ cells, and ii) a fully unstructured mesh constructed in GMSH [123], in which the mesh is aligned with the still water level at $d = 2.5 \text{ m}$. On both meshes, the number of particles is controlled on the fly by keeping the number of particles in a cell between 20 and 35 particles. The time interval of interest is $I \in (0, 30] \text{ s}$. As in the preceding two examples the dynamic viscosity is set to $\mu = 5.0 \text{ kg} \cdot \text{m}^{-1} \cdot \text{s}^{-1}$. Details of the distinguishing features of the two setups, including the runtime on 14 Intel Xeon CPU E5-2690 v4 cores are listed in Table 5.5.

Table 5.5: Wave train: properties of the structured and unstructured mesh.

Triangulation	# cells	# particles ($t = 0$)	Δt [s]	Computational time [h]
Structured	256,000	5.51e6	1.25e-2	17.5
Unstructured	85,994	2.9e6	1e-2	8.0

The boundary at the top of the numerical flume is open, and at the horizontal bottom and the vertical wall at the right the normal component of the velocity $\mathbf{u} \cdot \mathbf{n}$ is set to zero. In order to generate a regular wave train, the normal velocity $\mathbf{u} \cdot \mathbf{n} = u_x$ at the left vertical boundary is prescribed using the linear wavemaker theory for a piston wave generator [124], i.e.

$$u_x(t) = \frac{H}{2C} \omega \cos \omega t, \quad (5.43)$$

where a target wave frequency ω of $3.83 \text{ rad} \cdot \text{s}^{-1}$ and a target wave height H of 0.2 m will be used, furthermore,

$$C = \frac{2(\cosh 2kd - 1)}{\sinh 2kd + 2kd}, \quad (5.44)$$

with d the still water depth of 2.5 m, and k the wave number determined from the dispersion relation $k = \omega^2 / (g \tanh(kd))$. With these parameters, the boundary motion should generate a monochromatic, linear wave train.

As a result of the piston-type wave generation at the right boundary, the mesh deforms with velocity

$$\hat{U}_h = \begin{cases} \frac{u_x}{x_L - x_R} (x - x_R) & x_L \leq x \leq x_R \\ 0 & \text{elsewhere} \end{cases} \quad (5.45)$$

in which x_L the location of the piston, and $x_R > x_L$ a location in the domain, typically much smaller than the length of the domain of interest so that the mesh motion is restricted to a part of the domain. In this case, x_R is set to 5 m.

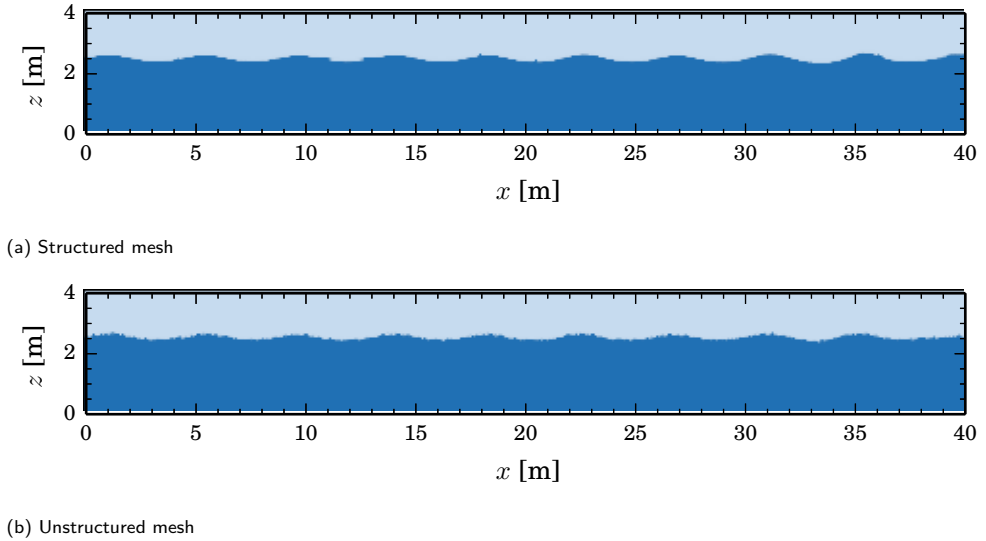
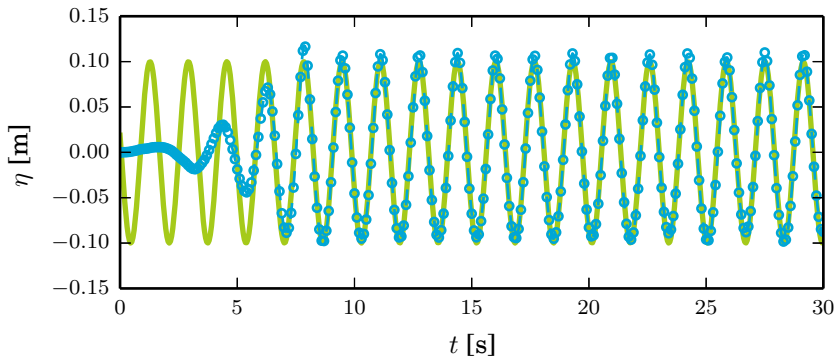


Figure 5.12: Wave train: particle density field at $t = 30$ s. For specification of underlying mesh and particle field, see Table 5.5.

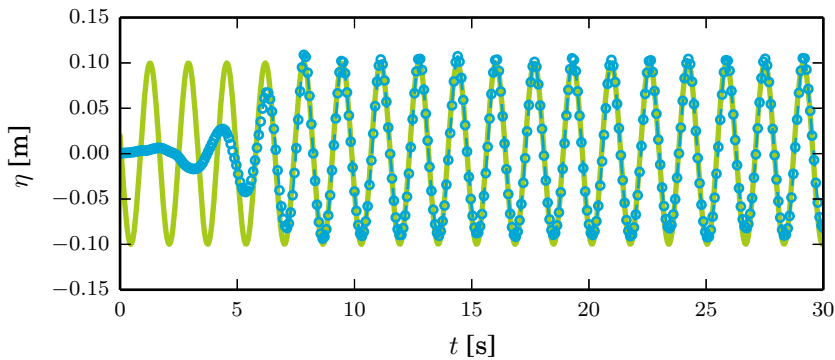
The simulated particle fields with their corresponding density are plotted at $t = 30$ s in Fig. 5.12. Overall, the results for the structured and the unstructured mesh exhibit good correspondence, and at first sight the locations of the wave troughs and wave crests appear to coincide. Careful comparison of the two cases reveals that the location of the free surface is somewhat less sharp for the unstructured mesh, compared to the structured mesh. This is attributed to the lower mesh and particle resolution for this case, see also Table 5.5. Not visible in these plots is that at $t = 30$ s, the piston wave boundary is located at $x_L = 0.05$ m.

Numerical predictions for the wave elevation η at a distance of $3d = 7.5$ m from the wave maker are depicted in Fig. 5.13. In terms of the phases, an excellent match between simulated and theoretical results is observed. In terms of the amplitude, good agreement is observed between high-resolution results obtained on the structured mesh and the theoretical results, Fig. 5.12a. On the lower-resolution unstructured mesh, the agreement between the expected and the simulated amplitudes is also reasonable, although the wave troughs are slightly under-predicted. Overall, the obtained results indicate that wave generation via a moving boundary is a viable route in the developed particle-mesh scheme.

Although not further shown here, mass and momentum conservation over the PDE-constrained particle-mesh projections are confirmed. This of course requires to account for the mass and momentum flux through the open boundary at the top of the domain.



(a) Structured mesh



(b) Unstructured mesh

Figure 5.13: Wave train: simulated results (dashed line with \circ markers) and theoretical (solid green line) water level elevations as a function of time at $x = 7.5$ m.

5.5.6. Dambreak

In the previous two examples, the dynamics were governed by relatively small deviations from hydrostatic equilibrium. Such cases, in which the (air-water) interface typically remains of a rather simple topological shape, were shown to be challenging for the particle-mesh scheme, and purely mesh-based methods can probably compete with the particle-based methods for such problems by, e.g., keeping the mesh aligned with the interface, see e.g. [15]. In the next two examples, situations are considered in which violent dynamics cause the interface to attain complex shapes, including breaking and reconnection. For such cases, advantages of using Lagrangian particles are anticipated in that they can track the complex-shaped interface. Furthermore, for such dynamic situations, one is typically interested in the forces exerted on, e.g., a coastal structure. In this and the upcoming example, special attention is therefore paid to the simulated impact pressures and forces, by comparing against physical laboratory experiments documented in the literature. Compared to the preceding examples, these two numerical examples are undoubtedly the most relevant for anticipated applications of the developed

particle-mesh scheme.

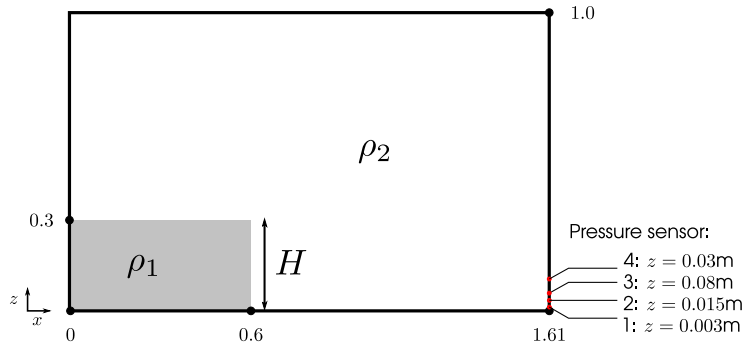


Figure 5.14: Dambreak: setup of geometry and locations of pressure probes, with the lengths given in meters.

The dambreak problem, a classical benchmark for free surface flow solvers, is considered as a first such example. Under the influence of gravity, a column of water collapses following an abrupt release of a gate. This test is frequently used to validate the propagation of the surge front against available experimental data, and to illustrate that a numerical scheme qualitatively captures the complex interfaces when the front collapses onto and reflects from the opposing wall. The test presented here aims to go one step further, and in addition verifies whether the computed impact pressures on this wall are accurately simulated.

To this end, a comparison with the detailed experimental results reported by Lobovsky et al. [125] is made. The domain of interest is a closed box with base length $L = 1.61$ m and total height $D = 1.0$ m. The column of water, with density $\rho_1 = 1000 \text{ kg.m}^{-3}$ initially occupies the sub-domain $\Omega_W := [0, 0.6] \times [0, H]$ m, where $H = 0.3$ the height of the column. The remaining part of the domain, $\Omega \setminus \Omega_W$, is filled with air of density $\rho_2 = 1 \text{ kg.m}^{-3}$. The viscosity is set to a constant value of $0.05 \text{ kg.m}^{-1}.\text{s}^{-1}$ throughout the entire domain. This is only a factor 50 larger than the molecular viscosity of water, compared to the factor 5000 larger dynamic viscosity in the foregoing standing and progressive wave test. By doing so, we anticipate the dynamic, i.e. convection dominated, character for the dambreak test for which interface artifacts are expected to be less pronounced compared to the foregoing tests, thus admitting lower viscosity values. As in the experimental setting, the pressure is probed at different locations. Fig. 5.14 presents an overview of the numerical setup and the locations of the pressure probes.

The computational domain is partitioned into $161 \times 100 \times 2$ regular triangular elements, and 894,000 particles are randomly seeded on the domain, yielding approximately 28 particles per cell. The number of particles per cell is kept constant throughout the computation. At the boundaries, no-flux conditions are specified. The time step is set to a fixed value of $\Delta t = 1e-3$ s, and the considered time interval is $I \in (0, 1.4]$ s. With expected peak velocities in the range of 1 m.s^{-1} to 10 m.s^{-1} during the impact event, the CFL number is in the range $U\Delta t/h_K \approx 0.1 - 1.0$. It is noted that this fixed time

step size will be rather conservative for large parts of the considered time interval, and a dynamic time step size certainly deserves further investigation when optimizing the scheme in terms of computational runtime. This is, however, not further investigated in this research.

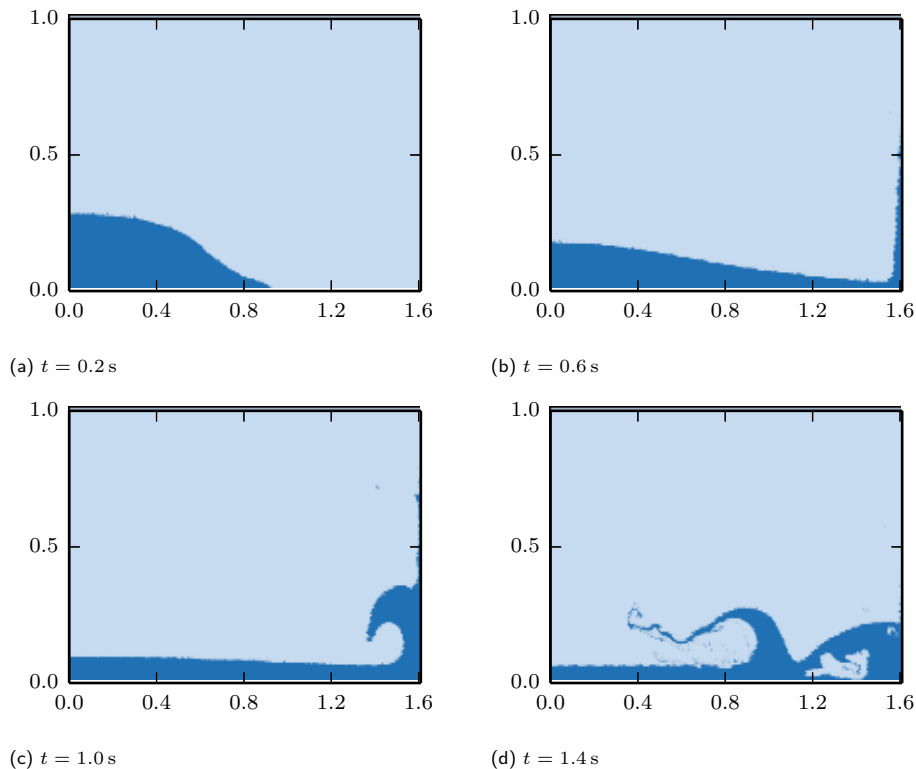


Figure 5.15: Dambreak: air-water interface for the dambreak flow at different time instants.

Particle fields at different time instants are depicted in Fig. 5.15, displaying an advancing smooth wave front at $t = 0.2 \text{ s}$. This wave front collapses onto the opposing vertical wall, leading to a strong upward directed jet in which observed velocities are of the order 5 m.s^{-1} . Under the influence of gravity, the jet falls back and forms a plunging breaker type, Fig. 5.15c. During this process, a pocket of air is entrapped, Fig. 5.15d. This pocket is not just a void, as would be the case in a single-phase approach.

The position of the advancing wave front is depicted in Fig. 5.16. Overall, a good correspondence between simulated and experimental results is obtained. At later time instants the position of the wave front slightly overestimates the experimental results. A possible explanation for this behavior is that friction and surface tension effects slow down the propagating front in the experimental setting, whereas these effects are not accounted for in the numerical experiment.

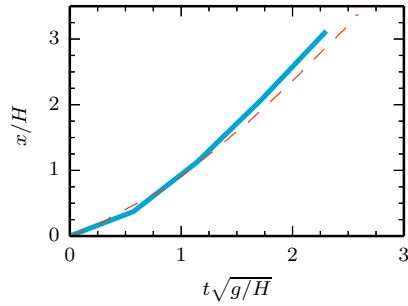


Figure 5.16: Dambreak: simulated position of the heavy density front (solid line) compared with the experimental values from Lobovsky et al. [125] (dashed line).

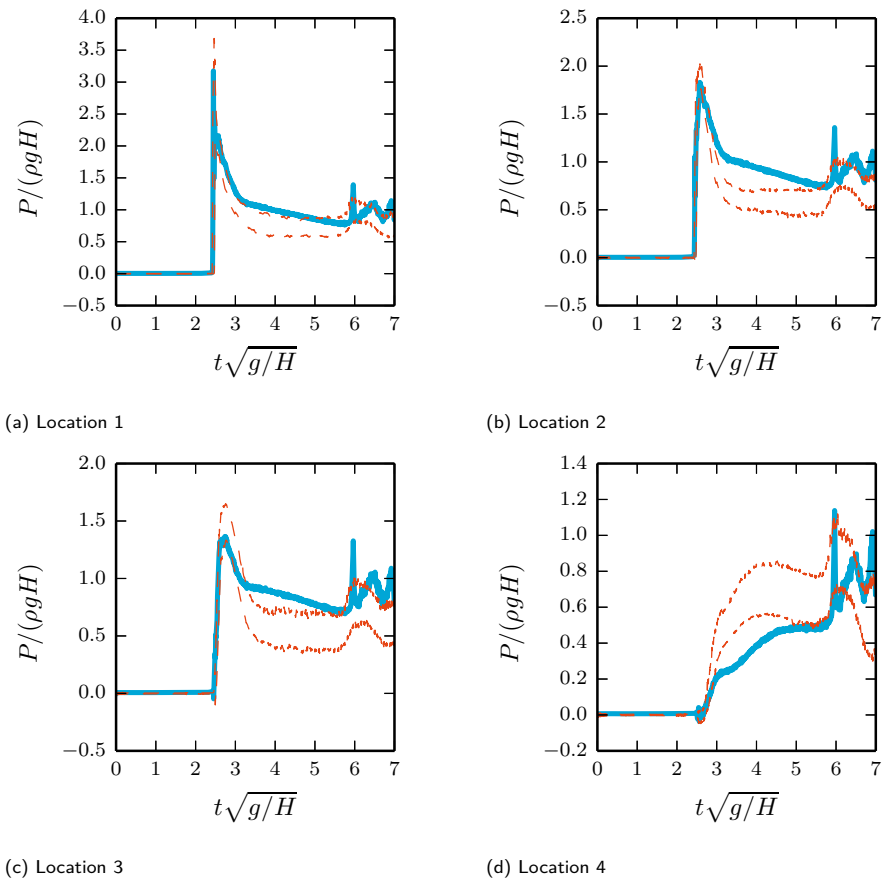


Figure 5.17: Dambreak: impact pressures at the vertical plate, numerical results (solid line), and 2.5 and 97.5 percentiles from [125] (dashed lines). See Fig. 5.14 for locations of pressure probes.

