

Chapter 4

Models and Applications

Abstract In this section, we discuss the different multicomponent and multiscale models, which are later applied in simulations. We focus on the coupling of microscopic and macroscopic models, while the microscopic model is related on finer spatial and time scales and the macroscopic model is related to the coarser spatial and time scales. We discuss exemplary engineering problems in the field of electronic application and transport reaction applications in Plasma models. Here, the models and their underlying multiscale and multicomponent methods are discussed. Based on the aligned methods, we see the data flow between the disparate scales and can estimate the accuracy in each micro- and macroscopic model, such that we obtained truly working multiscale and multicomponent approaches.

We deal with the following characterization based on the different spatial and time scales of the models, where we decompose the models into the following, see [1]:

- Microscopic Models: Multicomponent Kinetics (discrete treatment) and
- Macroscopic Models: Multicomponent Fluids (continuous treatment).

Further, we deal with multiscale models, which covered the different microscopic and macroscopic scales and applied methods to overcome the large-scale differences, see [2].

Remark 4.1 We concentrate on multiscale models, which describes different models—e.g. a microscopic and macroscopic model—and also only macroscopic models but with embedded microscopic scales to resolve material properties—e.g. electromagnetic behaviour of a magnetizable fluid, see [3, 4].

4.1 Multicomponent Fluids

Abstract In this section, we discuss the models and applications based on the different multicomponent fluid models. Here, we assume to have a macroscopic scale, i.e. we can upscale the microscopic behaviour into the macroscopic scales. We deal with a continuum description and discuss some models based on the multicomponent

fluid problems. Here, standard splitting and multiscale methods are modified with respect to the requirements of the applications. Then, we can close the gap between pure theoretical treatment of numerical methods and their numerical analysis and the necessary adaptation of such standard numerical schemes to engineering applications with the relation to the model problems.

4.1.1 Multicomponent Transport Model for Atmospheric Plasma: Modelling, Simulation and Application

4.1.1.1 Introduction

In the following, we discuss a multicomponent transport model for atmospheric (normal pressure) plasma applications.

In such models, it is important to take into account the mixture of the plasma species.

We are motivated to understand atmospheric plasmas within non-thermal equilibrium, which are applied in etching, deposition and sterilization applications, see [5, 6]), and further in emission filtering processes.

We deal with weakly ionized gas mixtures and chemical reactions in room temperature. Each behaviour of a single species and the mixture is complex and needs additional mixture terms that extend the standard models, see [7–10].

Furthermore, the motivation arose of different applications in the so-called jet stream plasma apparatus, for example [11–13]. In such applications, the understanding of the flow and reaction of the species are important.

We assume to deal with a modelling in a time- and spatial- scale, which we can decompose into heavy particles (molecules, atoms, ions) and light particles (electrons)—i.e. we have $Kn \ll 1.0$ where Kn , Knudsen number, is the ratio of the molecular mean free path length to a representative physical length scale—e.g. length of the apparatus, and therefore, we can apply a macroscopic model.

In the following, we discuss the so-called macroscopic models, also called fluid models, for the plasma model, which is discussed in [14, 15].

We present the special models with respect to their benefits, starting from a two-component fluid model till a multicomponent fluid model with Stefan–Maxwell equation for the mixture of the species. With such a complex model, we achieve an optimal mixture model, which represents the individual single heavy particle.

The underlying conservation laws result in the equations of mass, momentum and energy and additional with conditions related to the Stefan–Maxwell equation, e.g. summation of the mass rates is 1 ($\sum_{i=1} w_i = 1$) and summation of the mass fluxes is 0 ($\sum_{i=1} j_i = 0$).

Such equations with additional conditions are quasilinear, strong coupled parabolic differential equations, see [16].

Such equations need a larger computational amount based on the nonlinearities in the diffusion part. Standard models, based on the Fickian's approach, compared the ideas in [17], are much more simpler to solve and the extended model has taken into account singularities and nonlinear behaviours, see [16, 18, 19].

In the following, we discuss step-by-step approach of the novel models and the development of the underlying solver methods.

4.1.1.2 Introduction and Overview

Since recent years, the application in normal pressure plasmas arose important and therefore the understanding of the reactive chemical species in the plasma and during its mixture is necessary. For such delicate problems, the standard models which are known in the literature have to be extended by the reactive parts of the mixture. Such an important detail can be modelled by the diffusion operator, and the Stefan–Maxwell equation is a possibility to take into account such mixture behaviours, see [16].

In such reactive plasmas, we obtain due to the typical known processes, as ionization and collision, and additional processes, the so-called chemical reactions.

Such chemical processes are dominant for normal pressure plasmas and they are used in the plasma medicine technology.

While they applied air as a plasma background, we have the highly reactive elements oxygen O^2 and nitrogen N^2 in the complex gas mixtures.

Therefore, it is important to extend the standard modelling and simulation techniques, see [20, 21], and embed the nonlinear structures of the Stefan–Maxwell approach.

The diffusive processes are modelled by the so-called *multicomponent diffusion*, which are more and more studied in the Stefan–Maxwell approaches in fluid-dynamical models, see [22].

We obtain an improvement of the so-called binary diffusion processes in the transport reaction models, if we have no dominant species, e.g. only minor at species, which means we do not have a dominant background matrix. Such observations made it necessary to deal with a more detailed modelling, see [17, 23–25].

In comparison to pure fluid-dynamical models, see porous media models [26] or elementary modelling [27], or so-called neutral fluids, in macroscopical plasma models, we have additional terms, for example electric fields. We assume additional to deal with weak-ionized particles that such weak-ionized heavy particles can be modelled by a multicomponent fluid model, vgl. [14].

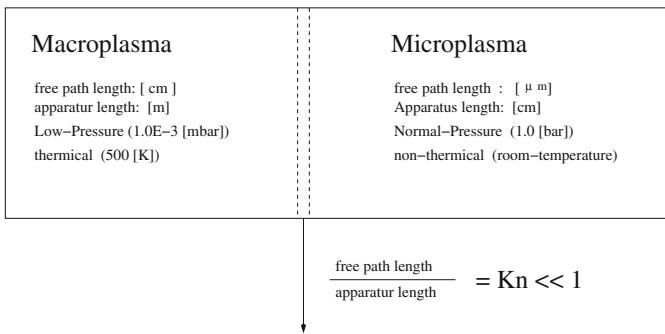
In modelling plasmas, we deal with a so-called scaling, which allows to distinguish between macroscopic plasma models and microscopic plasma models, confer Table 4.1 and Fig. 4.1.

Table 4.1 Parameters for the macro- and microplasma and their applications

Mean free path length electrons	Pressure	Temperature	Length of the reactor	
			Plasma	Neutral gas
<i>MacroPlasma (CCP, ICP: Etching and Deposition)</i>				
0.01–1 (cm)	1–100 (Pa)	300–500 (K)	≈10 (cm)	≈100 (cm)
<i>MicroPlasma (Plasmajets, DBD: Deposition and Sterilization)</i>				
1 (μm)	10 ⁵ (Pa)	300–500 (K)	0.1–1 (mm)	1–10 (cm)

Both plasmas have the same characteristics in the Knudsen number, i.e. $Kn \ll 1$ and can be treated and simulated as macroscopic models

Multicomponent–transport–model (Fluid–model)



Macroscopic Model:

Multi–component–Transport–Model with Stefan–Maxwell Approach

Fig. 4.1 Macroscopic plasma models

We discuss the following steps in the next sections:

- In Sect. 4.1.1.3, we discuss the derivation of the multicomponent transport models. We begin with a simple model (two-component fluid model) and end up with a delicate multicomponent transport model (multifluid flow model).
- In Sect. 4.1.1.4, we discuss the mathematical classification and the numerical treatment of such delicate transport models with embedded Stefan–Maxwell approximations.
- The conclusions are discussed in Sect. 4.1.1.5.

4.1.1.3 Discussion of the Multicomponent Transport Models for Normal Pressure Plasmas

We deal in the following with the so-called hierarchical model equations, see [28, 29], which approximate the behaviour of the normal pressure plasmas.

For the first start, we can simply deal with a two-fluid formulation, where we decouple heavy particles (ions, molecules, atoms) into light particles (electrons). Furthermore, a more appropriate model is done with the multifluid formulation, where we can apply for each heavy particle species (e.g. we distinguish between the different ions and atoms of O , N , \dots) and apply an individual distribution function.

Furthermore, we extend the transport equations with the Stefan–Maxwell equation, see the ideas in [25].

As a start point to derive the hierarchical equations with heavy and light particles in the plasma bulk, we use the Boltzmann equation:

$$\frac{\partial}{\partial t} f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f = \langle f \rangle, \quad (4.1)$$

- f : Density function of the a general particle species;
- \mathbf{v} : General velocity in the bulk;
- q : Particle charge in general;
- m : Mass of the species;
- $\langle f \rangle$: Collision term in general;
- \mathbf{E} : Electrical field vector; and
- \mathbf{B} : Magnetical field vector.

For the heavy particle in general and electrons, we derive the fluid model with the help of the velocity moments to obtain the macroscopic quantities, see [30].

Two-Component Fluid Model

In the following, we assume a simple description of the all heavy particles i (i.e. all ions and neutrons) and all electrons e .

We have the following Assumption 4.1:

- Assumption 4.1** • We concentrate on the density function of the heavy particles (we neglect the electrons, based on their relative small mass compared to the ions and neutrons).
- We assume that we do not have mixture of the different species and we only have to model the pure transport of one particle species.
 - An exact distribution function is not necessary for such regimes, while we do not consider a kinetic behaviour.
 - The extension between electrons and heavy particles (e.g. scattering) is sufficient by a approximated collision term, see also [15].