Simulated and experimental pressures at the 4 pressure probe locations are compared in Fig. 5.17. Overall, the simulated pressures stay within 2.5 and the 97.5 percentiles of the experimental results from [125]. In particular, the first impact at the probe locations is accurately predicted in the simulations, both in terms of timing and magnitude. Noteworthy to mention is that also small details, observed in the experiments, are captured in the numerical simulations, including the pressure troughs just preceding the moment of impact at probe locations 3 and 4, Fig. 5.17c and Fig. 5.17d. Also, the moment and magnitude of the simulated pressures during the formation of the plunging wave around $t\sqrt{g/H} = 6.0$ is in relatively good agreement with the experimental values. The largest differences are observed in between the dimensionless times $3.5 \leq t\sqrt{g/H} \leq 6.0$, particularly so for probe locations 2 and 3 (simulation overestimates the pressures from the laboratory experiment), and location 4 (simulation underestimates the laboratory values). It is hard to conjecture about the underlying reasons, but a possible explanation on the numerical side can be that the pressure field - which is piecewise constant for $k = 1$ by virtue of the function space definition Eq. (5.10) - has a relatively coarse resolution (approximately 1 cm) compared to the length scale at which the dynamics occur. In the experimental setting, for instance, the pressure transducers have a sensing diameter of 0.42 cm only. On the experimental side, Lobovsky et al. noticed - without explanation - that their pressure values were lower than obtained in experiments conducted by other researchers, particularly so for location 2 (see Fig. 31 in [125]). Despite this discrepancy, it can be concluded that overall, the developed particle-mesh scheme accurately captures the impact pressures for this very dynamic test case.

5.5.7. Breaking solitary wave

As a last example, the laboratory experiments from [126] are considered. In this test, the impact of a solitary wave onto a coastal structure is investigated for three different still water levels. Fig. 5.18 presents a schematic of the experimental setup. The bottom topography is characterized by a 1:20 slope, starting at 7 m from the wave maker as in the numerical configuration, presented parallel to the experiments in [126]. On this slope, a trapezoidal caisson-like structure with seaward 1:4 and landward 1:1.8 slopes is situated. Of particular interest for this experiment, is the wave force exerted on the seaward and landward facing slopes of this caisson structure. In the experimental configuration, 12 pressure transducers are placed on the slopes from which the wave forces can be derived. Furthermore, a number of wave gauges were placed in the laboratory flume to investigate the wave propagation process. ‘Numerical gauges’ are placed at a number of corresponding locations listed in Table 5.6, where the location labels match those reported in [126].

Table 5.6: Breaking solitary wave: wave gauge labels and locations.

Label	g1	g10	g15	g22	g28	g39	g40	g46
x -position [m]	5.9	9.644	10	10.46	10.732	11.045	11.12	11.47

Three solitary wave types are considered, differing in the wave amplitude over undisturbed depth ratio $\epsilon = H_0/h_0$, see Table 5.7. A piston wave maker is used to generate

the distinct waves, a full account of the wavemaker motion can be found in, e.g. [29] which uses the theory by Goring [127].

Table 5.7: Breaking solitary wave: wave conditions according to [126] and mesh characteristics for the numerical experiments.

	Wave			Mesh		
	h_0 [m]	H_0 [m]	$\epsilon = \frac{H_0}{h_0}$	Cells	$h_{K,\min}$ [m]	$h_{K,\max}$ [m]
Type 1	0.2	0.07	0.35	115,042	2.0e-3	2.4e-2
Type 2	0.22	0.0638	0.29	119,456	2.8e-3	2.4e-2
Type 3	0.256	0.0589	0.23	127,533	1.9e-3	2.6e-2

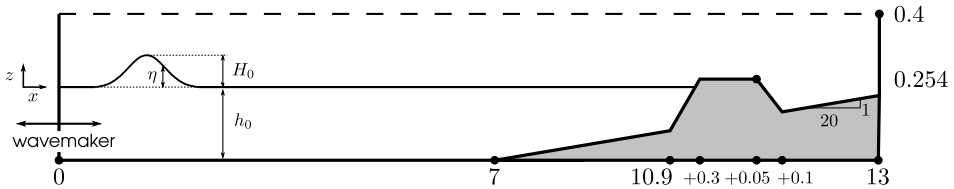
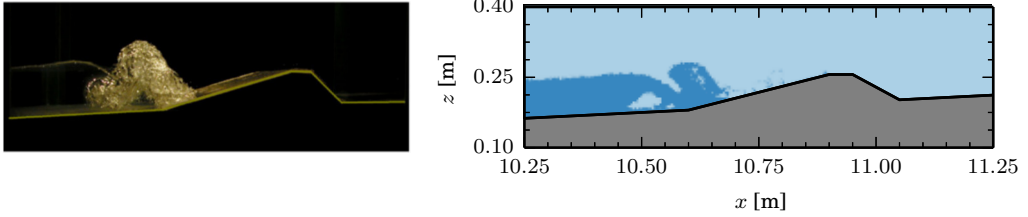


Figure 5.18: Breaking solitary wave: setup from [126], length is in meters.

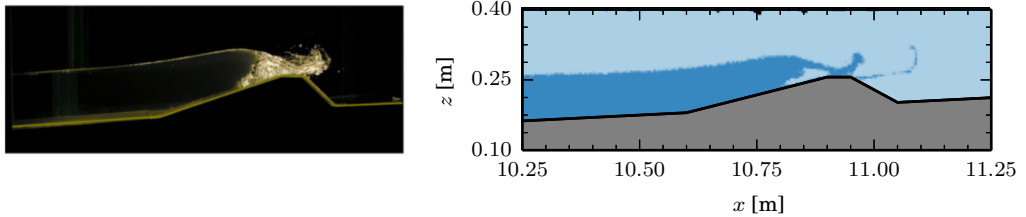
Given this setup, the purpose of this numerical test is three-fold: i) to illustrate the performance of the scheme in an applied, coastal engineering setting, and more specifically, the test case is used ii) to assess the propagation of the wave generated at the boundary up to and including the wave breaking and wave run-up stages, and iii) to compare the computed forces on the coastal structure with corresponding experimental results. The latter quantity is believed to be particularly valuable from a practical perspective. However, wave forces are not only challenging to measure in a laboratory setting, but also difficult to simulate numerically, since accurately simulating the wave propagation and the breaking process are prerequisite for reliable results. On top of that, any inconsistencies or inaccuracies in the simulated pressure field would immediately deteriorate the simulated impact forces.

One of the advantages of the developed particle-mesh scheme is that it allows to follow complex bottom topographies by partitioning the domain in an unstructured simplicial mesh. This feature owes to the underlying FEM strategy, and is exploited in this numerical experiment. For each wave type, a different mesh is generated so that the still water level h_0 is aligned with the mesh. The total numbers of cells and the minimum and maximum cell sizes $h_{K,\min}$ and $h_{K,\max}$ are reported in Table 5.7. Particles are inserted and deleted every time step, so as to maintain in between 20 and 35 particles per cell, resulting in approximately 3.5-4.0e6 particles in total. The specific momentum assigned to newly created particles is found by evaluating the underlying mesh field at the particle location. The binary-valued density carried by the particle is determined using Eq. (D.2) from Appendix D, so as to assign the particle either the air or the water density. It is emphasized that annihilation and insertion of particles leaves the conservation proofs unaffected, since particles do not carry integral quantities. The

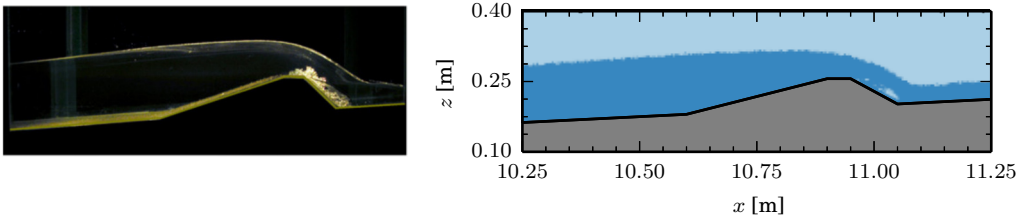
dynamic viscosity is set to $\mu = 0.1 \text{ kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$. Furthermore, a fixed time step of $\Delta t = 2.5\text{e-}3 \text{ s}$ is used to partition the time interval of interest $I \in (0, 12] \text{ s}$.



(a) Type 1, snapshot at $t = 7.61 \text{ s}$ from experiment (left) and from simulation at $t = 7.5 \text{ s}$ (right).



(b) Type 2, snapshot at $t = 7.84 \text{ s}$ from experiment (left) and from simulation at $t = 7.75 \text{ s}$ (right).



(c) Type 3, snapshot at $t = 8.00 \text{ s}$ from experiment (left) and from simulation at $t = 8.00 \text{ s}$ (right).

Figure 5.19: Breaking solitary wave: Left column pictures: from the experiment by Hsiao and Lin [126]; Right column: the simulated density field on the particles around the same time instant.

Given this time step and the mesh size as reported in Table 5.7, the CFL number is around 1 for an expected order of magnitude of the velocity of $1 \text{ m}\cdot\text{s}^{-1}$. Around the wave impact event, velocities of about $5 \text{ m}\cdot\text{s}^{-1}$ are observed for the Type 3 wave, implying a CFL-number of approximately 5 for the small cells around the coastal structure, thus indicating a rather mild CFL-requirement. Similar to the dambreak problem, it would be beneficial in view of the total run time to use a dynamic time step size. This is

not further pursued, and the total runtime for the computations is approximately 17.5 hours, using 14 Intel Xeon CPU E5-2690 v4 cores.

Fig. 5.19 shows snapshots from the experiment [126] as well as the simulated density field at the particle level around the same time instant. In the visual norm, the simulated results bear striking resemblance with the experimental results. For the Type 1 soliton, a plunging wave breaks already at the mild-sloping foreshore. During this process an air pocket is entrapped, and the overtopping jet impinges on the still water. This, in turn, causes the typical splash-up in front of the breaking wave, with this feature clearly visible in the simulation results, Fig. 5.19a. For the Type 2 wave, the wave collapses upon the coastal structure as a plunging wave, enclosing a significant amount of air. The overtopping wave generates a violent jet, reflecting on the crown of the coastal structure. Comparing to the experimental results, this process seems to be well-captured in the numerical setting, see Fig. 5.19b. The most striking result for the Type 3 wave is the occurrence of an entrapped air pocket on the leeward slope of the coastal structure. This air pocket is also clearly observed in the numerical results, Fig. 5.19c.

For a quantitative assessment of the quality of the simulations, the simulated and measured free surface elevations at the different wave gauges are compared in Fig. 5.20. Overall, experimental and simulated wave elevations exhibit an excellent agreement. Even small length scale features - such as the trailing wave at gauge location $g1$ - as well as short time scale features - such as the elevations at the wave gauges $g(10,15,22,28,29)$ where wave breaking occurs - are captured accurately in the simulations in terms of time and magnitude. Also for the wave gauges placed on the leeward slope - $g(39,40,46)$ - experimental and simulated results are in good correspondence. The largest difference is observed for the Type 1 wave at gauge $g40$. This owes to the fact that the amount of overtopping water will be smaller compared to the other two types, hence leading to only a thin layer of water on the leeward facing slope for Type 1.

The dynamic wave force on the structure is obtained in the numerical setting by integrating the dynamic pressures along the seaward and the leeward slope. Noteworthy is that no *heuristic* pressure filtering is required as in the numerical results presented in [126]. Results for the three wave types are shown in Fig. 5.21. Again, excellent agreement is obtained between experimental and numerical results¹. The timing of impact and the magnitude of the impact are excellently captured by the particle-mesh scheme. This includes subtle features such as the local maximum in the vertical impact force on the seaward slope for the Type 1 wave around $t = 7.5$ s, Fig. 5.21a. Compared to the numerical results presented in [126], the obtained pressures with the particle-mesh scheme arguably match the experimental results better on the leeward slope, particularly so for the Type 2 wave, see Fig. 5.21d. Overall, the results presented in this section clearly reveal the potential of the developed particle-mesh scheme to simulate violent wave breaking processes, where one is typically interested in accurate wave-induced forces exerted on coastal structures.

¹Fig. 10 in [126] mistakenly interchanges the horizontal and the vertical component force, which is corrected in Fig. 5.21.

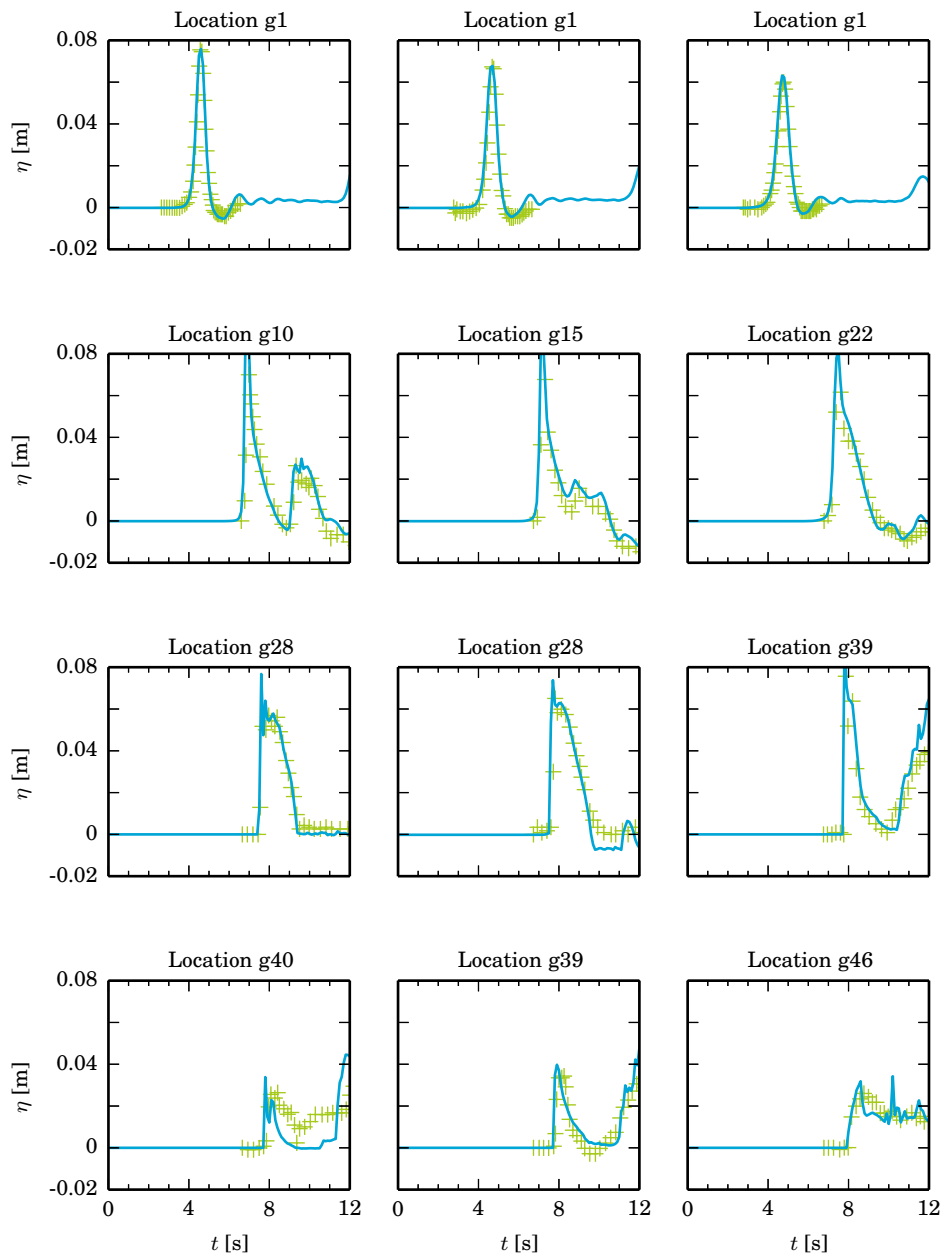


Figure 5.20: Breaking solitary wave: time history of surface elevations at different locations for Type 1 wave (left column), Type 2 wave (middle) and Type 3 wave (right column). For specification of the location labels, see Table 5.6.

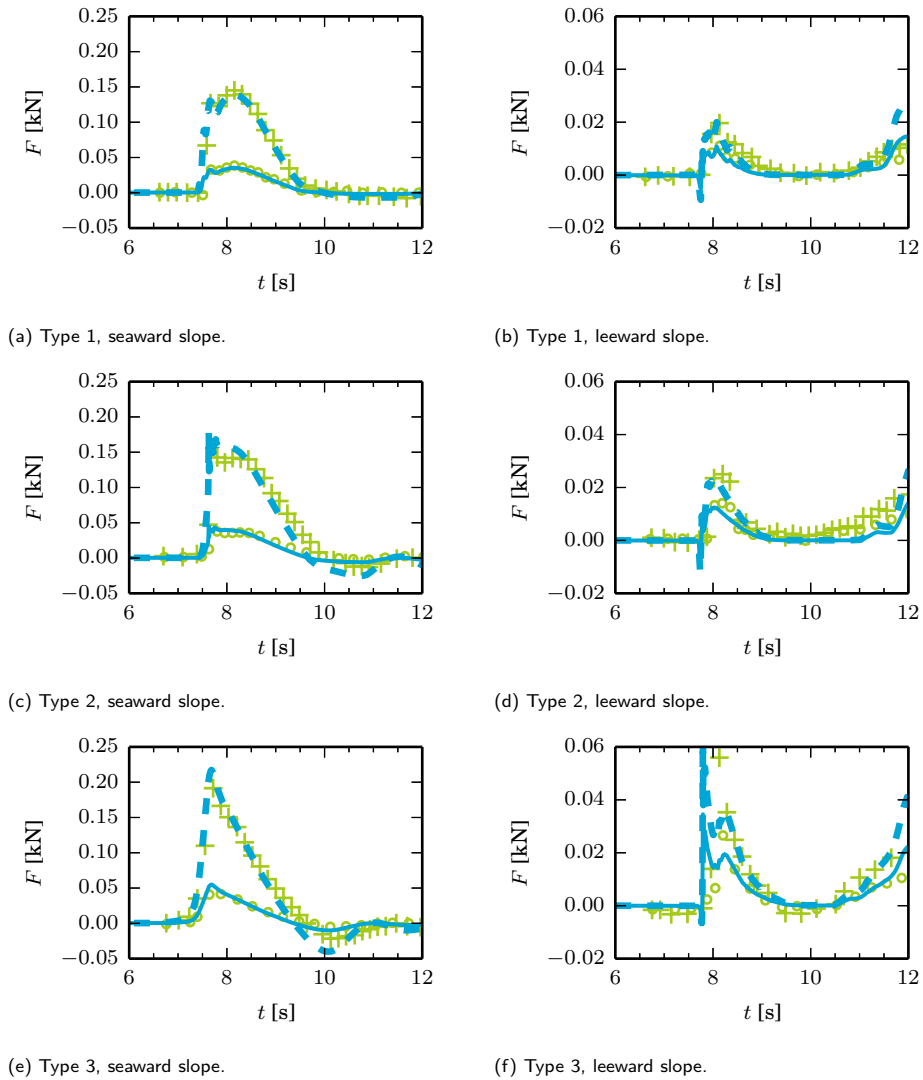


Figure 5.21: Breaking solitary wave: integrated force on seaward facing slope (left panels) and leeward facing slope (right panels) for three different wave types. Solid line is the vertical force F_y (positive downwards) and dashed line indicates the horizontal force F_x (positive to the right). Also shown are experimental values for F_y (+) and F_x (o) from [126].

5.6. Conclusion

The particle-mesh tools developed in the preceding two chapters have been combined in this chapter with the purpose of simulating incompressible, immiscible multifluid flows. The approach is based on a mixture formulation of the multiphase Navier-Stokes equations, in which material dependent parameters and specific momentum are tracked at the Lagrangian particles. Using the PDE-constrained particle-mesh projection to reconstruct density and momentum fields at the mesh, yields a scheme that conserves mass and momentum.

Various implementation considerations have been discussed, among which a penalty term to limit over- and undershoot in the density projection, and a penalty formulation for enforcing no-flux conditions at exterior boundaries. The seemingly simple hydrostatic case revealed that hydrostatic equilibrium is exactly represented in the particle-mesh scheme only if the mesh and the interface are aligned. To prevent spurious currents at the interface for non-aligned mesh-interface configurations, a simple cure is to increase the viscosity. This fix is as yet essential in cases where the dynamics are governed by small deviations from hydrostatic equilibrium. Arguably, the interface artifacts and potential solutions thereof, deserve further analysis.

The performance and applicability of the method is illustrated by a range of numerical examples. Excellent, and mass and momentum conserving results were obtained for a rising bubble and the lock exchange problem, in which the density ratios between the different liquid fractions are relatively moderate. Using an increased value for the dynamic viscosity, good agreement between experimental results and theoretical results for large-density ratio problems can be obtained. This included a test in which a regular wave train was generated from a moving piston-type wave maker, thereby showing the applicability of the scheme to reproduce, e.g., laboratory wave flume tests.

The applicability of the particle-mesh approach to practical problems arising in, e.g., civil engineering hydraulics, was further illustrated by considering a laboratory experiment for an air-water dambreak problem, and the propagation and breaking of a boundary-generated solitary wave onto a coastal structure. Excellent agreement between numerical and laboratory experiments was obtained, both in terms of the kinematics - i.e. interface topology and propagation - but most notably so in terms of dynamics, with the impact pressures and forces accurately predicted by the developed particle-mesh scheme.

6

Conclusion and outlook

We now return to the main objective of this research of combining diffusion-free Lagrangian particle-based techniques with the accuracy and conservation properties of state-of-the-art Eulerian mesh-based strategies for simulating advection dominated flows. The main conclusions collected in the several chapters are briefly recapitulated, and suggestions for future research are given in the second half of this chapter.

6.1. Conclusions

Framework: operator splitting

A first essential ingredient to combine particle- and mesh-based techniques successfully into one approach, requires a solid understanding of the role played by the particles and the mesh. Investigating the blueprint of existing particle-mesh schemes, such as PIC, FLIP and MPM, revealed that these strategies can be interpreted as an operator splitting approach. The splitting is based on the different physical processes by conveniently using the Lagrangian particles for the advection, i.e. the kinematic part of the problem, whereas the mesh is employed for the constitutive modeling to account for the inter-particle interactions.

This operator-splitting interpretation sheds new light on various aspects of particle-mesh schemes. First of all, it guides in developing time integration methods and elucidates that particle-mesh approaches inevitably introduce a *splitting error*, with the latter topic being extensively studied in the vast amount of literature available on - mesh-based - operator splitting methods. However, whereas a mesh-based operator splitting typically employs the same spatial discretization for the different operators, this is not so for particle-mesh schemes in which part of the problem is solved in a *particle space* and the other part is discretized in a *mesh space*. As a result, the introduction of data transfer operations is required, acting as projection operators between the particle space and the mesh space and *vice versa*. A last, yet crucial result of the operator-splitting interpretation of the particle-mesh scheme is that the aforementioned particle-mesh projections should be *auxiliary* steps. This implies that these steps should i) not interfere

nor obscure the relevant physics, and ii) should not compromise on conservation and accuracy properties of the particle-based advection step or the mesh-based diffusion step. The former criterion leads to a consistency requirement for the particle-mesh and the mesh-particle projections, in that the composition of the projection operators should yield the identity operator, see Eq. (2.31). The accuracy and conservation requirements are crucial when formulating the particle-mesh projection operators, as will be highlighted next.

Particle-mesh projections: a variational approach

Building upon the particle-mesh operator splitting interpretation, a generic variational framework for the particle-mesh interaction analogous to the mesh-based solver has been proposed. Essential for this framework is to interpret the particles as *sampling points* of the continuum, rather than that they represent volumetric blobs of the continuum. This seemingly subtle difference opens new perspectives for the reconstruction of mesh fields from the scattered particle data. More precisely, a *reconstruction* of mesh fields from the particle data in terms of a least squares minimization problem has been proposed in Chapter 2. The accompanying mesh-particle projection could be casted in a similar minimization problem, differing only in terms of the minimizer. This feature is key to render the particle-mesh and the mesh-particle projection mutually consistent.

The approach for the projection operators integrates particularly well with a discontinuous FEM discretization for the mesh-based problem, and factually intertwines these two steps. By employing a (H)DG framework, the particle-mesh projections amount to efficient cellwise operations, and high-order accuracy can be established. Mutual consistency of the particle-mesh and the mesh-particle projection was illustrated for the diffusion of the Gaussian hump on a 1D domain.

Particle-mesh projections: conservation principles

Chapter 3 explores the variational (H)DG framework to develop a particle-mesh operator splitting scheme for the scalar advection-diffusion equation. The major challenge in this chapter is to reconcile accuracy with discrete conservation properties for advection-dominated cases. To the author's knowledge, this was an as yet open issue in existing particle-mesh schemes, and the solution presented in Chapter 3 is therefore one of the key novelties of this thesis.

Extending the least squares objective function for the particle-mesh projection with a suitably chosen PDE-constraint, is crucial for obtaining rigorous (local) conservation properties. The key idea in formulating the constraint is that the projected particle motion must satisfy a hyperbolic conservation law from the perspective of the mesh. By expressing the control variable of the optimization problem in terms of single-valued advective fluxes at cell interfaces, the HDG method - which also exploits interface fields - naturally provides the necessary ingredients for formulating the optimality system. As such, the proposed scheme may be conceived alternatively as an HDG method in which stable advective fluxes are directly inferred from Lagrangian particle motion, thereby avoiding artificial dissipation as typically present in existing (H)DG methods.

Consistency and conservation of the constrained interaction were proven. Consistency requirements specific to particle-mesh methods were formulated, and a particular

fully-discrete formulation was derived to obey these requirements. Apart from optimal convergence rates and local conservation properties, the absence of artificial diffusion was highlighted as one of the distinguishing assets of the resulting particle-mesh scheme. In addition, the scheme circumvents measures for stabilizing the hyperbolic advection operator. The combination of these properties renders the particle-mesh approach particularly attractive to problems characterized by sharp flow features, such as interfaces or thin shear layers.

Although solving a global system is inevitable for the PDE-constrained particle-mesh projection, an efficient implementation is possible using static condensation, expressing the global problem in the control variable only. As an alternative and bound preserving option, a local (i.e. cellwise) ℓ^2 particle-mesh projection can be used, although this comes at the expense of compromising on exact conservation. In view of potential practical applications, it was shown that the PDE-constrained particle-mesh projection is amenable to a straightforward formulation on moving meshes.

Particle-distribution in incompressible flows

Chapter 4 extends the approach to the incompressible Navier-Stokes equations. The non-linearity of the advective term in these equations requires careful consideration, and a linearized version of the PDE-constrained particle-mesh projection was formulated. Another crucial aspect for simulating incompressible flows is that a uniform particle distribution has to be maintained throughout the simulations, since empty cells (i.e. cells not containing particles) would deteriorate the accuracy of the particle-mesh projection. One of the major contributions of Chapter 4 is to elucidate the influence of the divergence of the discrete advective velocity field on the particle distributions. Comparing a Taylor-Hood discretization for the unsteady Stokes equations, which only yields globally divergence-free velocity field, with locally divergence-free and H(div) conforming HDG discretizations revealed that a uniform particle-distribution is inherently maintained by the latter. Maintaining such a uniform particle distribution is much harder, if not impossible, for methods which do not satisfy the incompressibility constraint pointwise. Equipped with this insight, it is shown that even for challenging benchmark tests, such as high Reynolds number flows and/or flows involving stagnation points and flow separation, a uniform particle distribution is maintained without *ad hoc* particle shifting algorithms.

Application: multiphase flows

Chapter 5 investigates the applicability of the developed method to simulate multiphase flows as one of the potential application areas. In this context, Lagrangian particles are used to track the density fields in a diffusion-free manner. In combination with a PDE-constrained particle-mesh projection, this density tracking can be made mass conservative. In addition to tracking the density fields, the set of Lagrangian particles also carry the specific momentum, and analogous to the density projection, the particle-mesh projection step for this quantity can be rendered momentum conservative by virtue of the PDE-constrained projection. A number of practical issues were discussed, among which a straightforward implementation of surface tension forces, and a penalty term to limit over- and undershoot in the density projection.

As illustrated for a rising bubble and the lock exchange test, good results were obtained for problems in which the density ratio between the different fluid fractions is close to unity. Much more challenging, however, are cases characterized by a strong density jump, e.g. as for air-water interfaces. If the interface and the mesh are not aligned, spurious currents at the interface deteriorate model results. This artifact roots in failing to comply with the hydrostatic balance, and the deteriorating effect is most pronounced for cases in which the dynamics are governed by small deviations from hydrostatic equilibrium, such as small amplitude standing - or propagating waves. For the time being, the interface issue was mitigated by simply increasing the dynamic viscosity. It deserves, however, to investigate more advanced measures.

The potential of the particle-mesh scheme is demonstrated in Chapter 5 by simulating a number of experiments with violent free surface motion, including the simulation of a dambreak problem and the propagation, breaking, and overtopping of a solitary wave on a coastal structure. The complex topology of the air-water interface was accurately predicted by the proposed numerical scheme for these tests, and good, not to say excellent, correspondence between simulated and measured impact pressures and forces were obtained with reasonable computational effort.

The proposed particle-mesh scheme, or parts thereof, can be used to simulate a wide range of advection-dominated flows, and potential applications certainly extend beyond the examples shown for multiphase flows. To encourage users to apply the suite of particle-mesh tools to their own problems, the developed code - labeled LEOPart - is freely available under an open source license via https://bitbucket.org/jakob_maljaars/leopart. Several implementation details, features and performance tests for LEOPart are presented in Appendix D.

6.2. Outlook and perspectives

The conservative, accurate and numerical diffusion-free particle-mesh techniques developed in this work, offer a compelling alternative to existing particle- or mesh-based approaches for incompressible, advection-dominated problems. At the same time, some of the concepts were only briefly touched in the scope of this research, and further extensions and analyses could strengthen and generalize the underlying theoretical basis. Moreover, solving a number of fundamental and practical challenges encountered throughout, could further bolster the applicability of the particle-mesh tools.

To provide guidance, various suggestions and directions for further research are given in this section. This includes a number of more fundamental recommendations regarding the time stepping and the particle-mesh projections, as well as a number of implementation recommendations and application perspectives.

Time stepping accuracy

Consulting the existing particle(-mesh) literature, the conclusion seems valid that recent developments mainly focus on improving the spatial accuracy. In a sense, this work forms no exception to this. Whereas the proposed scheme can be rendered arbitrary order accurate in space, no attempts were really made to push the time accuracy beyond second-order. The operator-splitting interpretation of particle-mesh schemes de-

veloped in Chapter 2, nevertheless provides a clear framework to formulate alternative, and possibly higher-order time stepping schemes. Particularly attractive candidates are multistep implicit-explicit (IMEX) schemes, in which an explicit scheme is used for the (Lagrangian) advection, and an implicit scheme for solving the mesh-based dynamic equations, e.g. the diffusion equation or the Stokes equations [128].

Another topic which is worth investigating is the admissible time step size. In this thesis, the time step size was typically chosen such that the CFL number is approximately around a value of 1. This can be expected to be a rather conservative choice, and it can be questioned if there is a necessary CFL condition at all. Evidently, this is not so when combining the Lagrangian particle advection with the local ℓ^2 particle-mesh projection for reconstructing fields at the mesh: since for this setup no information is stored at the mesh between consecutive time steps, and the Lagrangian advection step is obviously not subjected to a necessary CFL condition. It is an as yet unanswered question whether or not this remains the case for the PDE-constrained particle-mesh projection, in which a mesh field is stored between consecutive time steps.

Analysis of the (constrained) projections

The PDE-constrained particle-mesh projection is one of the key novelties in this work, and apart from the earlier mentioned analysis on the CFL-condition, several other aspects are worth investigating. This includes, for instance, an in-depth analysis of the role played by the Lagrange multiplier. In this thesis, the approximation space for the Lagrange multiplier was set to the lowest-order, piecewise constant polynomial approximation, which is sufficient to ensure mass and momentum conservation. However, a potential advantage of using higher-order polynomial approximation might be the conservation of quadratic quantities such as energy.

Another fruitful area for further research is to render the PDE-constrained particle-mesh projection bound preserving or, even better, strictly monotone. This will be a powerful asset in problems for which the particle-mesh strategy is particularly apt, i.e. the simulation of sharp features. A potential starting point is the bound preserving remap strategy proposed by Bochev and coworkers [92], which imposes additional box constraints. Another idea is to further investigate the gradient penalty term, introduced in Chapter 5 to alleviate under- and overshoot. Instead of making a somewhat *ad hoc* choice for the penalty term ζ , a physics-based choice for this term can be made by exploring variation entropy concepts, for which the work by Ten Eikelder [94] offers a first starting point.