By applying the velocity momentums, we obtain the conservation equations of the heavy particles i and the electrons e , in the following equation with $\alpha = \{i, e\}$, see also [14]:

$$\frac{\partial \rho_\alpha}{\partial t} + \nabla_{\mathbf{x}} \cdot (\rho_\alpha \mathbf{u}) = m_\alpha Q_n^{(\alpha)}, \quad (4.2)$$

$$\begin{aligned} \frac{\partial}{\partial t} \rho_\alpha \mathbf{u}_\alpha + \nabla_{\mathbf{x}} \cdot (\rho_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha + nT \underline{I} - \underline{\tau}^*) \\ = q_\alpha n_\alpha (\mathbf{E} + \mathbf{u}_\alpha \times \mathbf{B}) - Q_m^e, \end{aligned} \quad (4.3)$$

$$\begin{aligned} \frac{\partial}{\partial t} E_{total}^* + \nabla_{\mathbf{x}} \cdot (E_{total}^* \mathbf{u} + \mathbf{q}^* + nT \mathbf{u} - \underline{\tau}^* \cdot \mathbf{u}) \\ = q_\alpha n_\alpha \mathbf{E} - Q_\varepsilon^{(e)}, \end{aligned} \quad (4.4)$$

- ρ_α : Mass density of the species α ;
- \mathbf{u}_α : Averaged velocity of the species α ;
- $Q_n^\alpha, Q_m^e, Q_\varepsilon^e$: Collision integral based on the mass, momentum and energy conservation;
- q_α : Heat flow of the species α ;
- n_α : Density of species α ;
- \mathbf{E} : Electrical field vector;
- \mathbf{B} : Magnetical field vector; and
- E_{total}^* : Total energy of all species.

Furthermore, we have to add the Maxwell equations for the electro-magnetic field, see [15].

Multicomponent Fluid Model with Fickian's Approach without Stefan–Maxwell Approach)

In the following, we apply a first multicomponent model based on the work of [9, 14], where all the heavy particles are described. The Fickian's approach is used and we assume to have dominant species, e.g. majorant species, which can be applied as a matrix background such that binary diffusion is sufficient, see [25].

We have therefore the following Assumption 4.2.

Assumption 4.2 The assumptions for the Fickian's approach are given as follows:

- Each heavy particle species is described with an individual density function.
- We apply only a simple summation of the transport parameters, which results in a phenomenological result (not the derivation with Stefan–Maxwell equations).
- The electrons are modelled in the same manner as in the two-component fluid model, see [15].

We have the following notation and constraints of the heavy particle.

The notation for the multicomponent formulation is given as follows:

- N : Number of species;
- n_s : Particle density of species $s, s = 1, \dots, N$;
- $n = \sum_{s=1}^N n_s$: Total particle density;
- T : Particle energy of all heavy particles, e.g. $T = k_B T_{gas}$;
- $\rho = \sum_{s=1}^N \rho_s$: Mass density of all particles with ρ_s as the mass particle density of species s ;

- $\rho_s = m_s n_s$, n_s , m_s : Mass of species s ;
- $\mathbf{c}_s = \mathbf{u}_s - \mathbf{u}$, \mathbf{c}_s : Difference or diffusion velocity of species s ;
- \mathbf{u}_s : Drift velocity of species s ; and
- \mathbf{u} : Drift velocity of the total system and given as $\mathbf{u} = \frac{1}{\rho} \sum_{s=1}^N \rho_s \mathbf{u}_s$.

The model equation with the binary diffusion coefficients, see in the paper of Senega/Brinkmann [14], is given for the heavy particles $s \in \{1, \dots, N\}$:

$$\frac{\partial}{\partial t} n_s + \nabla_{\mathbf{x}} \cdot (n_s \mathbf{u}_s + n_s \mathbf{c}_s) = Q_n^{(s)}, \quad (4.5)$$

$$\frac{\partial}{\partial t} \rho \mathbf{u} + \nabla_{\mathbf{x}} \cdot (\rho \mathbf{u} \mathbf{u} + nT \underline{\mathbf{I}} - \underline{\boldsymbol{\tau}}^*) = \sum_{s=1}^N q_s n_s \langle E \rangle, \quad (4.6)$$

$$\begin{aligned} \frac{\partial}{\partial t} E_{total}^* + \nabla_{\mathbf{x}} \cdot (E_{total}^* \mathbf{u} + \mathbf{q}^* + nT \mathbf{u} - \underline{\boldsymbol{\tau}}^* \cdot \mathbf{u}) \\ = \sum_{s=1}^N q_s n_s (\mathbf{u} + \mathbf{c}_s) \cdot \langle E \rangle - Q_{inel,ST}^{(e)}, \end{aligned} \quad (4.7)$$

where

$$E_{total}^* = \sum_{s=1}^N \frac{1}{2} \rho_s \mathbf{c}_s^2 + \frac{1}{2} \rho \mathbf{u}^2 + \frac{3}{2} nT + \sum_{s=1}^N \rho_s \Delta h_{f,s}^0, \quad (4.8)$$

see also in paper [14].

An improvement of the standard derivation of such models is obtained with the individual density functions for all different heavy particle species, such that we obtain the following representation for the values \mathbf{c}_s , \mathbf{q}^* and $\underline{\boldsymbol{\tau}}^*$ with

$$\mathbf{c}_s = -d_T^{(s)} \nabla_{\mathbf{x}} T - \sum_{\alpha=1}^N D_n^{(\alpha,s)} \frac{1}{n_s} \nabla_{\mathbf{x}} n_{\alpha}, \quad (4.9)$$

$$\mathbf{q}^* = \lambda_E \langle E \rangle - \lambda \nabla_{\mathbf{x}} T - \sum_{s=1}^N \sum_{\alpha=1}^N \lambda_n^{(\alpha,s)} \frac{1}{n_s} \nabla_{\mathbf{x}} n_{\alpha}, \quad (4.10)$$

$$\underline{\boldsymbol{\tau}}^* = -\eta \left(\nabla_{\mathbf{x}} \mathbf{u} + (\nabla_{\mathbf{x}} \mathbf{u})^T - \frac{2}{3} (\nabla_{\mathbf{x}} \cdot \mathbf{u}) \underline{\mathbf{I}} \right). \quad (4.11)$$

The production terms (e.g. collision terms, reaction terms) are approximated in the following operators:

$$Q_n^{(e)} = \int_{v_s} \langle f_s \rangle d^3 v_s = \sum_r a_{\text{sign},r} k_{\alpha,r} n_\alpha n_r, \quad (4.12)$$

$$Q_n^{(s)} = \int_{v_s} \langle f_s \rangle d^3 v_s = \sum_r a_{\text{sign},r} k_{\alpha,r} n_\alpha n_r, \quad (4.13)$$

where $k_{\alpha,r}$ is the parameter of the averaged collision rates, see [14] and $a_{\text{sign},r}$ is the signum function, $a_{\text{sign},r} = 1$ is a source term and $a_{\text{sign},r} = -1$ is a sink term.

Multicomponent Fluid Model with Stefan–Maxwell Equation

In the following, we discuss the extended multicomponent description, which is generalized via the Stefan–Maxwell approach.

The Stefan–Maxwell equation allows a systematical derivation of the diffusion processes, where the mixture of the different species is considered, such that we can also discuss counter diffusion, which is possible in ternary diffusion processes. Also, the thermodynamical behaviour is discussed accurately without heuristic assumptions as in the Fickian’s approach.

We discuss in the following the extension of the transport parameters with respect to the Stefan–Maxwell equation, see [16].

We assume the following:

- Each heavy particle species can be described with an individual density function.

Our notations are used as in the section “Multicomponent Fluid model with Stefan–Maxwell Equation”.

We apply the transport equation:

$$\frac{\partial}{\partial t} n_s + \nabla_{\mathbf{x}} \cdot (n_s \mathbf{u}_s + n_s \mathbf{c}_s) = Q_n^{(s)}, \quad (4.14)$$

with the diffusion velocity:

$$\mathbf{c}_s = -d_T^{(s)} \nabla_{\mathbf{x}} T - \sum_{\alpha=1}^N D_n^{(\alpha,s)} \frac{1}{n_s} \nabla_{\mathbf{x}} n_\alpha, \quad (4.15)$$

which is extended in the following with the Stefan–Maxwell equation.

We decompose into two fluxes:

$$\mathbf{c}_s = \mathbf{c}_{s,1} + \mathbf{c}_{s,2}, \quad (4.16)$$

where $c_{s,1}$ is the thermal flux and $c_{s,2}$ is the diffusive flux

$$\mathbf{c}_{s,1} = -d_T^{(s)} \nabla_{\mathbf{x}} T, \quad (4.17)$$

$$\mathbf{c}_{s,2} = j_s, \quad (4.18)$$

where j_s is the so-called driving force of the species s .

In our case, we restrict us to the chemical potential as driving force:

$$j_s = n_s \nabla_{\mathbf{x}} \mu_s, \quad (4.19)$$

where $\mu_s = \log(\gamma_s n_s)$ and γ_s is the so-called activation constant ($\gamma_s > 0$) and we obtain

$$j_s = \nabla_{\mathbf{x}} n_s, \quad (4.20)$$

where $\sum_{s=1}^N j_s = 0$, i.e. the sum of all fluxes is equal to 0 and the also the sum of the mass rates is zero

$$\sum_{s=1}^N y_s = 0, \quad (4.21)$$

where $y_s = \frac{\rho_s}{\rho}$.

The Stefan–Maxwell equation is given as

$$j_s = \left(\sum_{j=1}^N \frac{1}{\tilde{D}_{sj}} (y_s j_j - y_j j_s) \right). \quad (4.22)$$

We can compute the flux matrix $j = (j_1, \dots, j_N)^T \in \mathbb{R}^{N \times N}$, where j^s is the column vector of j with $M = \text{diag}(m_s)$, $e = [1, \dots, 1]^T$, $P(y) = I - y \otimes e = I - (\cdot, |e\rangle y$ (where \otimes is the dyadic product), and we obtain the equation

$$\begin{cases} B(y) j^\alpha = P(y) M^{-1} \partial_{x_\alpha} y, \quad \alpha = 1, \dots, n, \\ B(y) = [b_{ij}(y)], \quad b_{ij}(y) = f_{ij} y_i, \\ \text{fur } i \neq j, \quad b_{ii}(y) = -\sum_{l=1}^N f_{il} y_l, \quad i, j = 1, \dots, N, \end{cases} \quad (4.23)$$

furthermore, $\tilde{D}_{ij} = f_{ij}$, $i, j = 1, \dots, N$ is the multidiffusion coefficient and n is the number of spatial dimensions, e.g. $n = 2$ or $n = 3$, see [19].

4.1.1.4 Solver Ideas for the Multicomponent System with the Stefan–Maxwell Equation

We can apply different numerical schemes to solve the multicomponent system with Stefan–Maxwell equation. Some are discussed in the following:

- Implicit Ideas: Solve the coupled nonlinear transport equation with relaxation methods.
- Explicit Ideas: Direct solving of Stefan–Maxwell equations, where we apply the overdetermined equation system and solve analytically the parameters (such a

analytical method is very delicate and only applicable to binary or ternary diffusion operators, see [31]).

- Variational formulations: We apply an additional Poisson's equation to solve the constraint of the Stefan–Maxwell equation. We obtain a saddle point problem, which can be solved by standard mixed finite element methods.

Implicit Method

We apply an implicit method with iteration scheme and rewrite the full equation system into a quasilinear, strong coupled parabolic differential equation, where we consider for simplicity only the mass conservation:

$$\rho \frac{\partial}{\partial t} y + \text{Div}_x (A(y)P(y)M^{-1}[\nabla_x y]^T)) = Q_n, \text{ in } \Omega, t > 0, \quad (4.24)$$

$$\frac{\partial}{\partial n} y = 0, \text{ auf } \partial\Omega, t > 0, \quad (4.25)$$

$$y(0) = y_0, \text{ in } \Omega, \quad (4.26)$$

$y = (y_1, \dots, y_N)$ and $Q_n = \rho(m_1, Q_n^1, \dots, m_N, Q_n^N)$. Furthermore, we have $A(y) = (B(y)|_{E_{xt}})$, where the matrix B is extended to an invertable matrix. We have $\nabla_x y = [\partial_\alpha y_j] \in \mathbb{R}^{n \times N}$ and Div_x is the divergence in each row of the matrix.

We can show, under some conditions, that we have a existing solution, see [19].

Based on the existing solution, we can apply the following iterative scheme, for the time intervals $n = 0, \dots, N$ and iterative steps $i = 0, \dots, I$:

$$U'_{i+1} = A_1(U_i)U_{i+1} + A_2(U_i)U_i, t \in [t^n, t^{n+1}], \quad (4.27)$$

$$U_i(t^n) = U(t^n), \quad (4.28)$$

where $U(t^n)$ is the approximated solution of the last iterative cycle and $U_0(t)$ is an estimated initial or starting solution for the next cycle, e.g. $U_0(t) = U(t^n)$. The stopping criterion is given as $\|U_{i+1}(t^{n+1}) - U_i(t^{n+1})\| \leq \text{err}$ or the limit of the number of iterative steps $i = I$. Furthermore, the operator A_1 is the convection part and A_2 is the diffusion part of the transport equation.

The iterative method is convergent with the assumption of the existence of the solution and boundedness of the operators, see [20, 21].

Explicit Method

Here, we have the benefit of a fast and direct solver for small systems, e.g. binary or ternary systems.

A main drawback is the application to larger systems of quaternary or higher mixtures, while it is hard to find the explicit equations.

We show the method based on a three-component system, given as

$$\partial_t \xi_i + \nabla \cdot N_i = 0, \quad 1 \leq i \leq 3, \quad (4.29)$$

$$\sum_{j=1}^3 N_j = 0, \quad (4.30)$$

$$\frac{\xi_2 N_1 - \xi_1 N_2}{D_{12}} + \frac{\xi_3 N_1 - \xi_1 N_3}{D_{13}} = -\nabla \xi_1, \quad (4.31)$$

$$\frac{\xi_1 N_2 - \xi_2 N_1}{D_{12}} + \frac{\xi_3 N_2 - \xi_2 N_3}{D_{23}} = -\nabla \xi_2, \quad (4.32)$$

where we have $\Omega \in \mathbb{R}^d$, $d \in \mathbb{N}^+$ mit $\xi_i \in C^2$.

We can simplify to

$$\partial_t \xi_i + \nabla \cdot N_i = 0, \quad 1 \leq i \leq 2, \quad (4.33)$$

$$\frac{1}{D_{13}} N_1 + \alpha N_1 \xi_2 - \alpha N_2 \xi_1 = -\nabla \xi_1, \quad (4.34)$$

$$\frac{1}{D_{23}} N_2 - \beta N_1 \xi_2 + \beta N_2 \xi_1 = -\nabla \xi_2, \quad (4.35)$$

where $\alpha = \left(\frac{1}{D_{12}} - \frac{1}{D_{13}} \right)$, $\beta = \left(\frac{1}{D_{12}} - \frac{1}{D_{23}} \right)$.

We obtain the explicit solvable Stefan–Maxwell equation with the multidiffusion coefficients:

$$D_{12} = \tilde{D}_{12} \left[1 + \frac{w_3}{M_3} \left(\frac{M_3}{M_2} \tilde{D}_{13} - \tilde{D}_{12} \right) + \frac{w_3}{D_3} \tilde{D}_{12} \right], \quad (4.36)$$

where \tilde{D}_{ij} are the binary diffusion coefficients, M_i is the molar mass of the species i and w_i is the mass rate of the species i , see also the derivation in the paper [18, 32].

Variational Formulation

Here, we can apply standard software codes, which are done in the direction of the Poisson's equation.

A drawback of the method is that we have to solve a saddle point problem, which needs iterative solver methods, which are expensive and apply special solver schemes, e.g. Lagrangian multipliers.

Formulation with respect to the Poisson's equation is

$$-\Delta u = r, \quad \text{in } \Omega, \quad (4.37)$$

$$u = f, \quad \text{auf } \partial\Omega, \quad (4.38)$$

where $\partial\Omega$ is the boundary of the domain Ω .

A solution of the problem equation is given via a mixed formulation as a saddle point problem:

$$p - \nabla u = 0, \text{ in } \Omega, \quad (4.39)$$

$$\nabla \cdot p = r, \text{ in } \Omega, \quad (4.40)$$

which means that we find a solution for the mixed formulation of $(p, u) \in Q \times V$:

$$\int_{\Omega} (pq + u \nabla \cdot q) = \int_{\partial \Omega} f q \cdot n ds, \quad (4.41)$$

$$\int_{\Omega} v \nabla \cdot p dx = - \int_{\Omega} r v dx, \quad (4.42)$$

where n is the outer normal vector of $\partial \Omega$.

The variational formulation of the Stefan–Maxwell equation is given as

$$\int_{\Omega} (-\nabla \xi_i q_i) dx = \int_{\Omega} \frac{1}{c_{tot}} \left(\sum_{i=1}^N \alpha_{ij} (\xi_i J_j q_i - \xi_j J_i q_i) \right) dx, \quad (4.43)$$

$$\int_{\Omega} v_i \nabla \cdot J_i dx = \int_{\Omega} r_i v_i dx, \quad (4.44)$$

where r_i is the reaction rate (e.g. collision term), J_i is the flux, and ξ_i is the molar rate of species i . c_{tot} is the total concentration of the mixture.

Regularization Method: Regularization of the Transport Model with Stefan–Maxwell Equation

There exist several more methods; a well-known idea is the regularization method.

We start with the macroscopic model and extend the Stefan–Maxwell equation to a regular and solvable system.

The flux term is given as

$$c_s = \frac{1}{\rho_s} \nabla j_s, \quad (4.45)$$

where j_s are the mass flux densities with the following constraints:

- $\sum_{s=1}^N j_s = 0$ (i.e. all fluxes are zero), and
- $\sum_{s=1}^N w_s = 1$ (i.e. all mass rates are 1),

where $x_s = w_s \frac{\tilde{M}}{M_s}$ and x_s are the molar rates, M_s is the molar mass of species s , \tilde{M} is the molar mass of the mixture and further the density of the mixture is given as $\tilde{\rho} = (1 - \sum_{s=1}^{N-1} w_s) \rho_N + \sum_{s=1}^{N-1} w_s \rho_s$.

The Stefan–Maxwell approach is the equibalance of the molar rates for each individual diffusive flux:

$$-\nabla x_s = \frac{\tilde{M}}{\tilde{\rho}} \left(\sum_{j=1}^N \frac{1}{\tilde{D}_{sj}} \left(x_s \frac{j_j}{M_j} - x_j \frac{j_s}{M_s} \right) \right), \quad (4.46)$$

and we obtain the equation system

$$FV = -d, \quad (4.47)$$

where $d = (\nabla x_1, \dots, \nabla x_M)^t$, $V = (j_1, \dots, j_M)^t$ and F is a singular matrix of the equation system (4.46).

The next step is the regularization of the singular equation system and we obtain the novel diffusion matrix:

$$\tilde{F} = F + \alpha y \otimes y, \quad (4.48)$$

where $y = (n_1, \dots, n_M)^t$, α is a parameter for the solver method and \otimes is the dyadic product.

Based on this regularisation, we can apply a standard iterative method and solve the Stefan–Maxwell equation and also the heavy-particle equations together in a large linear equation system. Such a combination allows to apply fast linear equation solvers, e.g. SuperILU solvers.

4.1.1.5 Conclusion

The extension of the known standard heavy particle model with an improved diffusion part can be done with the Stefan–Maxwell equation.

The former summation approach is replaced by the balance approach, see [16, 18, 19], which is done with the Stefan–Maxwell equation.

The former modelling approaches are extended and the solver methods can be applied. But we have to extend also the analytical or numerical methods for the singular perturbed novel equation system.

Therefore, we have to modify the simulation packages with respect to the novel diffusion part.

While explicit methods to solve the Stefan–Maxwell equations are fast and simple to implement, they lack with larger systems and larger species in the mixture. Implicit methods are more flexible and also resolve higher mixtures but are more time-consuming in the computations, while we apply iterative schemes.

At the end, it is an approach how large the systems and the mixtures are, while for small systems, we apply an explicit method and for large systems we have to apply an implicit approach.

4.1.2 Multicomponent Fluid Transport Model for Groundwater Flow

We concentrate on such models, which deal with the transport behaviour of fluids in the porous media, see [26].

Such models arose of the background to understand flow of water in aquifers, transport of pollutants in aquifers or underlying rocks and propagation of stresses, see [26, 33].