At an abstract level, the PDE-constrained approach for combining particle-based and mesh-based techniques is drastically different from existing solution strategies for discretizing (hyperbolic) conservation laws. Instead of the conservation law itself, the starting point is the minimization of a certain objective function, subject to the constraint of satisfying a conservation law. The HDG approach appears particularly suited for such problems, since the interface fields naturally provides the necessary optimality control. This idea of '*optimization-via-hybridization*' can be used for other objective function plus constraint pairs, such as imposing the incompressibility condition as a constraint during the particle-mesh projection, and may find applications which extend beyond the particle-mesh setting altogether.

Implementation and application perspectives

Chapter 5 of the thesis showed the potential of the particle-mesh scheme for simulating multiphase flows. Results are compelling and promising. In particular, tracking the density fields in a diffusion-free, yet conservative way is one of the assets which sets the developed particle-mesh strategy apart from existing approaches. This feature renders the scheme particularly attractive for multiphase applications in which the interface between the different fluid fractions is of a complex shape. One such application is the simulation of breaking ocean waves, and it was demonstrated that different wave flume laboratory experiments were reproduced successfully by the numerical model. At the same time, various numerical examples clearly revealed the open challenges. Most profoundly, achieving hydrostatic equilibrium turned out a major challenge in the chosen one-fluid model, which results in interface artifacts for low-diffusion, high-density ratio problems. Possible research directions for solving this issue can include fundamental modifications, e.g. using a different solution strategy for the unsteady Stokes equations, or to depart from the mixture model approximation altogether by defining a separate set of equations for the different fluid fractions and couple the fractions by appropriate boundary conditions. Keeping the mesh and the interface aligned in regions where the dynamics are dominated by small deviations from the hydrostatic balance, while simultaneously relying on the particles to capture complex interface topologies - e.g. in wave breaking zones - could be another research direction.

From an implementation perspective, scalable, iterative solvers are the major missing piece to make the proposed particle-mesh scheme a viable option for solving large scale problems in three spatial dimensions. This is clearly illustrated in the performance tests reported in Appendix D, showing that solving the unsteady Stokes equations and the systems arising from the PDE-constrained particle-mesh projection consumes most of the computational time. Implementing the iterative scheme from [129] for solving the Stokes equations will be a first leap to reduce computational burden. Other practical suggestions for code improvement are found in the issue tracker of LEOPart.

Finally, it is believed that the suite of particle-mesh tools developed in this thesis allows application to a much wider range of problems than was discussed in this thesis. This includes, e.g., passive and active tracer modeling, which finds applications in groundwater modeling, mantle convection problems, atmospheric modeling, and reproducing PIV measurements in numerical simulations. Of particular interest are applications characterized by low physical diffusion, for which the presented particle-mesh strategy allows to maintain sharp features at subgrid level without numerical diffusion. Given the open-source computer code, users are invited to tailor the developed particle-mesh tools to their own problems and help the development of LEOPart to accelerate.

A

The local ℓ^2 -projections and moving least squares

This appendix shows that on simplices, the local ℓ^2 -projections onto the HDG basis can be considered as a specific variant of the moving least squares (MLS) reconstruction technique.

Without loss of generality this appendix considers a scalar-valued function $\psi(\mathbf{x})$ that varies in space only, ignoring time dependency. As a result of the latter, the subscripts in the particle set notations are dropped (i.e. \mathcal{S} denotes the time-independent counterpart of \mathcal{S}_t (Eq. (3.8)) and \mathcal{S}^K denotes the time-independent counterpart of \mathcal{S}_t^K (Eq. (3.9)). Furthermore, let $\psi_h(\mathbf{x}) \in W_h$ be an approximation of the scalar-valued function $\psi(\mathbf{x})$ on the piecewise continuous function space W_h spanned by Lagrange polynomials of order k , see Eq. (3.4). Let a monomial approximation $\tilde{\psi}_h$ of the scalar-valued function $\psi(\mathbf{x})$ be given by

$$\tilde{\psi}_h(\mathbf{x}) = \mathbf{p}^\top(\mathbf{x}) \mathbf{a}(\mathbf{x}), \quad (\text{A.1})$$

where $\mathbf{a}(\mathbf{x})$ is the (potentially spatially dependent) coefficient vector of the monomial basis $\mathbf{p}^\top(\mathbf{x})$ of order m . In 1D, this basis is of the form $\mathbf{p}^\top(x) = [1, x, x^2, \dots, x^m]$.

In MLS, the coefficients $\mathbf{a}(\mathbf{x})$ are obtained by minimizing the residual between the local approximation $\tilde{\psi}_h(\mathbf{x})$ and the scattered particle values, say ψ_p , in a weighted least-square sense, i.e.

$$\min_{\mathbf{a}(\mathbf{x})} J_1 := \sum_{p \in \mathcal{S}} w(\mathbf{x} - \mathbf{x}_p) (\mathbf{p}^\top(\mathbf{x}_p) \mathbf{a}(\mathbf{x}) - \psi_p)^2, \quad (\text{A.2})$$

with $w(\mathbf{x} - \mathbf{x}_p)$ being a weighting function, usually taken to be symmetric around the particle positions \mathbf{x}_p and having local support (see e.g. [40, 130]). Furthermore, ψ_p is a scalar value associated with particle p .

Performing the minimization (see, e.g., [40, 130]) leads to the following expression for the coefficients $\mathbf{a}(\mathbf{x})$:

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}(\mathbf{x})^{-1} \mathbf{B}(\mathbf{x}) \boldsymbol{\psi}_p, \quad (\text{A.3})$$

with

$$\mathbf{A}(\mathbf{x}) = \sum_{p \in \mathcal{S}} \mathbf{p}(\mathbf{x}_p) w(\mathbf{x} - \mathbf{x}_p) \mathbf{p}^\top(\mathbf{x}_p), \quad (\text{A.4})$$

$$\mathbf{B}(\mathbf{x}) \boldsymbol{\psi}_p = \sum_{p \in \mathcal{S}} \mathbf{p}(\mathbf{x}_p) w(\mathbf{x} - \mathbf{x}_p) \psi_p. \quad (\text{A.5})$$

If the particle weighting function $w(\mathbf{x})$ has limited support, the summations in Eq. (A.4) and Eq. (A.5) only involve a (small) subset of the particles in \mathcal{S} . Furthermore, it readily follows that the spatial dependency of the coefficients $\mathbf{a}(\mathbf{x})$ is a result of the weighting function being spatially dependent.

To show the similarity between the proposed ℓ^2 -projections and an MLS-reconstruction, consider an ‘indicator’ weighting function, being equal to 1 within the cell K hosting particle p and 0 elsewhere

$$w(\mathbf{x} - \mathbf{x}_p) = \begin{cases} 1, & \text{if } \mathbf{x}_p \in \bar{K}. \\ 0, & \text{otherwise.} \end{cases} \quad (\text{A.6})$$

For this particular choice, Eq. (A.2) results in a sum of independent quadratic forms over all cells $K \in \mathcal{T}$

$$J_1 = \sum_K J_1^K = \sum_K \sum_{p \in \mathcal{S}^K} (\mathbf{p}^\top(\mathbf{x}_p) \mathbf{a} - \psi_p)^2. \quad (\text{A.7})$$

Furthermore, the coefficients $\mathbf{a}(\mathbf{x})$ within a cell K are independent of the spatial coordinates $\mathbf{x} \in K$ for this particular choice of the weighting function, see Eq. (A.3).

For the presented ℓ^2 -projections, the minimization problem defined by Eq. (3.13) for a scalar quantity reads

$$\min_{\psi_h \in W_h} J_2 := \sum_{p \in \mathcal{S}} (\psi_h(\mathbf{x}_p) - \psi_p)^2. \quad (\text{A.8})$$

This can also be rewritten as a set of independent minimization problems covering all cells $K \in \mathcal{T}$,

$$J_2 = \sum_K J_2^K \equiv \sum_K \sum_{p \in \mathcal{S}^K} (\psi_h(\mathbf{x}_p) - \psi_p)^2, \quad (\text{A.9})$$

since $\psi_h \in W_h$ is discontinuous between cells. Furthermore, $\psi_h(\mathbf{x}_p)$ can be expanded as

$$\psi_h(\mathbf{x}_p) = \sum_i N_i(\mathbf{x}_p) \psi_i \equiv \mathbf{N}(\mathbf{x}_p) \boldsymbol{\psi}, \quad (\text{A.10})$$

in which $N_i(\mathbf{x})$ and ψ_i denote the discontinuous Lagrange polynomial basis function and (unknown) scalar value at node i , respectively.

On simplices, the Lagrange polynomial basis of order m can be rewritten in terms of a monomial basis of order m times a square coefficient matrix \mathbf{C} , i.e.

$$\mathbf{N}(\mathbf{x}) = \mathbf{p}^\top(\mathbf{x}) \mathbf{C}, \quad (\text{A.11})$$

where the coefficient matrix \mathbf{C} is uniquely determined by the element geometry and the Kronecker δ -property.

Upon subsequent substitution of Eq. (A.11) in Eq. (A.10) and Eq. (A.9), the latter can be rewritten as

$$\sum_K J_2^K = \sum_K \sum_{p \in \mathcal{S}^K} (\mathbf{p}^\top(\mathbf{x}_p) \mathbf{C} \boldsymbol{\psi} - \psi_p)^2. \quad (\text{A.12})$$

Comparing Eq. (A.12) with the minimization problem resulting from an MLS-reconstruction (Eq. (A.7)) readily implies the equality

$$\mathbf{a} \equiv \mathbf{C} \boldsymbol{\psi}, \quad (\text{A.13})$$

where we recall that \mathbf{C} is a square coefficient matrix.

Hence, it is concluded that the presented ℓ^2 -projections onto a discontinuous function space can be interpreted as an MLS-reconstruction procedure, using a monomial basis and a piecewise constant indicator function as particle weighting functions, Eq. (A.6). Given this specific choice:

- the coefficients $\mathbf{a}(\mathbf{x}) \equiv \mathbf{a}$ are independent of the spatial coordinates within a cell K . Hence, a (small) matrix has to be inverted once for each cell. From Eq. (A.3) it follows that in general a matrix (albeit being small) has to be inverted for every point at which the MLS-reconstruction is to be evaluated.
- the influence of a particle remains strictly local to the hosting cell.

B

Time accuracy considerations

This appendix shows that a particle update using Eq. (3.33) in conjunction with a backward Euler discretization of the diffusion problem from Eq. (3.32), results in a scheme being second-order accurate in time.

Consider the abstract representation of the discrete diffusion equation, using the θ -scheme

$$\frac{\phi_h^{n+1} - \phi_h^{*,n}}{\Delta t} = \theta F_h^{n+1} + (1 - \theta) F_h^n, \quad (\text{B.1})$$

in which F_h includes the discretization of the diffusion operator and, optionally, the source term. Furthermore, $\theta \in [1/2, 1]$ is a parameter, with $\theta = 1$ resulting in a first-order accurate backward Euler scheme and $\theta = 0.5$ yields the second-order accurate Crank-Nicolson method.

Since Eq. (B.1) is linear, it can be splitted into the weighted sum of an implicit and an explicit update, i.e.

$$\frac{\phi_h^{n+1} - \phi_h^{*,n}}{\Delta t} = \theta \frac{\phi_{h,I}^{n+1} - \phi_h^{*,n}}{\Delta t} + (1 - \theta) \frac{\phi_{h,E}^{n+1} - \phi_h^{*,n}}{\Delta t}. \quad (\text{B.2})$$

Combining Eq. (B.2) with Eq. (B.1), the implicit and the explicit part are respectively given by

$$\dot{\phi}_{h,I}^{n+1} \equiv \frac{\phi_{h,I}^{n+1} - \phi_h^{*,n}}{\Delta t} = F_h^{n+1}, \quad (\text{B.3})$$

$$\dot{\phi}_{h,E}^{n+1} \equiv \frac{\phi_{h,E}^{n+1} - \phi_h^{*,n}}{\Delta t} = F_h^n. \quad (\text{B.4})$$

Comparing the right-hand sides of these equations, it becomes clear that

$$\dot{\phi}_{h,E}^{n+1} = \dot{\phi}_{h,I}^n. \quad (\text{B.5})$$

Hence, we can write the particle update (Eq. (3.33)) as

$$\begin{aligned} \frac{\psi_p^{n+1} - \psi_p^n}{\Delta t} &= \theta \dot{\phi}_{h,I}^{n+1}(\mathbf{x}_p^{n+1}) + (1 - \theta) \dot{\phi}_{h,I}^n(\mathbf{x}_p^n), \\ &= \theta \dot{\phi}_{h,I}^{n+1}(\mathbf{x}_p^{n+1}) + (1 - \theta) \dot{\phi}_{h,E}^{n+1}(\mathbf{x}_p^n), \end{aligned} \quad (\text{B.6})$$

where the implicit and explicit part have to be evaluated at the new and old position \mathbf{x}_p^{n+1} and \mathbf{x}_p^n , respectively.

From Eq. (B.6) the following observations are made

- Eq. (B.6) is equivalent to the presented mesh-particle projection Eq. (3.33).
- Hence, second-order accuracy in time can be obtained by storing the implicit result from the previous time step at particle level and reuse it as the explicit part in the current time step.
- The advantage of this ‘particle-based’ second-order accuracy over a ‘mesh-based’ second-order accuracy is that no information needs to be stored at the background mesh between consecutive time steps in the former approach when using the local ℓ^2 particle-mesh projection.
- The trade-off of the ‘particle-based’ second-order accuracy is that it i) requires an initialization step since $\dot{\phi}_{h,I}^n(\mathbf{x}_p^n)$ is not available for $n = 0$, and ii) additional data $\dot{\phi}_{h,I}^n(\mathbf{x}_p^n)$ has to be stored at the particle-level.

C

Surface tension force

The external forcing term in the local momentum balance, Eq. (5.22a), consists of the body force \mathbf{f}_G and the surface tension force \mathbf{f}_T . The discretization of the latter term \mathbf{f}_T is subject of ongoing research, particularly so on unstructured meshes, see e.g. the recent review article by Popinet [131]. In this thesis, the relatively straightforward continuum surface force (CSF) model introduced by Brackbill and coworkers [117] is used, in which the surface tension force is approximated as a volumetric force \mathbf{f}_T , i.e.

$$\int_{\Omega} \mathbf{f}_T \cdot \mathbf{w}_h \, d\Omega \approx \oint_{\Gamma_I} \sigma \kappa \mathbf{n}_I \delta_I \cdot \mathbf{w}_h \, d\Gamma, \quad (\text{C.1})$$

where σ the surface tension coefficient, κ the curvature of the interface, \mathbf{n}_I the normal to the interface Γ_I , and δ_I the Dirac-delta function being non-zero only on the interface. Referring to [117] for an in-depth derivation, Eq. (C.1) can be approximated in a fully-discrete setting as

$$\oint_{\Gamma_I} \sigma \kappa \mathbf{n}_I \delta_I \cdot \mathbf{w}_h \, d\Gamma \approx \sum_K \int_K \sigma \kappa^{n+1} \frac{\nabla \tilde{\rho}_h^{n+1}}{[\![\rho]\!] } \frac{\tilde{\rho}_h^{n+1}}{\langle \rho \rangle} \cdot \mathbf{w}_h \, d\Omega \quad (\text{C.2})$$

in which the jump $[\![\rho]\!]$ and the average $\langle \rho \rangle$ are defined as

$$[\![\rho]\!] := \rho_1 - \rho_2 \quad \text{and} \quad \langle \rho \rangle := \frac{\rho_1 + \rho_2}{2},$$

with ρ_1 the density of the heavy fraction and ρ_2 the density of the light fraction. Furthermore, a twice differentiable density field $\tilde{\rho}_h^{n+1}$ is obtained from the reconstructed and discontinuous density field ρ_h^{n+1} by the projection

$$\int_{\Omega} (\tilde{\rho}_h^{n+1} \pi_h + \epsilon \nabla \tilde{\rho}_h^{n+1} \cdot \nabla \pi_h) \, d\Omega = \sum_K \int_{\Omega} \rho_h^{n+1} \pi_h \, d\Omega \quad \forall \pi_h \in \Pi_h, \quad (\text{C.3})$$

in which ϵ a diffusion parameter, related to a characteristic mesh size, and $\Pi_h = \{\pi_h \in H^1(\mathcal{T}) : \pi_h \in P^2(\bar{K}) \forall K \in \mathcal{T}\}$ the function space of C^0 continuous polynomials being twice continuously differentiable inside a cell. The latter is required for obtaining the discrete curvature κ^{n+1} , computed as

$$\kappa^{n+1} = -\nabla \cdot \frac{\nabla \hat{\rho}_h^{n+1}}{\|\nabla \hat{\rho}_h^{n+1}\|}, \quad (\text{C.4})$$

which completes a fully-discrete implementation of Eq. (C.2).