We concentrate on introducing the mathematical models, see also [34–36] and discuss possible solver methods to simulate such models.

4.1.2.1 Introduction and Mathematical Model

We consider a steady-state groundwater flow that is described by a given velocity field $\mathbf{v} = \mathbf{v}(x)$ for $x \in \Omega \subset R^d$ for $d = 2$ or $d = 3$. In the groundwater, several radionuclides (or some other chemical species) are dissolved.

We suppose that these nuclides take part in irreversible, first-order chemical reactions. Particularly, each nuclide (a “mother”) can decay only to a single component (to a “daughter”), but each nuclide can be produced by several reactions, i.e. each daughter can have several mothers, see [34].

Moreover, the radionuclides can be adsorbed to the soil matrix. If equilibrium linear sorption is assumed with different sorption constants for each component, the advective–dispersive transport of each component is slowed down by a different retardation factor.

Summarizing, the mathematical model can be written in the form [33, 34]

$$R^{(i)} \phi \left(\partial_t c^{(i)} + \lambda^{(ij)} c^{(i)} \right) + \nabla \cdot \left(\mathbf{v} c^{(i)} - D^{(i)} \nabla c^{(i)} \right) = \sum_k R^{(k)} \phi \lambda^{(ki)} c^{(k)}, \quad (4.49)$$

where $i = 1, \dots, I_c$. The integer I_c denotes the total number of involved radionuclides. A stationary groundwater is supposed by considering only divergence-free velocity field, i.e.

$$\nabla \cdot \mathbf{v}(x) = 0, \quad x \in \Omega. \quad (4.50)$$

The unknown functions $c^{(i)} = c^{(i)}(t, x)$ denote the concentrations of radionuclides, where the space and time variables (t, x) are considered as $t \geq 0$ and $x \in \Omega$. The constant reaction rate $\lambda^{(ij)} \geq 0$ determines the decay (sink) term $\lambda^{(ij)} c^{(i)}$ for the concentration $c^{(i)}$ and the production (source) term for the concentration $c^{(j)}$. In general, the j th radionuclide need not to be included in the system (4.49), i.e. $j > I_c$. The indices k in the right-hand side of (4.49) run through all mothers of the i th radionuclide.

The remaining parameters in (4.49) include the diffusion–dispersion tensors $D^{(i)} = D^{(i)}(x, \mathbf{v})$ [33], the retardation factors $R^{(i)} = R^{(i)}(x) \geq 1$ and the porosity of medium $\phi = \phi(x) > 0$.

For the modelling of processes on the boundary $\partial\Omega$ of the domain Ω , we apply standard inflow and outflow boundary conditions. Particularly, we neglect the diffusive–dispersive flux at the outflow (and “noflow”) boundary $\partial^{out}\Omega := \{x \in \partial\Omega, \mathbf{n} \cdot \mathbf{v} \geq 0\}$,

$$\mathbf{n} \cdot D^{(i)} \nabla c^{(i)}(t, \gamma) = 0, \quad t > 0, \quad \gamma \in \partial\Omega, \quad (4.51)$$

where \mathbf{n} is the normal unit vector with respect to $\partial\Omega$. For the case of inflow boundary $\partial^{in}\Omega := \{x \in \partial\Omega, \mathbf{n} \cdot \mathbf{v} < 0\}$, we assume that the concentrations are prescribed by Dirichlet boundary conditions:

$$c^{(i)}(t, \gamma) = C^{(i)}(t, \gamma), \quad t > 0, \quad \gamma \in \partial^{in}\Omega. \quad (4.52)$$

The functions $C^{(i)}$ can describe decay reactions in a waste site (e.g. a nuclear waste repository), and, in such a way, they shall be related to each other, see, e.g. [37].

The initial conditions are considered in a general form:

$$c^{(i)}(0, x) = C^{(i)}(0, x), \quad x \in \Omega. \quad (4.53)$$

4.1.2.2 Solver Ideas for the Multicomponent Fluid Transport Model

If we assume simple domains, e.g. one-dimensional problems and special boundary and initial conditions, for the problem (4.49), we could derive analytical solutions, see for example [37].

Such analytical solutions solve the multicomponent behaviour analytically in an explicit equation.

For more general applications, e.g. multidimensions and general boundary and initial conditions, it is necessary to deal with a discretized equation.

Here, we have the following methods to discretize the spatial operators, for example:

- Finite element methods, see [38] and
- Finite-volume methods, see [39].

We concentrate on the finite-volume scheme, which allow to deal with the conservation equation and apply geometrically the derivation of the convection and diffusion term, see [40].

The finite-volume discretization method, see [42], allows to deal with a general velocity $\mathbf{v} = \mathbf{v}(x)$ and general boundary and initial conditions (4.51)–(4.53).

We have the following ideas:

- We apply analytical solutions for locally one-dimensional advection-reaction problems on boundaries between two finite volumes, see also Godunov algorithm [40]; and
- We split the diffusion part of (4.49) using operator splitting procedure and apply finite-volume method, see [41].

If we have nonlinearities, we apply a linearization method, e.g. fixpoint scheme or Newton's method. Based on the linearized equations in (4.49), linear splitting schemes can be applied and decoupled to several simpler problems. Applying afterwards the principle of superposition, one can obtain the solution of (4.49) by summing the solutions of such simpler problems.

4.1.2.3 Splitting Method for the Multicomponent Fluid Transport Model

We decompose the multicomponent fluid transport equation into a convection-reaction part and a diffusion part.

While the convection-reaction part is solved exactly with one-dimensional solutions and Godunov's scheme is applied, the diffusion part is solved in the spatial operators with finite-volume discretization scheme and in the time operator with implicit time discretization.

Convection-Reaction Part

We apply the following convection-reaction equation:

$$\partial_t \left(R^{(l)} \phi u^{(l)} \right) + \nabla \cdot \left(\mathbf{v} u^{(l)} \right) + \lambda^{(l)} R^{(l)} \phi u^{(l)} = \lambda^{(l-1)} R^{(l-1)} \phi u^{(l-1)}. \quad (4.54)$$

We apply the Godunov's method which means the solution of the one-dimensional convection-reaction equations, which are embedded as mass transfer to the finite-volume scheme, see [42].

So we solve for each underlying one-dimensional Ω_i and the mass concentration to the out-flowing cell $j \in \text{out}(i)$, a one-dimensional convection-reaction equations for each species $l = 1, \dots, I$:

$$R_i^{(l)} \phi_i \partial_t u_i^{(l)} + v_{ij} \partial_x u_i^{(l)} + \lambda^{(l)} R_i^{(l)} \phi_i u_i^{(l)} = \lambda^{(l-1)} R_i^{(l-1)} \phi_i u_i^{(l-1)}. \quad (4.55)$$

We transform to a directly solvable convection-reaction system, with the following as

$$c_i^{(l)} := R_i^{(l)} \phi_i u_i^{(l)}, \quad \tilde{v}_{ij} = \frac{v_{ij}}{R_i^{(l)} \phi_i}, \quad (4.56)$$

and we obtain

$$\partial_t c_i^{(l)} + \tilde{v}_{ij} \partial_x c_i^{(l)} + \lambda^{(l)} c_i^{(l)} = \lambda^{(l-1)} c_i^{(l-1)}. \quad (4.57)$$

For each cell, we compute the total outflow fluxes

$$\tau_{l,i} = \frac{V_i R^{(l)}}{v_i}, \quad v_i = v_{ij}, \quad j = out(i).$$

Based on the restriction of the local time, we have the minimum over all possible cell time steps:

$$\tau^n \leq \min_{\substack{l=1,\dots,m \\ i=1,\dots,I}} \tau_{l,i},$$

and we obtain a velocity of the finite-volume cell:

$$v_{l,i} = \frac{1}{\tau_{l,i}}.$$

Then, we can calculate the mass, which is important to embed into the FV discretization:

$$\begin{aligned} m_{ij,rest}^{(l),n} &= m_1^{(l),n}(a, b, \tau^n, v_{1,i}, \dots, v_{l,i}, R^{(1)}, \dots, R^{(l)}, \lambda^{(1)}, \dots, \lambda^{(l)}), \\ m_{ij,out}^{(l),n} &= m_2^{(l),n}(a, b, \tau^n, v_{1,i}, \dots, v_{l,i}, R^{(1)}, \dots, R^{(l)}, \lambda^{(1)}, \dots, \lambda^{(l)}), \end{aligned}$$

where $a = V_i R^{(l)}(c_{ij}^{(l),n} - c_{ij'}^{(l),n})$, $b = V_i R^{(l)} c_{ij'}^{(l),n}$ and $m_i^{(l),n} = V_i R^{(l)} c_i^{(l),n}$ are the parameters and $j = out(i)$, $j' = in(i)$.

The discretization with the embedded analytical mass is given by

$$m_i^{(l),n+1} = m_{ij,rest}^{(l),n} + m_{j'i,out}^{(l),n},$$

where $m_{ij,rest}^{(l),n} = m_i^{(l),n} - m_{ij,out}^{(l),n}$ is rest mass coming from the total mass and the outflow mass, see [42].

Diffusion Part

We discretize the diffusion part with the finite-volume methods. We can concentrate on the following equation:

$$\partial_t R c - \nabla \cdot (D \nabla c) = 0, \quad (4.58)$$

where $c = c(x, t)$ with $x \in \Omega$ and $t \geq 0$. The diffusion is given as $D \in \mathbb{R}^+$ and the retardation factor is $R > 0.0$.

The equation is integrated over time and space (implicit time and mass averaging in space):

$$\int_{\Omega_j} \int_{t^n}^{t^{n+1}} \partial_t R(c) dt dx = \int_{\Omega_j} \int_{t^n}^{t^{n+1}} \nabla \cdot (D \nabla c) dt dx. \quad (4.59)$$

After applying Green's formula and the approximation in the finite cells (i.e. Γ_j is the boundary of the finite-volume cell Ω_j), we have for one finite cell

$$V_j R(c_j^{n+1}) - V_j R(c_j^n) = \tau^n \sum_{e \in \Lambda_j} \sum_{k \in \Lambda_j^e} |\Gamma_{jk}^e| \mathbf{n}_{jk}^e \cdot D_{jk}^e \nabla c_{jk}^{e,n+1}, \quad (4.60)$$

where $|\Gamma_{jk}^e|$ is the length of the boundary element Γ_{jk}^e .

We calculate the gradients via piecewise finite element function ϕ_l and obtain

$$\nabla c_{jk}^{e,n+1} = \sum_{l \in \Lambda^e} c_l^{n+1} \nabla \phi_l(\mathbf{x}_{jk}^e). \quad (4.61)$$

Then, we obtain the finite-volume discretization for the diffusion part:

$$\begin{aligned} & V_j R(c_j^{n+1}) - V_j R(c_j^n) \\ &= \tau^n \sum_{e \in \Lambda_j} \sum_{l \in \Lambda^e \setminus \{j\}} \left(\sum_{k \in \Lambda_j^e} |\Gamma_{jk}^e| \mathbf{n}_{jk}^e \cdot D_{jk}^e \nabla \phi_l(\mathbf{x}_{jk}^e) \right) (c_j^{n+1} - c_l^{n+1}), \end{aligned} \quad (4.62)$$

where the finite cells are given as $j = 1, \dots, m$.

For such a discretization, we can embed the convection-reaction part via a splitting approach, which is given in the following.

Coupling Part

The different parts of the full equations are coupled via a operator splitting method.

We apply the following splitting approach:

$$c^*(t^{n+1}) = c(t^n) + \tau_n A c(t^n) \quad (4.63)$$

$$c^{**}(t^{n+1}) = c^*(t^{n+1}) + \tau_n B c^{**}(t^{n+1}), \quad (4.64)$$

where the time step is $\tau^n = t^{n+1} - t^n$ and $n = 1, \dots, N$ are the number of time steps. The operator A is the convection-reaction operator, which can be resolved in the equation analytically. The operator B is the diffusion operator, which is solved via FV methods and implicit Euler method.

Based on the analytical resolution of the convection-reaction part, we have the following splitting approach:

$$c^{**}(t^{n+1}) = (I - \tau_n B)^{-1} c^*(t^{n+1}), \quad (4.65)$$

where $c^*(t^{n+1})$ is the analytical solution of the convection-reaction part.

The splitting error is of the first order based on the non-commuting operators, see [42].

Remark 4.2 Based on the analytical embedding of the convection-reaction equation, we can speed up the solver scheme and concentrate on solving the diffusion part. Here, based on the first-order splitting scheme, we can see the method as following: The diffusion equation is only perturbed by a convection-reaction part, see [43].

4.1.3 Conclusion

For the multicomponent fluid transport model, it is important to decompose into simpler and faster solvable equation-parts. Each equation-part, e.g. convection-reaction part or diffusion part, can be solved with more adequate schemes, which are more effective and faster as a full equation solver. We have applied fast solver methods for the convection-reaction part, e.g. modified Godunov's method embedded to finite volume schemes, and for the diffusion part, e.g. finite volume schemes to discretize the spatial operators. The parts are coupled with fast operator splitting schemes, which allow to concentrate on the diffusion solver, while the convection-reaction part can be embedded as an explicit solved part. Such effective methods allow to solve the multicomponent fluid transport model with high accuracy and acceleration. In future, an extension of multicomponent fluid transport models with respect to additional equation-parts, e.g. multiphase parts or growth parts, are possible and the splitting schemes can be modified to such additional parts.

4.2 Multicomponent Kinetics

Abstract In this section, we discuss the models and applications based on the different multicomponent kinetic models. Here, we assume to have a microscopic scale, i.e. we deal with the fine resolution in the atomic scale. So we have a discrete description and discuss some models based on the multicomponent kinetics problems.

4.2.1 Multicomponent Langevin-Like Equations

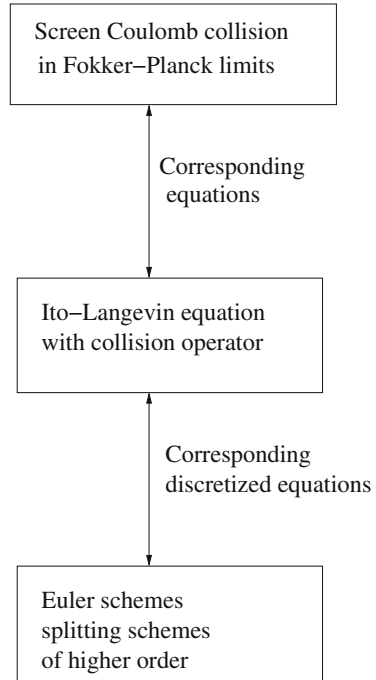
The idea is to apply an alternative model based on the Coulomb collision in plasma to reduce the computational in particle simulations.

The alternative models are based on Langevin equations, which are coupled nonlinear stochastic differential equations, see [44].

Historically, we have two ideas for algorithms for Coulomb collisions in particle simulations:

Fig. 4.2 Screen Coulomb collision in the Fokker–Planck limit

Collision Algorithm



- Binary algorithm: Particles in a finite cell, see particle in cell, are organized into discrete pairs (therefore binary algorithm) of interacting particles. The collision is based on Coulomb collision of two particles, see [45].
- Test particle algorithm: The collisions are modelled by defining a dual particles (test particles) and primary particles (field particles). The velocity of the test particle is modelled by Langevin equation, which is deposited on the space mesh [46].

The idea of the alternative approach is given in Fig. 4.2.

The main contribution to deal with the stochastic model is based on the following Remark 4.3 of the Coulomb collision approach:

Remark 4.3 Coulomb collisions can be approximated via defining test and field particles. The test particle velocity is subjected to drag and diffusion in three velocity dimensions using Langevin equations, see [47].

4.2.2 Introduction to the Model Equations

We are motivated to develop fast algorithms to solve Fokker–Planck equation with Coulomb collisions in plasma simulations.

The Fokker–Planck equations are given as

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E(x) \frac{\partial f}{\partial v} = \frac{\partial}{\partial v} \left(-\gamma v f + \beta^{-1} \gamma \frac{\partial f}{\partial v} \right), \quad (4.66)$$

where we could decouple such a FP equation into the PIC (particle in cell) part and the SDE part.

- PIC part

$$\frac{\partial f}{\partial t} + v \frac{\partial f}{\partial x} - E(x) \frac{\partial f}{\partial v} = 0, \quad (4.67)$$

- SDE part

$$\frac{\partial f}{\partial t} = \frac{\partial}{\partial v} \left(-\gamma v f + \beta^{-1} \gamma \frac{\partial f}{\partial v} \right), \quad (4.68)$$

where we solve the characteristics.

- PIC part

$$\frac{dx}{dt} = v, \quad (4.69)$$

$$\frac{dv}{dt} = -E(x) = \frac{\partial U}{\partial x}, \quad (4.70)$$

where U is the potential.

- SDE part

$$\frac{dx}{dt} = 0, \quad (4.71)$$

$$dv = -\gamma v dt + \sqrt{2\beta^{-1}\gamma} dW. \quad (4.72)$$

We apply the following nonlinear SDE problem:

$$\frac{dx}{dt} = v, \quad (4.73)$$

$$dv(t) = \frac{\partial}{\partial x} U(x) - \gamma v dt + \sqrt{2\beta^{-1}\gamma} dW, \quad (4.74)$$

where W is a Wiener process, γ is the thermostat parameter and β is the inverse temperature.

A long solution to the SDE is distributed according to a probability measure with density π satisfying

$$\pi(x, v) = C^{-1} \exp\left(-\beta \left(\frac{v^2}{2} + U(x)\right)\right), \quad (4.75)$$

where $x > 0.0$, $v \in \mathbb{R}$.

4.2.3 Analytical Methods for Mixed Deterministic–Stochastic Ordinary Differential Equations

In the following, we present an algorithm, which is based on solving the mixture of deterministic and stochastic ordinary differential equations.

The idea is based on the deterministic variation of constants to embed perturbed right-hand sides.

We deal with the following equations:

$$\frac{dX}{dt} = V, \quad (4.76)$$

$$dV = -E(x)dt - AVdt + BdW, \quad (4.77)$$

$$\text{with } X(0) = X_0, \quad V(0) = V_0,$$

where W is a Wiener process with the $N(0, \sqrt{\Delta})$ distributed.

We rewrite to a linear operator and a nonlinear and stochastic function.

$$\frac{d\mathbf{X}}{dt} = \tilde{A}\mathbf{X} + \mathbf{E}(\mathbf{X}) + \frac{d\mathbf{W}}{dt}, \quad (4.78)$$

$$\text{with } \mathbf{X}_0 = (X_0, V_0)^t,$$

where $\mathbf{X} = (X, V)^t$ is the solution vector, $\mathbf{X}_0 = (X_0, V_0)^t$ is the initial vector, the matrix is $\tilde{A} = \begin{pmatrix} 0 & 1 \\ 0 & -A \end{pmatrix}$, the nonlinear function is $\mathbf{E} = \begin{pmatrix} 0 \\ -E(X) \end{pmatrix}$ and the stochastic function is $\frac{d\mathbf{W}}{dt} = \begin{pmatrix} 0 \\ B \frac{dW}{dt} \end{pmatrix}$.

The analytical solution is given with the exact integration of the $\exp(\tilde{A}s)$ (variation of constants):

$$\begin{aligned} \mathbf{X}(t^{n+1}) &= \exp(\tilde{A}\Delta t)\mathbf{X}_0 + \int_{t^n}^{t^{n+1}} \exp(\tilde{A}(t^{n+1} - s)) \mathbf{E}(\mathbf{X}(s)) ds \\ &\quad + \int_{t^n}^{t^{n+1}} \exp(\tilde{A}(t^{n+1} - s)) d\mathbf{W}_s, \end{aligned} \quad (4.79)$$

$$\mathbf{X}(t^{n+1}) = \exp(\tilde{A}\Delta t)\mathbf{X}_0 + \tilde{\mathbf{E}}(\mathbf{X}_0) + \tilde{\mathbf{W}}(\mathbf{X}_0),$$

where the electric field integral is computed with a higher order exponential Runge–Kutta method.