D

Computer implementation

In this appendix, the essentials of the computer code - coined LEOPart - developed as part of this research are briefly discussed. Not surprisingly, this code entails a 'mesh part' and a 'particle part', the latter for handling the particle-related parts, and the former for solving the equations arising at the mesh, including the projections to and from the mesh to the particles.

To invite researchers to apply the particle-mesh strategy developed in this thesis to their own challenges, LEOPart is available under an open-source license and is hosted at https://bitbucket.org/jakob_maljaars/leopart/. When needed, reference is made to the relevant filenames in this repository in the sequel of this appendix.

D.1. FEniCS

In Chapter 2 variational formulations were proposed as the encompassing framework for the mesh problem and the particle mesh interactions. To solve the resulting weak formulations at the background mesh, tools from the open-source finite element library FEniCS are used [90]. The FEniCS project aims at automating the solution of partial differential equations (PDEs) by providing a generic framework for formulating, discretizing and solving variational forms. The philosophy behind FEniCS is to generate efficient, low-level code from a high-level, pseudo-mathematical formulation of the variational forms describing the problem of interest. This generic approach renders the library applicable to a wide range of problems, see [90] for a selection of applications. Moreover, it allows researchers to experiment with an extensive collection of finite element function spaces and arbitrary polynomial orders, without having to worry about tedious implementation details. At least, as long as one trusts the tools provided by the employed finite element library (see proposition 1 in [75]).

In short, the procedure of solving PDEs using the FEM approach drops down to the following sequence of steps:

1. Formulate the variational forms of the PDEs, supplemented with proper boundary and initial conditions.

2. Discretize the variational forms at the level of the individual element.
3. Assemble of the element contributions into a global system of equations.
4. Solve the global system of equations.

Via different components, FEniCS facilitates the computer implementation for each of these steps. The so-called 'Unified Form Language' (UFL) enables a user to express the discrete variational form in a way that closely resembles the mathematical notation (step 1). As an example, consider the following weak form of the Laplace operator

$$A := \int_{\Omega} \nabla \psi \cdot \nabla w \, d\Omega - \oint_{\partial\Omega} \nabla \psi \cdot \mathbf{n} w \, d\Gamma. \quad (\text{D.1})$$

Defining this operator on a unit square mesh using piecewise continuous linear shape functions requires just the following lines of code using the UFL language:

Python code

```
from dolfin import (UnitSquareMesh, FunctionSpace, TestFunction,
                   TrialFunction, FacetNormal)

# Define 20 x 20 mesh and facet normals
mesh = UnitSquareMesh(20, 20)
n = FacetNormal(mesh)

# Define function space and test and trial function
W = FunctionSpace(mesh, 'cg', 1)
psi = TrialFunction(W)
w = TestFunction(W)

A = dot(grad(psi), grad(w)) * dx - dot(grad(psi), n) * w * ds
```

The Fenics Form Compiler (FFC) is invoked to translate the high-level UFL description into low-level and efficient C++ code that is used to evaluate and assemble the local element tensors into a global system of equations (step 2 and 3). Finally, DOLFIN provides the FEniCS problem solving environment in Python and C++, and provides an interface to various linear algebra backends such as PETSc to solve the global system of equations (step 4). For an extensive description of these components, reference is made to the FEniCS book [90].

D.1.1. Static condensation

Elimination of the local unknowns via static condensation forms an essential part of the solution strategy when employing an HDG discretization. Static condensation features, however, are not natively provided in the FEniCS library, and the assembling of the local problems into the condensed global system is done with customized assembling routines provided by LEoPart. UFL and FFC routines are conveniently used for obtaining the element tensors for the relevant discrete variational forms arising at the mesh, and creating a blueprint of the global problem. To render the customized assembling routines computationally efficient, the implementation stays as close as possible to the data structures provided by the FEniCS library.

Illustrative is to consider a minimal working example (MWE), demonstrating how LEOPart can be used to solve the algebraic system arising from a HDG discretization of the steady Stokes equations using static condensation. The notations in this code example closely follow the block structure from Eq. (4.42):

Python code

```

from dolfin import (FunctionSpace, MixedElement,
                   VectorElement, FiniteElement, Function,
                   UnitSquareMesh, Constant, DirichletBC)
from leopart import (FormsStokes, StokesStaticCondensation)
import numpy.random as np_rand

# (nx, k, nu) --> (mesh resolution, polynomial order, viscosity)
(nx, k, nu) = (16, 1, Constant(1))
alpha = Constant(6*k*k)

mesh = UnitSquareMesh(nx, nx)

# Define HDG FunctionSpaces and functions
(V, Q) = (VectorElement("DG", mesh.ufl_cell(), k),
         FiniteElement("DG", mesh.ufl_cell(), k-1))
(Vbar, Qbar) = (VectorElement("DGT", mesh.ufl_cell(), k),
               FiniteElement("DGT", mesh.ufl_cell(), k))

mixedL = FunctionSpace(mesh, MixedElement([V, Q]))
mixedG = FunctionSpace(mesh, MixedElement([Vbar, Qbar]))

(Uh, Uhbar) = (Function(mixedL), Function(mixedG))

# Initialize and assign random source term
f = Function(FunctionSpace(mesh, V))
f.vector().set_local(np_rand.rand(f.vector().local_size()))

# No-slip boundary conditions
bc0 = DirichletBC(mixedG.sub(0), Constant((0, 0)), 'on_boundary')

# Set forms, and prepare static condensation class
forms = FormsStokes(mesh, mixedL, mixedG, alpha).forms_steady(nu, f)
ssc = StokesStaticCondensation(mesh,
                               forms['A_S'], forms['G_S'],
                               forms['B_S'],
                               forms['Q_S'], forms['S_S'], [bc0])

# Assemble global system, assemble_lhs=True
ssc.assemble_global_system(True)
# Solve using mumps
ssc.solve_problem(Uhbar, Uh, "mumps", "default")

```

On one hand, this MWE clearly illustrates the advantages of building LEOPart on top of DOLFIN as this allows using the meshing tools, function spaces, and imposing boundary conditions. Simultaneously, although perhaps hidden at first sight, it also shows that there is a price to be paid in terms of flexibility. Most profoundly, the current stable FEniCS version (2018.2.0) dictates a formulation of the problem in terms of a *mixed* form for the local problem, and a *mixed* form for the global problem. This comes with the

major drawback in that the implementation of efficient iterative solvers becomes cumbersome, since this typically requires control over all the different contributing blocks, see e.g. [129]. As a result, solving the global system for the Stokes equations is as yet only possible with direct sparse solvers, such as MUMPS or SuperLU.

D.2. Particle administration

The second part of the computer implementation constitutes the handling of the particles. In particular, it concerns the question how to keep track of the particles on the mesh in an efficient and generic manner and how to deal with the particles in the presence of boundaries.

D.2.1. The particle class

The `particles` class is the backbone for dealing with the Lagrangian particles in `LEoPart`. Operations such as the particle-advection and the particle-mesh projections require as input an instance of this `particles` class. In `LEoPart`, particles always possess a position at slot 0, the other slots - read: particle values - can be user specified. For instance, if one wants to instantiate particles carrying a vector-valued velocity field and a scalar valued density field, this is done as follows:

Python code

```
from dolfin import UnitSquareMesh
from leopart import particles
import numpy as np

msh = UnitSquareMesh(2, 2)

# Create 4 particles in 2D
xp = np.random.rand(4, 2)
up = np.zeros([xp.shape[0], xp.shape[1]])
rhoh = np.zeros(xp.shape[0])

# Initialize particles class
p = particles(xp, [up, rhoh], msh)
```

Note that particles in `LEoPart` are always defined on a background mesh.

Interpolating a mesh field to a particle slot, say slot 2 on which the particle density is stored, is as easy as extending the code with

Python code

```
from dolfin import (FunctionSpace, Function, Expression)

# Make density field at mesh
Q = FunctionSpace(msh, 'DG', 1)
rhoh = Function(Q)
rhoh.assign(Expression('x[0]', degree=1))
# Interpolate to particle, density stored at slot 2
p.interpolate(rhoh, 2)
```

D.2.2. Particle advection

Particle advection in LEOpart is handled via the advection classes available in the header file `advect_particles.h`. As yet, LEOpart supports the first-order Euler forward method - via the `advect_particles` class - and the two and three stage Runge-Kutta method - available via the `advect_rk2` and the `advect_rk3` class, respectively. The multi-stage Runge-Kutta advection schemes inherit from the `advect_particles` class. Multi-step schemes such as the Adams-Bashforth schemes are as yet not supported.

To illustrate how the particle-advection schemes are interfaced, consider the advection of a disk-shaped patch of particles in the solenoidal velocity field defined by Eq. (3.51) with $g(t) = \cos \frac{1}{8}\pi t$. Apart from its coordinates, a particle carries a scalar value ρ_p , with this value being initialized based on the initial distance to center of the disk-shaped region filled with particles. Using the three-stage Runge-Kutta scheme for the particle advection, and leaving out post-processing commands, the code for this example reads

Python code

```

from dolfin import (UnitSquareMesh, VectorFunctionSpace,
                   Function, Expression, Constant)
from leopart import (particles, advect_rk3, RandomCircle)
import numpy as np

# Time and timestep
(t, dt, tend) = (Constant(0), Constant(0.05), 8.)
gt = 'cos(0.125*pi*t)'
v_e = Expression((gt + '*pow(sin(pi*x[0]), 2)*sin(2*pi*x[1])',
                  gt + '*-1*pow(sin(pi*x[1]), 2)*sin(2*pi*x[0])'),
                 t=float(t), degree=1)

# Initialize mesh and velocity field
msh = UnitSquareMesh(20, 20)
V = VectorFunctionSpace(msh, "CG", 1)
v = Function(V)
v.assign(v_e)

# Particle position and scalar particle field
(ctr, r, npt) = ((0.25, 0.5), 0.15, 100)
xp_0 = RandomCircle(center=ctr, radius=r).generate([npt, npt])
rhop_0 = (-np.linalg.norm(xp_0 - np.array(ctr), axis=1)+r)/r

# Initialize particles and advection, do time stepping
p = particles(xp_0, [rhop_0], msh)
ap = advect_rk3(p, V, v, 'closed')
while float(t) < tend:
    ap.do_step(float(dt))
    t.assign(float(t) + float(dt))
    v_e.t = float(t)
    v.assign(v_e)
(xp_n, rhop_n) = (p.return_property(msh, 0), p.return_property(msh,
1))

```

Executing this piece of code, yields the particle fields as visualized in Fig. D.1 at $t = (0, 4, 8)$.

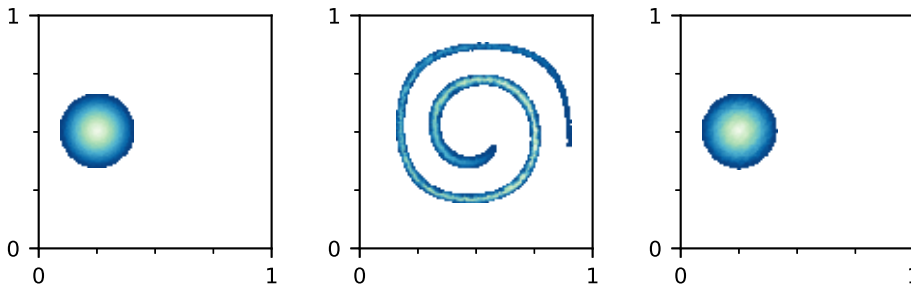


Figure D.1: Particle advection: particle fields at $t = (0, 4, 8)$ (left-to-right) using LEOPart tools.

D.2.3. Particle tracking

The particle advection as well as the particle projection require the evaluation of mesh fields and hence the evaluation of the basis function values. It therefore must be known in which cell a particle resides. This wouldn't be much of an issue on a regular Cartesian mesh, but on arbitrary polyhedral - and possibly fully-unstructured meshes - this particle tracking is less straightforward. One option indeed might be to do a global search every time step to find the hosting cell for each particle, but this doesn't seem particularly efficient as it ignores information from the previous time steps. Indeed, for a sufficiently small time step Δt , particles are expected to stay within or at least close to the cell which originally hosted the particle. This assumption is exploited to keep track of the moving particle at the mesh efficiently.

At a meta-level, two options are available in order to do so: a first option is that each particle carries a notion of its hosting cell. As soon as a particle crosses a cell border, this particle property is updated. The second option is to perceive a cell as a bucket filled with particles. Rather than that a particle keeps track of its hosting cell, the bookkeeping is done at the cell level: each cell contains a list of particles. When a particle leaves a cell, the particle is removed from the cell's 'particle bucket' and the particle is added to the receiving cell's particle bucket. In the developed code, the second path is pursued. Main motivation for doing so, is that it renders the evaluation of a mesh-field at the particle positions more efficiently: the evaluation of the expansion coefficient for a mesh-field is done only once per cell, and subsequently reused for all particles contained within that cell. Secondly, in parallel distributed memory computations, the domain decomposition of the mesh - taken care of by FEniCS - provides a convenient structure to store the particle data on the different processes. From a load balance perspective, it is noted that this approach is expected to be particularly suited for situations in which particles are spreaded over the entire domain.

A second question is how to efficiently keep track of the hosting cell for the Lagrangian particles at the unstructured polyhedral mesh. Several procedures have been developed to this end, based on, for example, information from linear basis functions, superposition of a coarse Cartesian mesh onto the unstructured mesh [33], or methods based on barycentric interpolation [82]. An alternative method is the convex polyhedron method [49]. This method assumes that the mesh consists of convex polyhedral cells, which

indeed is the case for the simplicial cells used in this thesis. The midpoint, the unit normal, and the connectivity for each *facet* in the mesh are pre-computed. Concerning the connectivity, a facet has two neighboring cells for facets internal to the mesh, and only one neighboring cell for boundary facets, including the internal facets which are located on processor boundaries. From the perspective of cell K the sign of the facet unit normals is adapted so as to make sure that the unit normals are outward directed.

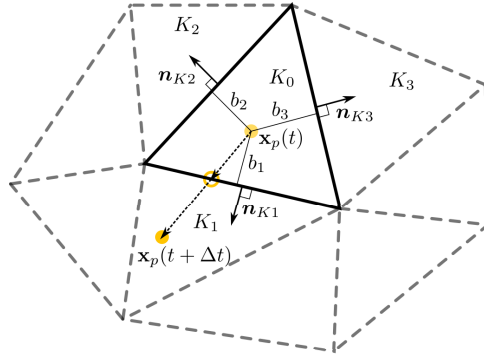


Figure D.2: Particle tracking using the convex polyhedron strategy.

The convex polyhedron particle tracking then proceeds as follows for a particle p , having location $\mathbf{x}_p(t)$ at time t within cell K_0 , see Fig. D.2 for a principle sketch. Assume the particle has a velocity $\mathbf{u}(\mathbf{x}_p, t)$ and the time step used for advecting the particle is $\Delta t^{(0)}$, where the use of a superscript will become clear shortly. First, the time to intersect the i -th facet of element K_0 is computed as $\Delta t_i = b_i / (\mathbf{u}(\mathbf{x}_p, t) \cdot \mathbf{n}_{K_i})$ for all facets, with b_i the orthogonal distance between the particle and facet with index i . Next, the minimum, yet positive, time to intersection is computed as $\Delta t_{i_{\min}} = \min(\max(0, \Delta t_i))$. If $\Delta t_{i_{\min}} > \Delta t^{(k)}$ the particle is pushed to its new position using timestep $\Delta t^{(k)}$, and the time step is terminated by setting $\Delta t^{(k+1)} = 0$. If $\Delta t_{i_{\min}} < \Delta t^{(k)}$, the particle p is pushed to the facet intersection $\mathbf{x}_p^{(k)}$ using $\Delta t_{i_{\min}}$, and the hosting cell is updated to facet i_{\min} , that is the facet with the index corresponding to $\Delta t_{i_{\min}}$. Furthermore, the timestep is decremented to $\Delta t^{(k+1)} = \Delta t^{(k)} - \Delta t_{i_{\min}}$, after which the particle tracking continues until the time step for a particle has zero time remaining. Algorithm 1 contains the pseudo-code for the convex polyhedron particle tracking.

The convex polyhedron procedure might appear cumbersome at first sight, but this facet-based particle tracking nevertheless comes with a number of advantages, also pointed out in [49]. First of all, it is applicable to arbitrary polyhedral meshes, both in two and three spatial dimensions, on the premise that cells are convex. Secondly, by marking the facets on the boundary of the domain, it is conveniently detected if, when and where a particle encounters a certain boundary. Apart from the external boundaries, this feature also comes in convenient for internal boundaries in parallel computations using mesh decomposition. Finally, the fraction of the time step spent in a certain cell is explicitly known in the convex polyhedron method. Although not further pursued in the code in its present form, this can facilitate the updating of the particle velocity along the trajectory [99].