Integration of the E-field function with fourth-order Runge–Kutta method is as follows:

$$\mathbf{k}_1 = \Delta t \mathbf{E}(\mathbf{X}^n), \quad (4.80)$$

$$\mathbf{k}_2 = \Delta t (\mathbf{E}(\exp(\tilde{A} \Delta t / 2) \mathbf{X}^n + \frac{1}{2} \exp(\tilde{A} \Delta t / 2) \mathbf{k}_1)), \quad (4.81)$$

$$\mathbf{k}_3 = \Delta t (\mathbf{E}(\exp(\tilde{A} \Delta t / 2) \mathbf{X}^n + \frac{1}{2} \mathbf{k}_2)), \quad (4.82)$$

$$\mathbf{k}_4 = \Delta t (\mathbf{E}(\exp(\tilde{A} \Delta t) \mathbf{X}^n + \exp(\tilde{A} \Delta t / 2) \mathbf{k}_2)), \quad (4.83)$$

$$\tilde{\mathbf{E}}(\mathbf{X}^n) = \frac{1}{6} \left(\exp(\tilde{A} \Delta t) \mathbf{k}_1 + 2 \exp(\tilde{A} \Delta t / 2) (\mathbf{k}_2 + \mathbf{k}_3) + \mathbf{k}_4 \right), \quad (4.84)$$

and the stochastic integral is computed as

$$\begin{aligned} \tilde{\mathbf{W}}(\mathbf{X}^n) &= \int_{t^n}^{t^{n+1}} \exp(\tilde{A}(t^{n+1} - s)) d\mathbf{W}_s \\ &= \sum_{j=0}^{N-1} \exp\left(\tilde{A} \left(\frac{t^{n,j} + t^{n,j+1}}{2}\right)\right) (\mathbf{W}(t^{n,j+1}) - \mathbf{W}(t^{n,j})), \end{aligned} \quad (4.85)$$

$$\Delta t = (t^{n+1} - t^n) / N, \quad t^{n,j} = \Delta t + t^{n,j-1}, \quad t^{n,0} = t^n. \quad (4.86)$$

Remark 4.4 Based on the perturbation and finer time scales, the stochastic integral is resolved with finer time steps as the non-stochastic parts. Therefore, we have applied an adaptive numerical integration method that allows to apply additional smaller time intervals with more integration points. We obtained more accurate numerical results of the stochastic integral and reduce the numerical error of the full scheme.

4.2.4 Conclusion

For the multicomponent kinetics, we have additional stochastic equation-parts. Therefore, it is important to resolve such stochastic parts with high accurate stochastic solvers. We have highly perturbed and finer time scale to resolve such multiscale parts. In our case, we proposed analytical methods, which solved the stochastic part with semi-analytical methods and embedded directly the results to the deterministic (non-stochastic) part. Here, we obtain high accurate results, while we could concentrate on the deterministic solver parts. In the future, an extension of multicomponent kinetics to many particle applications, e.g. plasma dynamics, is important and we can apply the idea of the analytical embedding of the stochastic part to the deterministic parts to reduce the computational time.

4.3 Additive Operator Splitting with Finite-Difference Time-Domain Method: Multiscale Algorithms

Abstract We discuss numerical methods based on additive operator splitting schemes, which are used to solve Maxwell equations, see [48]. The discretization schemes are given with Finite-Difference Time-Domain (FDTD) methods, which apply finite differences in time and space and allow to conserve the physical behaviour of the equations, see [49]. Because of the 3D Maxwell equations, we result into large semi-discretized equation systems, i.e. we have to deal with large systems of ordinary differential equations. Therefore, we are motivated to optimize 3D computations of electro-magnetic fields with decomposition methods, which decompose into different time and spatial scales. Here, we discuss additive operator splitting schemes, which allow to decompose into several independent solvable smaller equation systems, see [2]. We embed the FDTD schemes into the additive splitting and result into a multiscale approach into each spatial dimension.

4.3.1 Introduction

We are motivated to split large semi-discretized equation systems, e.g. resulted from FDTD schemes, see [3], with additive operator splitting schemes, which allow to concentrate on each individual dimension and each time and spatial scale of the underlying reduced equation systems, see [2].

In Fig.4.3, we present the multiscale splitting with the FDTD discretization scheme and the AOS (additive operator splitting scheme) as a multiscale splitting approach.

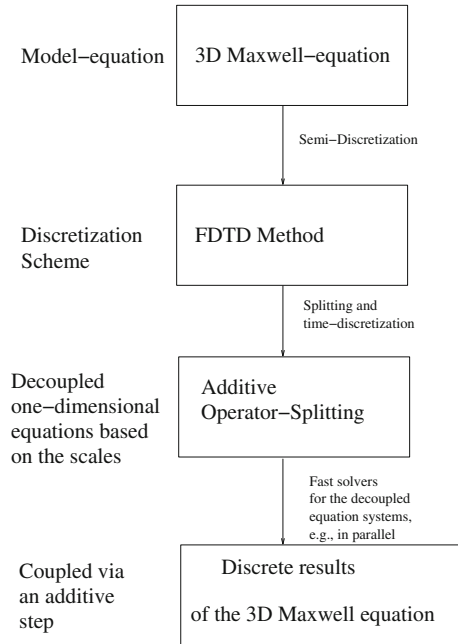
While explicit time-discretization schemes have restriction with respect to their CFL (Courant–Friedrichs–Lewy) condition, we also discuss implicit time-discretization schemes based on modified FDTD and AOS schemes, which overcome such restrictions, see [50, 51].

4.3.2 Introduction FDTD Schemes

One of the simplest FDTD schemes is the Yee’s algorithm, see [49]. The ideas are given in the following:

- We combine time and space discretization on a time–space grid. Using central difference schemes for both time and space, we obtain second-order methods with respect to the CFL condition of the discretization schemes.
- A staggered grid is necessary to obtain for both time and space second-order schemes and obtain a stable discretization scheme, see [49].

Fig. 4.3 Splitting approach based on an FDTD discretized Maxwell equation



In the following, we present the so-called Yee’s cells in 2D and 3D, see Figs. 4.5 and 4.4.

Such cells are applied with respect to the time and spatial discretization and their staggered behaviours allow to achieve a second-order scheme.

In the following example, we discuss a first-order FDTD method, see Example 4.1.

Example 4.1 We have the following preparations to achieve the higher order scheme:

- We discretize both time and space with a central difference, which is a second-order scheme.
- We decompose into a primary and dual grid, i.e. we apply a staggered grid for the magnetic and electric field equations.
- We step forward in time.

We start with the following 1D equations:

$$\frac{\partial E_x}{\partial t} = -\frac{1}{\epsilon_0} \frac{\partial H_y}{\partial z}, \tag{4.87}$$

$$\frac{\partial H_y}{\partial t} = -\frac{1}{\mu_0} \frac{\partial E_x}{\partial z}, \tag{4.88}$$

where we have an initial condition of the impulse and absorbing boundary conditions. We deal with a wave-front solution in the z direction.

We apply the 1D FDTD method as follows:

Fig. 4.4 Staggered grid: 2D

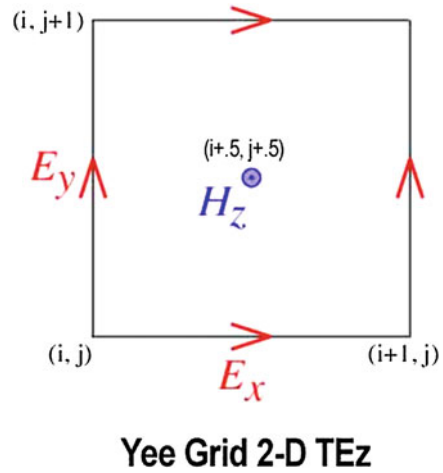
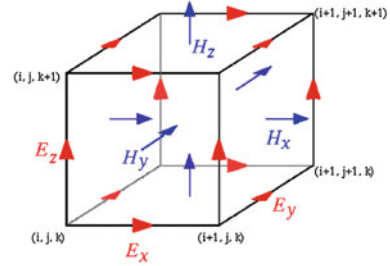


Fig. 4.5 Staggered Grid für FDTD Methoden für 3D



- The simplest 1D FDTD Schema is the Yee’s method.
- We stagger E_x and H_y in time and space with a half-time and half-spatial step.
- We apply the central difference scheme for the time and space coordinates.

We obtain the 1D equation as

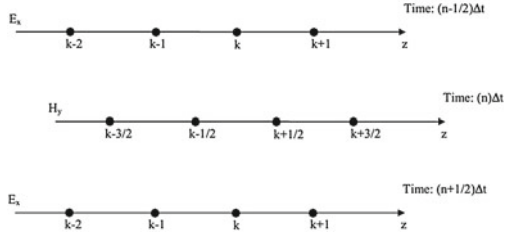
$$\frac{E_x^{n+1/2}(k) - E_x^{n-1/2}(k)}{\Delta t} = -\frac{1}{\epsilon_0} \frac{H_y^n(k + 1/2) - H_y^n(k - 1/2)}{\Delta z}, \quad (4.89)$$

$$\frac{H_y^{n+1}(k + 1/2) - H_y^n(k - 1/2)}{\Delta t} = -\frac{1}{\mu_0} \frac{E_x^{n+1/2}(k + 1) - E_x^{n+1/2}(k)}{\Delta z}, \quad (4.90)$$

where we have a so-called *leap-frog* algorithm, i.e. first we apply $E_x^{n+1/2}$ for all spatial points and then we apply H_y^{n+1} for all spatial points. We step forward in time. For the discretization points of a 1D Yee’s algorithm, see Fig. 4.6.

Based on the explicit method, i.e. we step forward in time, we have restrictions for the stability in the time step. The CFL condition for the simple 1D Maxwell equation is given as

Fig. 4.6 Staggered grid: 1D



$$\Delta t \leq \frac{\Delta z}{c_0} \tag{4.91}$$

where c_0 is the light speed.

Remark 4.5 For the explicit higher dimensional FDTD methods, we have also the same restriction as for the 1D methods. We have also to restrict our time step in 2D and 3D, as follows, see also [49, 52]:

$$\Delta t \leq \frac{\min_{i=1}^3 \{\Delta x_i\}}{c_0 \sqrt{d}} \tag{4.92}$$

where $\Delta x_i, i = 1, \dots, d$ are the spatial steps, and c_0 is the light speed and $d = 2, 3$.