Algorithm 1 Pseudo-code for the convex polyhedron particle tracking for a single particle initially located in a cell K .

```

1:  $k \leftarrow 0$ 
2:  $\Delta t^{(k)} \leftarrow \Delta t$ 
3:  $\mathbf{x}_p^{(k)} \leftarrow \mathbf{x}_p(t)$ 
4: while  $\Delta t^{(k)} > 0$  do
5:    $k \leftarrow k + 1$ 
6:   Time to facet intersection:  $\Delta t_i = b_i / (\mathbf{u}(\mathbf{x}_p, t) \cdot \mathbf{n}_{Ki})$ 
7:   Minimum, yet positive time:  $\Delta t_{i_{\min}} = \min(\max(0, \Delta t_i))$ 
8:   if  $\Delta t_{i_{\min}} > \Delta t^{(k)}$  then
9:     Particle remains in cell  $K$ 
10:     $\mathbf{x}_p^{(k)} \leftarrow \mathbf{x}_p^{(k-1)} + \mathbf{u}(\mathbf{x}_p, t) \cdot \Delta t^{(k)}$ 
11:     $\Delta t^{(k)} \leftarrow 0$ 
12:   else
13:     Push particle to facet
14:     $\mathbf{x}_p^{(k)} \leftarrow \mathbf{x}_p^{(k-1)} + \mathbf{u}(\mathbf{x}_p, t) \cdot \Delta t_{i_{\min}}$ 
15:     $\Delta t^{(k)} = \Delta t^{(k-1)} - \Delta t_{i_{\min}}$ 
16:    if Facet has two neighboring cells then
17:      Update hosting cell index:  $K \leftarrow i$ 
18:    else
19:      Apply boundary condition
20:    end if
21:   end if
22: end while
23:  $\mathbf{x}_p(t + \Delta t) \leftarrow \mathbf{x}_p^{(k)}$ 

```

D.2.4. Boundary conditions at particle level

Apart from enforcing the boundary conditions at the background mesh, modifications at the particle level are also required when a particle encounters a boundary. A boundary at the particle level is encountered if a particle is pushed to a facet with only one neighboring cell, see Line 19 in Algorithm 1. The following particle boundaries are supported in LEOpart, see for details `advect_particles.h`:

C++ code

```

// enum for external facet types
enum class facet_t : std::uint8_t
{
    internal,
    closed,
    open,
    periodic
};

```

The user can mark the different parts of the boundary using a `MeshFunction`, see e.g. the unit tests in `test_2d_advect.py`. The implementation of the different particle boundary conditions are briefly discussed below.

Periodic boundaries

When a particle encounters a periodic boundary, it should reappear at the opposing side of the domain. Facets on a periodic boundary, are matched against facets at the opposing side of the domain. A bi-periodic domain, for instance, can be marked in LEOPart as

Python code

```

from dolfin import (VectorFunctionSpace, Function, RectangleMesh,
                   SubDomain, MeshFunction, Point)
from leopart import (particles, advect_particles)
import numpy as np

class Boundaries(SubDomain):
    def inside(self, x, on_boundary):
        return on_boundary

(xmin, ymin, xmax, ymax) = (0., 0, 1., 1.)

mesh = RectangleMesh(Point(xmin, ymin), Point(xmax, ymax), 10, 10)
V = VectorFunctionSpace(mesh, "CG", 1)
v = Function(V)

# Mark the boundary as periodic
facet_marker = MeshFunction('size_t', mesh, mesh.topology().dim()-1)
facet_marker.set_all(0)
boundaries = Boundaries()
boundaries.mark(facet_marker, 3)

# Specify opposing boundaries, always comes in pairs
lims = np.array([[xmin, xmin, ymin, ymax], [xmax, xmax, ymin, ymax],
                [xmin, xmax, ymin, ymin], [xmin, xmax, ymax, ymax]])

# Locate dummy particle at (0.5, 0.5) and initialize advection class
p = particles(np.array([0.5, 0.5]), [], mesh)
ap = advect_particles(p, V, v, facet_marker, lims.flatten())

```

Full examples of how periodic boundaries can be applied in practice, can be found in `TaylorGreen_2D.py` and `TaylorGreen_3D.py`.

Open boundaries and particle insertion/deletion

At open boundaries particles escape, or enter the domain. When a particle is marked to cross an open boundary facet, it simply is deleted from the list of particles. Inflow boundaries, however, are less straightforward since new particles are to be created. As yet, this is done via the `AddDelete` class, which also allows a user to keep control over the number of particles per cell.

This `AddDelete` takes as its arguments the particle class, a lower and an upper bound for the number of particles per cell, and a list of DOLFIN functions which are to be used for initializing the particle values. If a cell is marked empty, i.e. the number of particles is lower than the preset lower bound for the number of particles per cell, the particle deficit is complemented by creating new particles. To this end, new particle locations in

a cell are determined using a random generator. To initialize the other particle slots, two options are at the user's convenience: the particle value can either be initialized based on a point interpolation from the underlying mesh field, or the particle value is assigned based on rounding-off the interpolated field value to a lower or upper boundary, i.e.

$$\psi_p = \begin{cases} \psi_{\min} & \text{if } \psi_h(\mathbf{x}_p) \leq \frac{\psi_{\min} + \psi_{\max}}{2}, \\ \psi_{\max} & \text{if } \psi_h(\mathbf{x}_p) > \frac{\psi_{\min} + \psi_{\max}}{2}. \end{cases} \quad (\text{D.2})$$

This feature is particularly useful when the particles carry binary fields, such as the density in two-fluid simulations.

A minimal example, demonstrating the LEOPart implementation of the two options for particle insertion/deletion is provided below, with results depicted in Fig. D.3.

Python code

```

from dolfin import (UnitSquareMesh, FunctionSpace, Function,
                    Expression,
                    Point, Constant)
from leopart import (particles, RandomRectangle, AddDelete,
                    assign_particle_values)

msh = UnitSquareMesh(2, 2)
Psi = FunctionSpace(msh, "DG", 1)
psi = Function(Psi)

psi_expression = Expression('x[0]', degree=1)
psi.assign(psi_expression)

(p_min, p_max) = (4, 8)
(psi_min, psi_max) = (0., 1.)

# Initialize particles
xp = RandomRectangle(Point(0., 0.), Point(1., 1.)).generate([1, 1])
psip_smooth = assign_particle_values(xp, psi_expression)
psip_binary = assign_particle_values(xp, psi_expression)

p = particles(xp, [psip_smooth, psip_binary], msh)

# Slot 2, psip_binary initialized using binary values
AD = AddDelete(p, p_min, p_max, [psi, psi], [2], [psi_min, psi_max])
# Sweep over mesh to delete/insert particles
AD.do_sweep()

(xp_n, psip_smooth, psip_binary) = (p.return_property(msh, 0),
                                     p.return_property(msh, 1),
                                     p.return_property(msh, 2))

```

The AddDelete class also allows to keep control over the maximum number of particles per cell, specified via the variable `p_max` in the above presented code. If a cell in the `do_sweep` method is marked to contain more particles than prescribed, the surplus of, say, m particles is removed by deleting m particles which have the shortest distance to another particle in that cell. This procedure ensures that particles are removed evenly from the cell interior.

As a final remark: an upwind initialization of the particle value, i.e. initializing the particle value near open boundaries based on the value at the (inflow) boundary facet, is expected to be a useful feature not yet included in LEOPart.

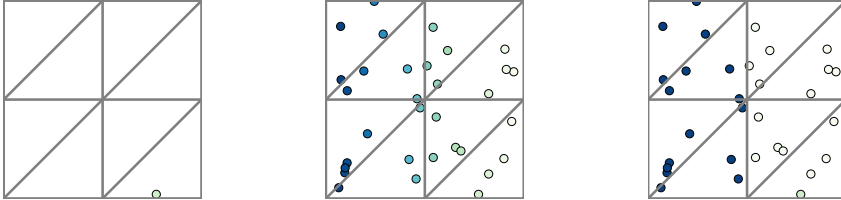


Figure D.3: Particle insertion: initial particle field (left), particle value assignment by interpolation (middle), binary particle value assignment (right).

Closed boundaries

If a particle hits a closed boundary during the time step, the particle is reflected by setting the particle velocity to its reflected value

$$\mathbf{u}_h(\mathbf{x}_p) = \mathbf{u}_h(\mathbf{x}_p) - 2(\mathbf{u}_h(\mathbf{x}_p) \cdot \mathbf{n})\mathbf{n}. \quad (\text{D.3})$$

D.3. Particle-mesh projections

In line with the material presented in the thesis, two options are available for the projection of particle data onto the mesh fields. The simplest of the two is via a (local) ℓ^2 -projection - accessible via the `l2projection` class - or the PDE constrained projection - available via the `PDEStaticCondensation`. For minimal examples for these two projections, the interested reader is referred to, e.g., `test_2d_project.py`.

D.4. Performance tests

The performance of the scheme is assessed for three examples. The rotation of a Gaussian pulse is used to illustrate the performance in terms of wall clock time and scaling for relatively small parallel computations, i.e. 1-24 cores. Secondly, the performance for a high-resolution 2D lock exchange test is assessed. Finally, some preliminary results on a 3D domain are shown, which illustrates that scalable, iterative solvers are a necessity rather than a luxury to make large scale 3D problems computationally tractable.

D.4.1. Rotation of Gaussian pulse

A rotating Gaussian pulse, as described in Section 3.8.1 is considered first. The unit disk is triangulated into 57,856 cells on which 723,564 particles are seeded. Linear polynomials are used in the PDE-constrained particle-mesh projection, yielding a $174,080 \times 174,080$ global system for the PDE-constrained projection after static condensation. Using piecewise constant polynomials for the Lagrange multiplier, a 4×4 matrix is to be inverted at every cell, see Eq. (3.38). Furthermore, particles are advected using the

three-stage Runge-Kutta scheme. In total 400 time steps of size $\Delta t = 5e-3$ are taken. All computations use a direct sparse solver (SuperLU) to solve the global system of equations.

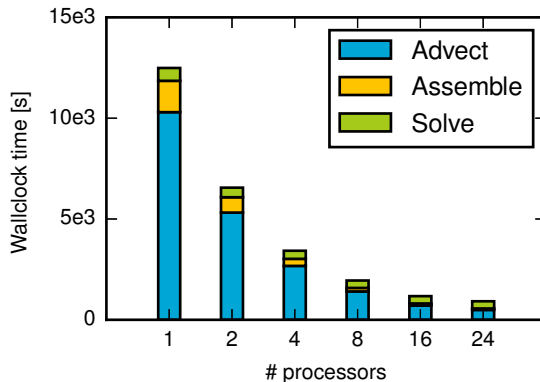


Figure D.4: Rotation of Gaussian pulse: wallclock time in s for the different steps in a scalar advection test on different number of processes.

Wallclock times for running the computation on different number of Intel Xeon CPU E5-2690 v4 processes are presented in Fig. D.4. Using one process, by far the largest part of the computational time (roughly 75%) is consumed on the particle advection. Assembling the global system, which includes the inversion of 57,856 local 4×4 matrices consumes roughly 15-20% of the computational time, and the global solve consumes only 5-10% of the wallclock time. Increasing the number of processes for the same problem - so-called 'strong scaling' - shows that the computational time spent on the advection and the assembling part sharply decreases. The scalability of these steps is attributed to the locality of these operations, that is steps are performed cellwise. Solving the global system for the PDE-constrained projection appears to scale much less, so that for 24 processes, about 40% of the total computational time is consumed by solving the global system for the PDE-constrained projection.

Table D.1: Rotation of Gaussian pulse: speed-up of the different model parts in parallel computations benchmarked against 1 processor run.

	# Processors					
	1	2	4	8	16	24
Particle advection	1	1.9	3.9	7.3	14.6	21.0
Assembling system PDE-map	1	2.0	4.5	8.9	14.5	21.6
Solve system PDE-map	1	1.3	1.6	1.7	1.8	1.8
Total	1	1.9	3.6	6.4	10.6	13.6

The scaling characteristics for this test are further investigated in Table D.1, summarizing the speed-up for the different tests relative to the run on one process. The particle

advection and the assembling step both show perfect linear scaling. This, however, is not so for the direct sparse solver used in the PDE-constrained projection. In terms of scalability, this step thus appears the limiting factor.

D.4.2. Lock exchange

Secondly, timings for a high-resolution run of the lock exchange test as reported in Section 5.5.2 are presented. The domain $\Omega := [0, 30] \times [-0.5, 0.5]$ is triangulated into $2,000 \times 80 \times 2 = 320,000$ regular triangular cells. Using the EDG-HDG method, see Eq. (4.9b), with $k = 1$ for solving the Stokes equations, the number of dofs in the global system amounts to 1,288,322. A direct sparse matrix solver (SuperLU) is used to solve the Stokes system. The total number of particles amounts to 9,408,000, where this number stays constant throughout the computation. Furthermore, 800 time steps of size $\Delta t^* = \Delta t \sqrt{H/g'} = 1.25e-2$ are performed.



Figure D.5: Lock exchange: density field at the mesh at $t^* = 10$ using ℓ^2 or PDE constrained particle-mesh projections. Note the different time scales used for the PDE-constrained projections and the local ℓ^2 -projections.

For the PDE-constrained projection, the global problem yields 964,160 unknowns for the density projection, and 1,928,320 unknowns for the momentum projection. The reason that the global system for the momentum projection is larger than the system for the unsteady Stokes equations, roots in the different facet function spaces. In the momentum projection, these facet functions are *discontinuous* in the vertices, whereas in the EDG-HDG formulation for the Stokes equations the facet functions for the velocity are *continuous* in the vertices, thereby significantly reducing the size of the global system. As an aside, it is remarked that i) this difference will be even more pronounced in 3D, and ii) in relative terms the difference in size will decrease for higher-order polynomial facet functions. The global systems resulting from the PDE-constrained projections are solved using a GMRES solver in conjunction with an algebraic multigrid preconditioner, where this solver/preconditioner pair is used as a black-box. Apart from showing results obtained with the PDE-constrained density and momentum map, the timings for using the local ℓ^2 -projection are reported also. The density projection in this case is rendered bound preserving by imposing box constraints on the local least squares problem. Since the ℓ^2 -projections are performed cell wise, this particle-mesh projection type is expected to be significantly faster than the PDE-constrained projections.

Visually, the mesh density fields at $t^* = 10$ are comparable in terms of the bulk behavior for both projections, Fig. D.5, although differences in the small scale features are observed. No further attempts are made to interpret and value these small scale differences between the local ℓ^2 -projections and the PDE-constrained projections.

Timings are reported in Fig. D.6, using 32, 64 and 128 CPU cores on the Peta 4

supercomputing facility of the University of Cambridge.¹ Peta 4 contains 768 nodes, equipped with 2 Intel Xeon Skylake 6142 16-core processors each.

Results give a clear insight in the performance of the different parts, and indicate which parts of the scheme are critical for obtaining higher performance. Clearly, the computational time is dominated by the global solves for the Stokes system, and the PDE-constrained particle-mesh projections, Fig. D.6a. The advantage of using iterative solvers for the PDE-projections also becomes clear from this figure. Even though the system for the momentum projection is larger than the system for the Stokes projection, the wallclock time for the momentum projection is considerably smaller and appears to possess better scaling compared to the Stokes solve. Therefore, implementing the iterative solver for the Stokes solver proposed in [129] is believed to be an important step for improving the performance, and probably is indispensable when making the step to problems in three spatial dimensions. It is noteworthy to mention that the ℓ^2 particle-mesh projections exhibit excellent scaling properties, on top of their low computational footprint, see Fig. D.6b.

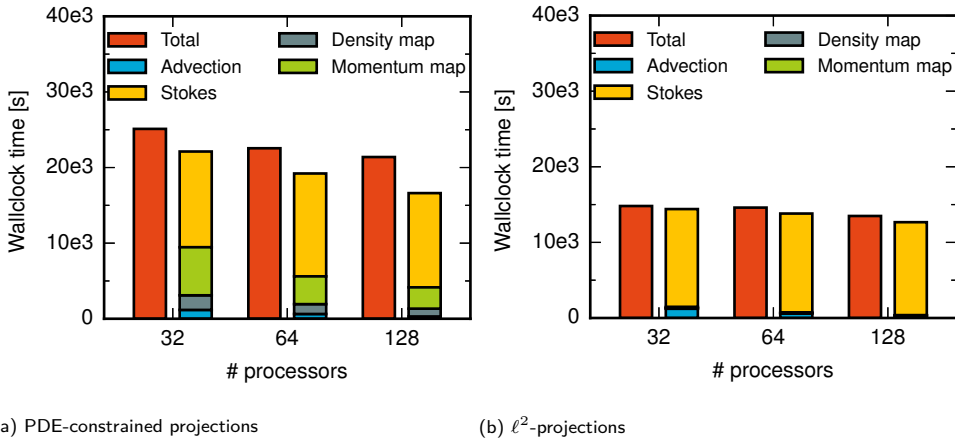


Figure D.6: Lock exchange: strong scaling results.