4.3.3 Additive Operator Splitting Schemes

The additive operator splitting scheme can be applied with respect to the different spatial dimensions. Based on their different scales, we can also apply the AOS scheme as a multiscale approach, see [2].

In the following, we discuss additive operator splitting schemes, see also [3].

We describe traditional operator splitting methods and focus our attention to the case of two linear operators, i.e. we consider the Cauchy problem,

$$\partial_t c(t) = \sum_{i=1}^m A_m(c) \quad t \in (0, T); \quad c(0) = c_0 \tag{4.93}$$

whereby the initial function c_0 is given, and A_1, \dots, A_m are assumed to be bounded nonlinear operators. (In many applications, they denote the spatially discretized operators, e.g. they correspond to the discretized in space convection and diffusion operators (matrices). Hence, they can be considered as bounded operators.)

We discuss the following schemes:

- AOS (explicit):

$$c^{n+1} = \left(I + t \sum_{i=1}^m A_i(c^n) \right) c^n, \quad (4.94)$$

while the method is closely related to the idea of the multiplicative splitting (A–B Splitting) in the explicit form:

$$\exp((A_1(c^n) + \dots + A_m(c^n))t) = \exp(A_1(c^n)t) \cdots \exp(A_m(c^n)t), \quad (4.95)$$

if one apply the explicit Euler to Eq. (4.93) and scheme, you neglect the second-order term $\mathcal{O}(t^2)$.

The scheme can be additively applied as

$$c_i^{n+1} = B_i(c^n)c^n, \quad i = 1, \dots, m, \quad (4.96)$$

$$c^{n+1} = c^n + \sum_{i=1}^m c_i^{n+1}, \quad (4.97)$$

with the operators $B_i(c^n) := t A_i(c^n)$.

- AOS (semi-implicit):

$$c^{n+1} = \left(I - t \sum_{i=1}^m A_i(c^n) \right)^{-1} c^n, \quad (4.98)$$

and further

$$c^{n+1} = \frac{1}{m} \left(\sum_{i=1}^m (I - m t A_i(c^n)) \right)^{-1} c^n, \quad (4.99)$$

with the operators $B_i(c^n) := \frac{1}{m}(I - m t A_i(c^n))$

The scheme can be additively applied as

$$c_i^{n+1} = B_i(c^n)^{-1} c^n, \quad i = 1, \dots, m, \quad (4.100)$$

$$c^{n+1} = \sum_{i=1}^m c_i^{n+1}. \quad (4.101)$$

4.3.4 Application to the Maxwell Equations

We have the following Maxwell equation:

$$\frac{\partial \mathbf{E}}{\partial t} = \frac{1}{\varepsilon} \nabla \times \mathbf{H} - \frac{1}{\varepsilon} \sigma \mathbf{E}, \quad (4.102)$$

$$\frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \mathbf{E}, \quad (4.103)$$

where the operators are $c = \mathbf{E}$, $v = \mathbf{H}$ and we have the abstract formulation:

$$\frac{\partial \mathbf{c}}{\partial t} = \mathcal{A} \mathbf{c} + \mathcal{A}_4 \mathbf{c}, \quad (4.104)$$

with $\mathbf{c} = (c, v)^t$, $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3 = \begin{pmatrix} 0 & \frac{1}{\varepsilon} A \\ -\frac{1}{\mu} A^* & 0 \end{pmatrix}$, $\mathcal{A}_4 = \begin{pmatrix} -\frac{\sigma}{\varepsilon} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{pmatrix}$,

where we have $\mathcal{A}, \mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4 \in \mathbb{R}^{6 \times 6}$ and $A, A_1, A_2, A_3, I_{3 \times 3}, 0_{3 \times 3} \in \mathbb{R}^{3 \times 3}$ with $I_{3 \times 3}$ as the identity matrix and $0_{3 \times 3}$ as the zero matrix.

The decomposition is given in the following steps. Each full $A = A_1 + A_2 + A_3$ is divided into a single dimension as

$$A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial x} & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 0 & \frac{\partial}{\partial y} \\ 0 & 0 & 0 \\ -\frac{\partial}{\partial y} & 0 & 0 \end{pmatrix}, A_3 = \begin{pmatrix} 0 & -\frac{\partial}{\partial z} & 0 \\ \frac{\partial}{\partial z} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

Here, we have to apply an AOS scheme with four operators.

The full version is given as

$$\frac{\partial \mathbf{c}}{\partial t} = \begin{pmatrix} -\frac{\sigma}{\varepsilon} & 0 & 0 & 0 & -\frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & -\frac{\sigma}{\varepsilon} & 0 & \frac{1}{\varepsilon} \frac{\partial}{\partial z} & 0 & -\frac{1}{\varepsilon} \frac{\partial}{\partial x} \\ 0 & 0 & -\frac{\sigma}{\varepsilon} & -\frac{1}{\varepsilon} \frac{\partial}{\partial y} & \frac{1}{\varepsilon} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{1}{\mu} \frac{\partial}{\partial z} & -\frac{1}{\mu} \frac{\partial}{\partial y} & 0 & 0 & 0 \\ -\frac{1}{\mu} \frac{\partial}{\partial z} & 0 & \frac{1}{\mu} \frac{\partial}{\partial x} & 0 & 0 & 0 \\ \frac{1}{\mu} \frac{\partial}{\partial y} & -\frac{1}{\mu} \frac{\partial}{\partial x} & 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{c}, \quad (4.105)$$

based on the equations, and when we apply AOS, we split into the following six matrices:

$$\mathcal{A}_{11} = \begin{pmatrix} -\frac{\sigma}{\varepsilon} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\varepsilon} \frac{\partial}{\partial z} & 0 \\ 0 & 0 & 0 & -\frac{1}{\varepsilon} \frac{\partial}{\partial y} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.106)$$

$$\mathcal{A}_{21} = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{\partial}{\partial y} \\ 0 & -\frac{\sigma}{\varepsilon} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\varepsilon} \frac{\partial}{\partial x} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.107)$$

$$\mathcal{A}_{31} = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{\partial}{\partial y} \\ 0 & 0 & 0 & 0 & -\frac{1}{\varepsilon} \frac{\partial}{\partial x} \\ 0 & -\frac{\sigma}{\varepsilon} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.108)$$

$$\mathcal{A}_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{\mu} \frac{\partial}{\partial z} & 0 & 0 & 0 & 0 \\ \frac{1}{\mu} \frac{\partial}{\partial y} & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.109)$$

$$\mathcal{A}_{22} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\mu} \frac{\partial}{\partial z} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{\mu} \frac{\partial}{\partial x} & 0 & 0 & 0 \end{pmatrix}, \quad (4.110)$$

$$\mathcal{A}_{32} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\mu} \frac{\partial}{\partial y} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\mu} \frac{\partial}{\partial x} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{4.111}$$

The splitting is based as follows:

$$\mathbf{c}^{n+1} = \frac{1}{6} \left(\sum_{i=1}^3 \sum_{j=1}^2 (I - 6 \Delta t \mathcal{A}_{i,j})^{-1} \right) \mathbf{c}^n, \tag{4.112}$$

for example, the first operator is given as

$$\mathbf{c}_1^{n+1} = \frac{1}{6} (I - 6 \Delta t \mathcal{A}_{11})^{-1} \mathbf{c}^n. \tag{4.113}$$

If we apply the finite difference discretization of a structured grid, we obtain the following matrices:

- We assume to have $N \times N \times N$ grid points, i.e. $H_x, H_y, H_z, E_x, E_y, E_z \in \Omega \in \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N = \mathbb{R}^{N^3}$.
- The matrices are given as $\mathcal{A}_{i,j} \in 6\mathbb{R}^{N^3} \times 6\mathbb{R}^{N^3}$, where $i = 1, 2, 3$ and $j = 1, 2$.
- For the discretization, we apply the following submatrices: $I \in \mathbb{R}^N \times \mathbb{R}^N$ is the identity matrix, $0 \in \mathbb{R}^N \times \mathbb{R}^N$ is the zero matrix and $M \in \mathbb{R}^N \times \mathbb{R}^N$ which is needed for the difference matrices and given as

$$M = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ -1 & 0 & 0 & \dots & 0 \\ 0 & -1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 0 \end{pmatrix},$$

- The difference matrices for $M_x, M_y, M_z \in \mathbb{R}^{N^3} \times \mathbb{R}^{N^3}$ are given as

$$M_x = \frac{1}{\Delta x} \begin{pmatrix} I + M & 0 & \dots & \dots & 0 \\ 0 & I + M & 0 & \dots & 0 \\ 0 & 0 & I + M & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & 0 & 0 & I + M \end{pmatrix},$$

$$M_y = \frac{1}{\Delta y} \begin{pmatrix} I & 0 & \dots & \dots & 0 \\ M & I & 0 & \dots & 0 \\ 0 & M & I & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & M & I \end{pmatrix},$$

$$M_z = \frac{1}{\Delta z} \begin{pmatrix} \tilde{I} & 0 & \dots & \dots & 0 \\ \tilde{M} & \tilde{I} & 0 & \dots & 0 \\ 0 & \tilde{M} & \tilde{I} & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \tilde{M} & \tilde{I} \end{pmatrix},$$

where $\tilde{I} \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2}$ is the identity matrix and $\tilde{M} \in \mathbb{R}^{N^2} \times \mathbb{R}^{N^2}$ is given as

$$\tilde{M} = \begin{pmatrix} M & 0 & \dots & \dots & 0 \\ 0 & M & 0 & \dots & 0 \\ 0 & 0 & M & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & M \end{pmatrix}.$$

Then for example the first operator is discretized as

$$\mathbf{C}_1^{n+1} = \frac{1}{6} (I_{Disc} - 6 \Delta t \mathcal{A}_{11, Disc})^{-1} \mathbf{C}^n, \quad (4.114)$$

where $I_{\mathcal{A}} \in \mathbb{R}^{N^3} \times \mathbb{R}^{N^3}$ is the identity matrix, $0_{\mathcal{A}} \in \mathbb{R}^{N^3} \times \mathbb{R}^{N^3}$ is the zero matrix and we have

$$I_{Disc} = \begin{pmatrix} I_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & I_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & I_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & I_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & I_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & I_{\mathcal{A}} \end{pmatrix}, \quad (4.115)$$

$$\mathcal{A}_{11, Disc} = \begin{pmatrix} \frac{\sigma}{\varepsilon} I_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & \frac{1}{\varepsilon} M_z & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & -\frac{1}{\varepsilon} M_y & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \\ 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} & 0_{\mathcal{A}} \end{pmatrix}, \quad (4.116)$$

furthermore, we have $\mathbf{C}^{n+1} = (\mathbf{E}_{x, disc}, \mathbf{E}_{y, disc}, \mathbf{E}_{z, disc}, \mathbf{H}_{x, disc}, \mathbf{H}_{y, disc}, \mathbf{H}_{z, disc})^T$ and all $\mathbf{E}_{x, disc}, \mathbf{E}_{y, disc}, \mathbf{E}_{z, disc}, \mathbf{H}_{x, disc}, \mathbf{H}_{y, disc}, \mathbf{H}_{z, disc} \in \mathbb{R}^{N^3}$.