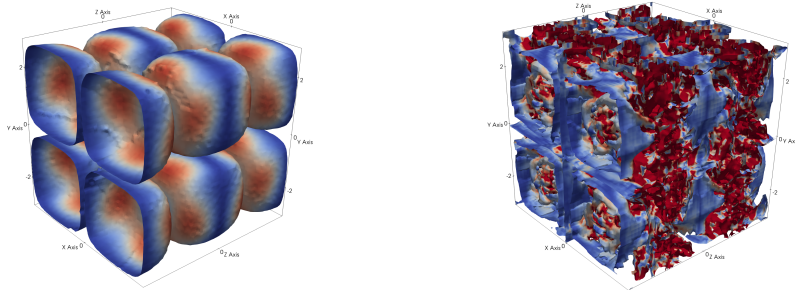
¹Chris Richardson's support in providing access to this resource, is gratefully acknowledged.

Table D.2: Lock exchange test: speed-up of the different model parts in parallel computations benchmarked against 32 processors run.

	PDE-projections			ℓ^2 projections		
	# Processors			# Processors		
	32	64	128	32	64	128
Particle advection	1	1.85	3.87	1	2.06	3.95
Density projection	1	1.48	1.87	1	1.75	3.75
Momentum projection	1	1.73	2.25	1	1.79	3.99
Stokes solve	1	0.93	1.17	1	0.99	1.05
Total	1	1.11	1.17	1	1.01	1.1

D.4.3. Three-dimensional Taylor-Green

As a proof-of-concept, a 3D Taylor-Green flow is considered. This example shows that LEOPart provides the machinery for a 3D implementation, rather than presenting a thorough analysis of the results. The tri-periodic cube with sides of length 2π is considered. The domain is triangulated using $48^3 \times 6 = 663,552$ regular triangular cells, and the number of particles is kept around a value of approximately 20 particles per cell. The kinematic viscosity in this test is set to $1/1600$. The time step is set to $\Delta t = 2.5e-2$, and the end-time of the simulation is set to $T = 10$. Furthermore, the local ℓ^2 projection is used for the projection of the specific momentum in order to reduce the computational costs. The computation is run on 24 Intel Xeon CPU E5-2690 v4 cores.

Figure D.7: Taylor-Green in 3D: $(-0.5, 0.5)$ vorticity in z -direction at $t = 0.125$, and $t = 10$. Color scale indicates the vorticity magnitude, and ranges between 0 and 1.

The -0.25 and 0.25 isosurfaces for the z -component of the vorticity are shown in Fig. D.7, clearly demonstrating the transition to turbulence. The timings for this computation are listed in Table D.3, illustrating that the Stokes solve consumes most of the computational time for this case, whereas the contribution of the particle advection and the field reconstruction using the local ℓ^2 particle-mesh projection to the computational time is very small. This hence again shows that formulating efficient solver(s) for the global system(s) at the mesh - resulting from the discretization of the Stokes system,

and/or the PDE-constrained projections - is of utmost importance to improve the performance and to make the developed HDG/particle-mesh approach an attractive option for an even wider range of practical problems.

Table D.3: Taylor-Green in 3D: overview of wallclock timings for different steps.

Step	Particle advection	ℓ^2 particle-mesh projection	Stokes solve	Total
Time [s]	2.2e3	0.43e3	1.41e5	1.48e5

Bibliography

- [1] B. Cockburn, G. E. Karniadakis, and C.-W. Shu, *The Development of Discontinuous Galerkin Methods*, in *Discontinuous Galerkin Methods* (Springer, Berlin, Heidelberg, 2000) pp. 3–50.
- [2] B. Cockburn, G. Kanschat, and D. Schötzau, *The local discontinuous Galerkin method for linearized incompressible fluid flow: a review*, *Computers & Fluids* **34**, 491 (2005).
- [3] R. J. Labeur and G. N. Wells, *A Galerkin interface stabilisation method for the advection–diffusion and incompressible Navier–Stokes equations*, *Comput. Methods Appl. Mech. Eng.* **196**, 4985 (2007).
- [4] N. Nguyen, J. Peraire, and B. Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations*, *J. Comput. Phys.* **228**, 3232 (2009).
- [5] H. Egger and J. Schöberl, *A hybrid mixed discontinuous Galerkin finite-element method for convection-diffusion problems*, *IMA J. Numer. Anal.* **30**, 1206 (2010).
- [6] R. J. Labeur and G. N. Wells, *Energy stable and momentum conserving hybrid finite element method for the incompressible Navier–Stokes equations*, *SIAM J. Sci. Comput.* **34**, 889 (2012).
- [7] C. Lehrenfeld and J. Schöberl, *High order exactly divergence-free Hybrid Discontinuous Galerkin Methods for unsteady incompressible flows*, *Comput. Methods Appl. Mech. Eng.* **1** **307**, 339 (2016).
- [8] S. Rhebergen and G. N. Wells, *A Hybridizable Discontinuous Galerkin Method for the Navier–Stokes Equations with Pointwise Divergence-Free Velocity Field*, *J. Sci. Comput.* , 1 (2018).
- [9] Y. Xing and C.-W. Shu, *High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms*, *J. Comput. Phys.* **214**, 567 (2006).
- [10] A. N. Brooks and T. J. Hughes, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, *Comput. Methods Appl. Mech. Eng.* **32**, 199 (1982).
- [11] T. J. Hughes, L. P. Franca, and G. M. Hulbert, *A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations*, *Comput. Methods Appl. Mech. Eng.* **73**, 173 (1989).

- [12] T. J. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy, *The variational multi-scale method—a paradigm for computational mechanics*, *Comput. Methods Appl. Mech. Eng.* **166**, 3 (1998).
- [13] R. Liska, M. Shashkov, P. Váchal, and B. Wendroff, *Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods*, *J. Comput. Phys.* **229**, 1467 (2010).
- [14] A. J. Barlow, P.-H. Maire, W. J. Rider, R. N. Rieben, and M. J. Shashkov, *Arbitrary Lagrangian–Eulerian methods for modeling high-speed compressible multimaterial flows*, *J. Comput. Phys.* **322**, 603 (2016).
- [15] R. Labeur, *Finite element modelling of transport and non-hydrostatic flow in environmental fluid mechanics*, Ph.D. thesis, Delft University of Technology (2009).
- [16] J. J. Monaghan, *Smoothed particle hydrodynamics*, *Reports Prog. Phys.* **68**, 1703 (2005).
- [17] S. Koshizuka and Y. Oka, *Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid*, *Nuclear Science and Engineering* **123**, 421 (1996).
- [18] G. A. Dilts, A. Haque, and J. Wallin, *Tuned Local Regression Estimators for the Numerical Solution of Differential Equations*, in *Meshfree Methods for Partial Differential Equations* (Springer, 2003) pp. 87–104.
- [19] P. Suchde, J. Kuhnert, S. Schröder, and A. Klar, *A flux conserving meshfree method for conservation laws*, *Int. J. Numer. Methods Eng.* **112**, 238 (2017).
- [20] P. Randles and L. Libersky, *Smoothed Particle Hydrodynamics: Some recent improvements and applications*, *Comput. Methods Appl. Mech. Eng.* **139**, 375 (1996).
- [21] S. Li and W. K. Liu, *Meshfree and particle methods and their applications*, *Appl. Mech. Rev.* **55**, 1 (2002).
- [22] M. Evans, F. Harlow, and E. Bromberg, *The particle-in-cell method for hydrodynamic calculations*, *Tech. Rep.* (Los Alamos Scientific Laboratory, 1957).
- [23] F. H. Harlow and J. E. Welch, *Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface*, *Phys. Fluids* **8**, 2182 (1965).
- [24] J. Brackbill and H. Ruppel, *FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions*, *J. Comput. Phys.* **65**, 314 (1986).
- [25] Y. Zhu and R. Bridson, *Animating sand as a fluid*, *ACM Trans. Graph.* **24**, 965 (2005).
- [26] D. M. Kelly, Q. Chen, and J. Zang, *PICIN: a particle-in-cell solver for incompressible free surface flows with two-way fluid-solid coupling*, *SIAM J. Sci. Comput.* **37**, 403 (2015).

- [27] Q. Chen, J. Zang, A. S. Dimakopoulos, D. M. Kelly, and C. J. Williams, *A Cartesian cut cell based two-way strong fluid–solid coupling algorithm for 2D floating bodies*, *Journal of Fluids and Structures* **62**, 252 (2016).
- [28] J. Maljaars, R. J. Labeur, M. Möller, and W. Uijtewaal, *A Numerical Wave Tank Using a Hybrid Particle-mesh Approach*, *Proc. Eng.* **175**, 21 (2017).
- [29] J. Maljaars, R. J. Labeur, M. Möller, and W. Uijtewaal, *Development of a hybrid particle-mesh method for simulating free-surface flows*, *J. Hydrodyn. Ser. B* **29**, 413 (2017).
- [30] D. Sulsky, Z. Chen, and H. Schreyer, *A particle method for history-dependent materials*, *Comput. Methods Appl. Mech. Eng.* **118**, 179 (1994).
- [31] S. Kularathna and K. Soga, *Implicit formulation of material point method for analysis of incompressible materials*, *Comput. Methods Appl. Mech. Eng.* **313**, 673 (2017).
- [32] F. Zhang, X. Zhang, K. Y. Sze, Y. Lian, and Y. Liu, *Incompressible material point method for free surface flow*, *J. Comput. Phys.* **330**, 92 (2017).
- [33] R. Löhner and J. Ambrosiano, *A vectorized particle tracer for unstructured grids*, *J. Comput. Phys.* **91**, 22 (1990).
- [34] Z. Chen, Z. Zong, M. B. Liu, and H. T. Li, *A comparative study of truly incompressible and weakly compressible SPH methods for free surface incompressible flows*, *Int. J. Numer. Methods Fluids* **73**, 625 (2013).
- [35] R. Garg, C. Narayanan, D. Lakehal, and S. Subramaniam, *Accurate numerical estimation of interphase momentum transfer in Lagrangian-Eulerian simulations of dispersed two-phase flows*, *Int. J. Multiph. Flow* **33**, 1337 (2007).
- [36] G. Jacobs and J. Hesthaven, *High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids*, *J. Comput. Phys.* **214**, 96 (2006).
- [37] D. Burgess, D. Sulsky, and J. Brackbill, *Mass matrix formulation of the FLIP particle-in-cell method*, *J. Comput. Phys.* **103**, 1 (1992).
- [38] E. Love and D. Sulsky, *An unconditionally stable, energy–momentum consistent implementation of the material-point method*, *Comput. Methods Appl. Mech. Eng.* **195**, 3903 (2006).
- [39] C. Jiang, C. Schroeder, and J. Teran, *An angular momentum conserving affine-particle-in-cell method*, *J. Comput. Phys.* **338**, 137 (2017).
- [40] D. Sulsky and M. Gong, *Improving the Material-Point Method*, in *Innovative Numerical Approaches for Multi-Field and Multi-Scale Problems* (Springer International Publishing, 2016) pp. 217–240.
- [41] E. Edwards and R. Bridson, *A high-order accurate particle-in-cell method*, *Int. J. Numer. Methods Eng.* **90**, 1073 (2012).

- [42] R. Xu, P. Stansby, and D. Laurence, *Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach*, J. Comput. Phys. **228**, 6703 (2009).
- [43] R. Vacondio, B. Rogers, P. Stansby, P. Mignosa, and J. Feldman, *Variable resolution for SPH: A dynamic particle coalescing and splitting scheme*, Comput. Methods Appl. Mech. Eng. **256**, 132 (2013).
- [44] S. Pope, *PDF methods for turbulent reactive flows*, Prog. Energy Combust. Sci. **11**, 119 (1985).
- [45] P. Jenny, S. Pope, M. Muradoglu, and D. Caughey, *A Hybrid Algorithm for the Joint PDF Equation of Turbulent Reactive Flows*, J. Comput. Phys. **166**, 218 (2001).
- [46] R. McDermott and S. Pope, *The parabolic edge reconstruction method (PERM) for Lagrangian particle advection*, J. Comput. Phys. **227**, 5447 (2008).
- [47] Y. Zhang and D. Haworth, *A general mass consistency algorithm for hybrid particle/finite-volume PDF methods*, J. Comput. Phys. **194**, 156 (2004).
- [48] R. M. Nestor, M. Basa, M. Lastiwka, and N. J. Quinlan, *Extension of the finite volume particle method to viscous flow*, J. Comput. Phys. **228**, 1733 (2009).
- [49] D. Haworth, *Progress in probability density function methods for turbulent reacting flows*, Prog. Energy Combust. Sci. **36**, 168 (2010).
- [50] A. D. Beck, P. Ortwein, P. Kopper, N. Kraiss, D. Kempf, and C. Koch, *Towards high-fidelity erosion prediction: On time-accurate particle tracking in turbomachinery*, (2018).
- [51] A. C. Bagtzoglou, D. E. Dougherty, and A. F. B. Tompson, *Application of particle methods to reliable identification of groundwater pollution sources*, Water Resour. Manag. **6**, 15 (1992).
- [52] E. Hathway, C. Noakes, P. Sleight, and L. Fletcher, *CFD simulation of airborne pathogen transport due to human activities*, Build. Environ. **46**, 2500 (2011).
- [53] O. Zienkiewicz and R. Taylor, *The finite element method-basic formulation and linear problems*, MacGraw-Hill, Maidenhead MATH (1989).
- [54] P. M. Gresho and R. L. Sani, *Incompressible flow and the finite element method. volume 1: Advection-diffusion and isothermal laminar flow*, (1998).
- [55] J. Donea and A. Huerta, *Finite element methods for flow problems* (John Wiley & Sons, 2003).
- [56] J. N. Reddy, *Energy principles and variational methods in applied mechanics* (John Wiley & Sons, 2017).

- [57] R. Témam, *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires*, *Archive for Rational Mechanics and Analysis* **32**, 135 (1969).
- [58] G. I. Marchuk, *Splitting and alternating direction methods*, *Handbook of Numerical Analysis* **1**, 197 (1990).
- [59] J. L. Guermond, P. Mineev, and J. Shen, *An overview of projection methods for incompressible flows*, *Comput. Methods Appl. Mech. Eng.* **195**, 6011 (2006).
- [60] J. G. Liu, J. Liu, and R. L. Pego, *Stable and accurate pressure approximation for unsteady incompressible viscous flow*, *J. Comput. Phys.* **229**, 3428 (2010).
- [61] R. Glowinski, S. J. Osher, and W. Yin, *Splitting Methods in Communication, Imaging, Science, and Engineering* (Springer, 2017).
- [62] S. G. Bardenhagen and E. M. Kober, *The generalized interpolation material point method*, *CMES* **5**, 477 (2004).
- [63] M. Steffen, R. M. Kirby, and M. Berzins, *Analysis and reduction of quadrature errors in the material point method (MPM)*, *Int. J. Numer. Methods Eng.* **76**, 922 (2008).
- [64] P. Wallstedt and J. Guilkey, *An evaluation of explicit time integration schemes for use with the generalized interpolation material point method*, *J. Comput. Phys.* **227**, 9628 (2008).
- [65] R. Tielen, E. Wobbes, M. Möller, and L. Beuth, *A high order material point method*, *Procedia Eng.* **175**, 265 (2017).
- [66] S. R. Idelsohn, N. M. Nigro, J. M. Gimenez, R. Rossi, and J. M. Marti, *A fast and accurate method to solve the incompressible Navier-Stokes equations*, *Eng. Comput.* **30**, 197 (2013).
- [67] R. Courant *et al.*, *Variational methods for the solution of problems of equilibrium and vibrations*, *Bulletin of the American Mathematical Society* **49**, 1 (1943).
- [68] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, *Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems*, *SIAM J. Numer. Anal.* **39**, 1749 (2002).
- [69] S. Güzey, B. Cockburn, and H. K. Stolarski, *The embedded discontinuous Galerkin method: application to linear shell problems*, *Int. J. Numer. Methods Eng.* **70**, 757 (2007).
- [70] B. Cockburn, G. Kanschat, and D. Schötzau, *A locally conservative LDG method for the incompressible Navier-Stokes equations*, *Math. Comput.* **74**, 1067 (2004).
- [71] V. John, A. Linke, C. Merdon, M. Neilan, and L. G. Rebholz, *On the Divergence Constraint in Mixed Finite Element Methods for Incompressible Flows*, *SIAM Rev.* **59**, 492 (2017).

- [72] S. Rhebergen and G. N. Wells, *An embedded–hybridized discontinuous Galerkin finite element method for the Stokes equations*, (2018), arXiv:1811.09194 .
- [73] J. T. Oden, *Historical comments on finite elements*, *A History of Scientific Computing* , 152 (1990).
- [74] M. J. Gander and G. Wanner, *From Euler, Ritz, and Galerkin to Modern Computing*, *SIAM Rev.* **54**, 627 (2012).
- [75] K. Olgaard, *Automated computational modelling for complicated partial differential equations*, Ph.D. thesis, Delft University of Technology (2013).
- [76] J. M. Maljaars, R. J. Labeur, N. Trask, and D. Sulsky, *Optimization Based Particle-Mesh Algorithm for High-Order and Conservative Scalar Transport*, in *Numerical Methods for Flows - FEF 2017 Selected Contributions*, *Lecture Notes in Computational Science and Engineering.*, Vol. 132, edited by A. Corsini, S. Perotto, H. E. van Brummelen, and G. Rozza (Springer, 2019) Chap. 23.
- [77] J. M. Maljaars, R. J. Labeur, N. Trask, and D. Sulsky, *Conservative, high-order particle-mesh scheme with applications to advection-dominated flows*, *Comput. Methods Appl. Mech. Eng.* **348**, 443 (2019).
- [78] H. Wendland, *Scattered data approximation*, Vol. 17 (Cambridge University Press, 2004).
- [79] D. Goldfarb and A. Idnani, *A numerically stable dual method for solving strictly convex quadratic programs*, *Math. Program.* **27**, 1 (1983).
- [80] A. Ralston, *Runge-Kutta Methods with minimum error bounds*, *Math. Comput.* **16**, 431 (1962).
- [81] A. York, D. Sulsky, and H. Schreyer, *Fluid–membrane interaction based on the material point method*, *Int. J. Numer. Methods Eng.* , 901 (2000).
- [82] J. Maljaars, *A hybrid particle-mesh method for simulating free surface flows*, (2016).
- [83] P. Thomas and C. Lombard, *Geometric conservation law and its application to flow computations on moving grids*, *AIAA journal* **17**, 1030 (1979).
- [84] R. J. Labeur and G. N. Wells, *Interface stabilised finite element method for moving domains and free surface flows*, *Comput. Methods Appl. Mech. Eng.* **198**, 615 (2009).
- [85] T. J. Hughes, W. K. Liu, and T. K. Zimmermann, *Lagrangian-Eulerian finite element formulation for incompressible viscous flows*, *Comput. Methods Appl. Mech. Eng.* **29**, 329 (1981).
- [86] H. Guillard and C. Farhat, *On the significance of the geometric conservation law for flow computations on moving meshes*, *Comput. Methods Appl. Mech. Eng.* **190**, 1467 (2000).

- [87] M. Lesoinne and C. Farhat, *Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations*, *Comput. Methods Appl. Mech. Eng.* **134**, 71 (1996).
- [88] C. Farhat, P. Geuzaine, and C. Grandmont, *The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids*, *J. Comput. Phys.* **174**, 669 (2001).
- [89] G. N. Wells, *Analysis of an Interface Stabilized Finite Element Method: The Advection-Diffusion-Reaction Equation*, *SIAM J. Numer. Anal.* **49**, 87 (2011).
- [90] A. Logg, K.-A. Mardal, and G. N. Wells, *Automated Solution of Differential Equations by the Finite Element Method*, Vol. 84 (Springer Science & Business Media, 2012) p. 724.
- [91] R. J. LeVeque, *High-Resolution Conservative Algorithms for Advection in Incompressible Flow*, *SIAM J. Numer. Anal.* **33**, 627 (1996).
- [92] P. Bochev, D. Ridzal, and K. Peterson, *Optimization-based remap and transport: A divide and conquer strategy for feature-preserving discretizations*, *J. Comput. Phys.* **257**, 1113 (2013).
- [93] D. Kuzmin, S. Basting, and J. N. Shadid, *Linearity-preserving monotone local projection stabilization schemes for continuous finite elements*, *Comput. Methods Appl. Mech. Eng.* **322**, 23 (2017).
- [94] M. ten Eikelder and I. Akkerman, *Variation entropy: a continuous local generalization of the TVD property using entropy principles*, (2018), arXiv:1808.06447 .
- [95] J. M. Maljaars, R. J. Labeur, and M. Möller, *A hybridized discontinuous Galerkin framework for high-order particle-mesh operator splitting of the incompressible Navier–Stokes equations*, *J. Comput. Phys.* **358**, 150 (2018).
- [96] N. Nguyen, J. Peraire, and B. Cockburn, *An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations*, *J. Comput. Phys.* **230**, 1147 (2011).
- [97] S. Rhebergen and G. N. Wells, *Analysis of a Hybridized/Interface Stabilized Finite Element Method for the Stokes Equations*, *SIAM J. Numer. Anal.* **55**, 1982 (2017).
- [98] L. D. G. Sigalotti, J. Klapp, E. Sira, Y. Meleán, and A. Hasmy, *SPH simulations of time-dependent Poiseuille flow at low Reynolds numbers*, *J. Comput. Phys.* **191**, 622 (2003).
- [99] P. P. Popov, R. McDermott, and S. B. Pope, *An accurate time advancement algorithm for particle tracking*, *J. Comput. Phys.* **227**, 8792 (2008).
- [100] J.-P. Hansen and I. R. McDonald, *Theory of simple liquids* (Elsevier, 1990).

- [101] S. Litvinov, X. Hu, and N. Adams, *Towards consistence and convergence of conservative SPH approximations*, *J. Comput. Phys.* **301**, 394 (2015).
- [102] P. Visser, S. Luding, and S. Srivastava, *Advanced programming in engineering lecture notes*, (2011).
- [103] H. J. H. Clercx and G. J. F. van Heijst, *Two-Dimensional Navier-Stokes Turbulence in Bounded Domains*, *Applied Mechanics Reviews* **62**, 020802 (2009).
- [104] P. Tabeling, *Two-dimensional turbulence: a physicist approach*, *Phys. Rep.* **362**, 1 (2002).
- [105] R. Kraichnan, *Inertial Ranges in Two-Dimensional Turbulence*, *Phys. Fluids* **10**, 1417 (1967).
- [106] R. Issa, E. S. Lee, D. Violeau, and D. R. Laurence, *Incompressible separated flows simulations with the smoothed particle hydrodynamics gridless method*, *Int. J. Numer. Methods Fluids* **47**, 1101 (2005).
- [107] B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schönung, *Experimental and theoretical investigation of backward-facing step flow*, *J. Fluid Mech.* **127**, 473 (1983).
- [108] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations*, *J. Comput. Phys.* **59**, 308 (1985).
- [109] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher, *Benchmark Computations of Laminar Flow Around a Cylinder*, in *Flow Simulation with High-Performance Computers II* (Springer, 1996) pp. 547–566.
- [110] J. Chessa and T. Belytschko, *An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension*, *Int. J. Numer. Methods Eng.* **58**, 2041 (2003).
- [111] S. Claus and P. Kerfriden, *A CutFEM method for two-phase flow problems*, *Comput. Methods Appl. Mech. Eng.* **348**, 185 (2019).
- [112] D. Schillinger and M. Ruess, *The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models*, *Arch. Comput. Methods Eng.* **22**, 391 (2015).
- [113] F. de Prenter, C. Verhoosel, G. van Zwieten, and E. van Brummelen, *Condition number analysis and preconditioning of the finite cell method*, *Comput. Methods Appl. Mech. Eng.* **316**, 297 (2017).
- [114] C. Hirt and B. Nichols, *Volume of fluid (VOF) method for the dynamics of free boundaries*, *J. Comput. Phys.* **39**, 201 (1981).
- [115] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, *J. Comput. Phys.* **79**, 12 (1988).

- [116] E. Olsson and G. Kreiss, *A conservative level set method for two phase flow*, J. Comput. Phys. **210**, 225 (2005).
- [117] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, J. Comput. Phys. **100**, 335 (1992).
- [118] J. Maddison, D. Marshall, C. Pain, and M. Piggott, *Accurate representation of geostrophic and hydrostatic balance in unstructured mesh finite element ocean modelling*, Ocean Model. **39**, 248 (2011).
- [119] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska, *Quantitative benchmark computations of two-dimensional bubble dynamics*, Int. J. Numer. Methods Fluids **60**, 1259 (2009).
- [120] R. J. Lowe, J. W. Rottman, and P. F. Linden, *The non-Boussinesq lock-exchange problem. Part 1. Theory and experiments*, J. Fluid Mech. **537**, 101 (2005).
- [121] V. K. Birman, J. E. Martin, and E. Meiburg, *The non-Boussinesq lock-exchange problem. Part 2. High-resolution simulations*, J. Fluid Mech. **537**, 125 (2005).
- [122] G. Wu and R. Eatock Taylor, *Finite element analysis of two-dimensional non-linear transient water waves*, Appl. Ocean Res. **16**, 363 (1994).
- [123] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities*, Int. J. Numer. Methods Eng. **79**, 1309 (2009).
- [124] R. G. Dean and R. A. Dalrymple, *Water wave mechanics for engineers and scientists*, Vol. 2 (World Scientific Publishing Company, 1991).
- [125] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias, *Experimental investigation of dynamic pressure loads during dam break*, J. Fluids Struct. **48**, 407 (2014).
- [126] S.-C. Hsiao and T.-C. Lin, *Tsunami-like solitary waves impinging and overtopping an impermeable seawall: Experiment and RANS modeling*, Coast. Eng. **57**, 1 (2010).
- [127] D. G. Goring, *Tsunamis - The propagation of long waves onto a shelf*, Ph.D. thesis, California Institute of Technology (1978).
- [128] G. E. Karniadakis, M. Israeli, and S. A. Orszag, *High-order splitting methods for the incompressible Navier-Stokes equations*, J. Comput. Phys. **97**, 414 (1991).
- [129] S. Rhebergen and G. N. Wells, *Preconditioning of a hybridized discontinuous Galerkin finite element method for the Stokes equations*, J. Sci. Comput. **77**, 1936 (2018).
- [130] T. Belytschko, Y. Y. Lu, and L. Gu, *Element-free Galerkin methods*, Int. J. Numer. Methods Eng. **37**, 229 (1994).
- [131] S. Popinet, *Numerical Models of Surface Tension*, Annu. Rev. Fluid Mech. **50**, annurev (2018), 1507.05135 .

Acknowledgements

Doing a PhD bears many resemblances with a journey to a somewhat 'ill-posed' destination. And for sure, if the destination is unknown, the path thither cannot be known beforehand either, let alone that it can be simply looked up in a travel guide. When overseeing the research in retrospect, it is therefore not surprising that the traveled path shows many funny bends, weird jumps, and went through a number of valleys, and yes, even crossed a couple of beautiful peaks. In fact, the mere notion that I am now able to oversee the path which is behind, implies enough in this respect.

Reaching this point, however, would not have been possible without the help of many people that I have met during the PhD, or who accompanied me throughout this journey. In particular, I am much indebted to my promotor Wim and daily supervisor Robert Jan. Thanks Wim, for pushing me always to take that one extra step, and giving me the opportunity and freedom to steer my research. To do this in a sensible way, however, wouldn't have been possible without the help of the daily supervisor. Robert Jan, your critical opinion, thorough insights, and patient support were indispensable in lifting this research to a higher level.

Though stumbling upon stones of 'scientific selfishness' several times during the journey, these moments were by far exceeded by the amount of 'scientific solidarity' I have received in the past years. Particularly, I want to thank Deborah Sulsky for hosting me at the University of New Mexico and bringing me in contact with people from Sandia National Labs. I couldn't foresee beforehand that I would spend a month of my research time working on flow-related problems in one of the most arid parts of the USA. Moreover, after about 2.5 years, it was a great experience to meet other people on the globe who were interested in the same type of problems. Also, I want to thank Chris Richardson from the BP Institute. Thanks for hosting me in Cambridge, giving me a crash-course in professional software development, and providing ample computational resources to push the capabilities of the developed tools to the limit. Closer to home, I would like to thank Matthias Möller for the many valuable discussions particularly during the early stages of the project.

Certainly one of the sad consequences of this PhD journey coming to an end, is to leave the very inspiring environment of the WaterLab. The discussions on research related and non-research related topics just added that extra bit of color to a working day. Moreover, the empathy and friendship received from the lab-colleagues during the various life changing events such as our wedding and more lately the birth of our new born, has been really, really much appreciated!

Mijn familie en schoonfamilie, die altijd als een warme deken om me heen zijn geweest, wil ik bedanken voor hun onvoorwaardelijke steun in de afgelopen jaren. In het bijzonder geldt dit wel voor mijn lieve vrouw Gerdine. Ook jouw bijdrage aan dit proefschrift valt moeilijk te overschatten. Al was het maar vanwege je impliciete tijdsinvestering op alle momenten dat je echtgenoot meende dat de laptop opengeklapt moest worden. Ondanks

dat, vond en vind ik bij jou altijd een luisterend oor om tegenslagen, frustraties maar ook successen te delen.

Boven alles dank ik God die mij de gezondheid, het verstand, en vele andere zegeningen gegeven heeft om dit proefschrift te voltooien. Immers, *'in Hem leven wij en bewegen ons en zijn wij'* (Handelingen 17:28/Acts 17:28). In deze wetenschap past het om Hem alleen de eer te geven. Soli Deo Gloria.

Curriculum Vitæ

Jakob Marinus Maljaars

11-12-1991 Born in Vlissingen, The Netherlands.

Education

2004–2010 VWO
Calvijn College, Goes (NL)

2010–2013 BSc in Civil Engineering (cum laude)
Delft University of Technology, The Netherlands

2013–2016 MSc in Civil Engineering (cum laude)
Delft University of Technology, The Netherlands
Specialisation: Excellence track in fluid and solid mechanics

2015–2019 PhD Delft University of Technology
Delft University of Technology, The Netherlands
Thesis: When Euler meets Lagrange - Particle-mesh modeling of
advection dominated flows
Promotor: Prof. dr. ir. W.S.J. Uijtewaal
Co-promotor: Dr. ir. R.J. Labeur

Grants and Awards

2013 Topsector Water study grant
2014 UfD-Damen bachelor award
2015 NWO research grant, awarded through the
JMBC-EM Graduate Programme on Solid & Fluid mechanics

List of Publications

Journal Publications

- [4] J.M. Maljaars, C.N. Richardson, N. Sime, *LEoPart: a particle library for FEniCS*, Computers & Mathematics with Applications (submitted).
- [3] J.M. Maljaars, R.J. Labeur, N.A. Trask, D. Sulsky, *Conservative, high-order particle-mesh scheme with applications to advection-dominated flows*, Computer Methods in Applied Mechanics and Engineering 348 (2019) 443-465.
- [2] J.M. Maljaars, R.J. Labeur, M. Möller, *A hybridized discontinuous Galerkin framework for high-order particle-mesh operator splitting of the incompressible Navier-Stokes equations*, Journal of Computational Physics 358 (2018) 150–172.
- [1] J.M. Maljaars, R.J. Labeur, M. Möller, *Development of a hybrid particle-mesh method for simulating free-surface flows*, Journal of Hydrodynamics, Ser. B 29 (2017) 413–422.

Book Chapters

- [1] J.M. Maljaars, R.J. Labeur, N. Trask, and D. Sulsky, *Optimization Based Particle-Mesh Algorithm for High-Order and Conservative Scalar Transport*, in *Numerical Methods for Flows - FEF 2017 Selected Contributions*, Lecture Notes in Computational Science and Engineering., Vol. 132, edited by A. Corsini, S. Perotto, H. E. van Brummelen, and G. Rozza (Springer, 2019) Chap. 23.

Conference Contributions

- [5] J. M. Maljaars, C. N. Richardson, R. J. Labeur, LEOPART: a particle add-on for FEniCS, *FEniCS Conference 2019*, 12-14 June 2019, Washington DC, USA.
- [4] J. M. Maljaars, R. J. Labeur, N. A. Trask, D. Sulsky, A Hybridized Discontinuous Galerkin Framework for High-Order Conservative Particle-Mesh Methods, *13th World Congress on Computational Mechanics*, 23-27 July 2018, New York City, USA.
- [3] J. M. Maljaars, R. J. Labeur, A conservative particle-mesh projection using PDE-constraints, *FEniCS Conference 2018*, 21-23 March 2018, Oxford, UK.
- [2] J. M. Maljaars, R. J. Labeur, M. Möller, W. S. J. Uijtewaal, A Numerical Wave Tank Using a Hybrid Particle-mesh Approach, *1st International Conference on the Material Point Method*, MPM conference 2017, 10 – 12 January 2017, Delft, The Netherlands.

- [1] J. M. Maljaars, R. J. Labeur, M. Möller, W. S. J. Uijttewaal, A hybrid particle-mesh method for simulating free-surface flows, *FEniCS Conference 2016*, 18-20 May 2016, Oslo, Norway.

Other Scientific Contributions

- [4] J. M. Maljaars, R. J. Labeur, A particle-mesh scheme for (multiphase) incompressible flow with local conservation properties, presentation at *BP Institute Seminar Series*, November 1, Cambridge, UK.
- [3] J.M. Maljaars, Imagine, Euler and Lagrange meet nowadays..., selected student pitch & poster at *WCCM 2018*, 23-27 July 2018, New York City, USA.
- [2] J. M. Maljaars, R. J. Labeur, M. Möller, A High-Order Particle-Mesh Method for the Incompressible Navier-Stokes Equations, Poster at *FEniCS Conference 2017*, 12-14 June 2017, Luxembourg.
- [1] J. M. Maljaars, R. J. Labeur, Simulating (nearly) incompressible flows with particle-mesh methods - Practical Insights and Best Practices, *Flow Slides Symposium*, 21 February 2018, Delft, The Netherlands.