Remark 4.6 Here, we have an application of a semi-implicit AOS scheme, while the nonlinearity in Eq. (4.93), i.e. $A_i(c^{n+1})$, is approximated via $A_i(c^n)$, which means that we restrict us to the linearization of the previous time point t^n and therefore, we embed also a CFL condition.

4.3.5 Practical Formulation of the 3D-FDTD Method

For more practical reasons, we consider on a simpler scheme based on the staggered time step method, such that we apply semi-implicit schemes.

Maxwell's equations in lossy and frequency independent materials are given as

$$\nabla \times \mathbf{E} = -\mu \frac{\partial \mathbf{H}}{\partial t}, \quad (4.117)$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t}, \quad (4.118)$$

$$\frac{\partial \mathbf{D}}{\partial t} = \sigma \mathbf{E} + \varepsilon_0 \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \quad (4.119)$$

where σ is the conductivity, μ is the permeability, ε_0 is the vacuum permittivity, ε_r is the relative permittivity, \mathbf{E} is the electric field, \mathbf{D} is the electric flux density and \mathbf{H} is the magnetic field. Equation (4.118) is Maxwell–Ampere equation without free currents.

We apply the operator $\nabla \times$ to the equations

$$\begin{aligned} \nabla \times \mathbf{E} &= \left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) \mathbf{i}_x + \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) \mathbf{i}_y + \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \mathbf{i}_z \\ &= -\mu \frac{\partial (H_x \mathbf{i}_x + H_y \mathbf{i}_y + H_z \mathbf{i}_z)}{\partial t}, \end{aligned} \quad (4.120)$$

$$\begin{aligned} \nabla \times \mathbf{H} &= \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \mathbf{i}_x + \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \mathbf{i}_y + \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \mathbf{i}_z \\ &= \frac{\partial (D_x \mathbf{i}_x + D_y \mathbf{i}_y + D_z \mathbf{i}_z)}{\partial t}, \end{aligned} \quad (4.121)$$

$$\begin{aligned} \frac{\partial \mathbf{D}}{\partial t} &= \frac{\partial (D_x \mathbf{i}_x + D_y \mathbf{i}_y + D_z \mathbf{i}_z)}{\partial t} \\ &= \sigma (E_x \mathbf{i}_x + E_y \mathbf{i}_y + E_z \mathbf{i}_z) + \varepsilon_0 \varepsilon_r \frac{\partial (E_x \mathbf{i}_x + E_y \mathbf{i}_y + E_z \mathbf{i}_z)}{\partial t}. \end{aligned} \quad (4.122)$$

where \mathbf{i}_x , \mathbf{i}_y and \mathbf{i}_z are the unit vectors in x , y and z directions. Then Eqs. (4.120),

(4.121) and (4.122) are expressed in a scalar manner as

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t}, \quad (4.123)$$

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t}, \quad (4.124)$$

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t}, \quad (4.125)$$

$$\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} = \frac{\partial D_x}{\partial t}, \quad (4.126)$$

$$\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} = \frac{\partial D_y}{\partial t}, \quad (4.127)$$

$$\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} = \frac{\partial D_z}{\partial t}, \quad (4.128)$$

$$\frac{\partial D_x}{\partial t} = \sigma E_x + \varepsilon_0 \varepsilon_r \frac{\partial E_x}{\partial t}, \quad (4.129)$$

$$\frac{\partial D_y}{\partial t} = \sigma E_y + \varepsilon_0 \varepsilon_r \frac{\partial E_y}{\partial t}, \quad (4.130)$$

$$\frac{\partial D_z}{\partial t} = \sigma E_z + \varepsilon_0 \varepsilon_r \frac{\partial E_z}{\partial t}. \quad (4.131)$$

In the following, we apply the semi-implicit version of an additive splitting approach to our equation.

4.3.6 Explicit Discretization

Here, the time and space derivatives are discretized by centred differences and the fields affected by the curl operators and staggered in time.

First, we discretize the conductivity term:

$$\frac{D_x^{n+1/2}(i, j, k) - D_x^{n-1/2}(i, j, k)}{\Delta t} = \sigma E_x^{n+1/2}(i, j, k) + \varepsilon_0 \varepsilon_r \frac{E_x^{n+1/2}(i, j, k) - E_x^{n-1/2}(i, j, k)}{\Delta t}, \quad (4.132)$$

$$\frac{D_y^{n+1/2}(i, j, k) - D_y^{n-1/2}(i, j, k)}{\Delta t} = \sigma E_y^{n+1/2}(i, j, k) + \varepsilon_0 \varepsilon_r \frac{E_y^{n+1/2}(i, j, k) - E_y^{n-1/2}(i, j, k)}{\Delta t}, \quad (4.133)$$

$$\frac{D_z^{n+3/2}(i, j, k) - D_z^{n+1/2}(i, j, k)}{\Delta t} = \sigma E_z^{n+1/2}(i, j, k) + \varepsilon_0 \varepsilon_r \frac{E_z^{n+1/2}(i, j, k) - E_z^{n-1/2}(i, j, k)}{\Delta t}. \quad (4.134)$$

Then we discretize the magnetic part:

$$\left\{ \frac{H_z^n(i, j, k) - H_z^n(i, j - 1, k)}{\Delta y} - \frac{H_y^n(i, j, k) - H_y^n(i, j, k - 1)}{\Delta z} \right\} \\ = \frac{D_x^{n+1/2}(i, j, k) - D_x^{n-1/2}(i, j, k)}{\Delta t}, \quad (4.135)$$

$$\left\{ \frac{H_x^n(i, j, k) - H_x^n(i, j, k - 1)}{\Delta z} - \frac{H_z^n(i, j, k) - H_z^n(i - 1, j, k)}{\Delta x} \right\} \\ = \frac{D_y^{n+1/2}(i, j, k) - D_y^{n-1/2}(i, j, k)}{\Delta t}, \quad (4.136)$$

$$\left\{ \frac{H_y^n(i, j, k) - H_y^n(i - 1, j, k)}{\Delta x} - \frac{H_x^n(i, j, k) - H_x^n(i, j - 1, k)}{\Delta y} \right\} \\ = \frac{D_z^{n+1/2}(i, j, k) - D_z^{n-1/2}(i, j, k)}{\Delta t}. \quad (4.137)$$

The last step is to discretize the electric part of the equation:

$$\left\{ \frac{E_z^{n+1/2}(i, j + 1, k) - E_z^{n+1/2}(i, j, k)}{\Delta y} - \frac{E_y^{n+1/2}(i, j, k + 1) - E_y^{n+1/2}(i, j, k)}{\Delta z} \right\} \\ = -\mu \frac{H_x^{n+1}(i, j, k) - H_x^n(i, j, k)}{\Delta t}, \quad (4.138)$$

$$\left\{ \frac{E_x^{n+1/2}(i, j, k + 1) - E_x^{n+1/2}(i, j, k)}{\Delta z} - \frac{E_z^{n+1/2}(i + 1, j, k) - E_z^{n+1/2}(i, j, k)}{\Delta x} \right\} \\ = -\mu \frac{H_y^{n+1}(i, j, k) - H_y^n(i, j, k)}{\Delta t}, \quad (4.139)$$

$$\left\{ \frac{E_y^{n+1/2}(i + 1, j, k) - E_y^{n+1/2}(i, j, k)}{\Delta x} - \frac{E_x^{n+1/2}(i, j + 1, k) - E_x^{n+1/2}(i, j, k)}{\Delta y} \right\} \\ = -\mu \frac{H_z^{n+1}(i, j, k) - H_z^n(i, j, k)}{\Delta t}, \quad (4.140)$$

Remark 4.7 We follow forward stepping $H^n \rightarrow E^{n+1/2} \rightarrow H^{n+1}$. Based on the staggered grid, we can follow such a forward staggering in time and space.

Furthermore, the spatial parts of the equations can be splitted by applying the explicit AOS scheme.

4.3.7 Combination: Discretization and Splitting

In the following, we discuss the combination of discretization and splitting. For example, (4.138) is split into the y direction part and the z direction part.

Therefore, we can apply the additive operator splitting scheme, where we decompose the electric field into a z - and y -part.

We have

$$-\mu \frac{\partial H_x}{\partial t} = \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z}, \quad H_x^n = H_x(t^n), \quad \Delta t = t^{n+1} - t^n, \quad (4.141)$$

and split into the two steps

$$-\mu \frac{\partial H_x^1}{\partial t} = \frac{\partial E_z}{\partial y}, \quad H_x^{n,1} = H_x(t^n), \quad \Delta t = t^{n+1} - t^n, \quad (4.142)$$

$$-\mu \frac{\partial H_x}{\partial t} = -\frac{\partial E_y}{\partial z}, \quad H_x^n = H_x^{n+1,1} = H_x^1(t^{n+1}), \quad \Delta t = t^{n+1} - t^n, \quad (4.143)$$

where the initial condition of the second equation is coupled by the solution of the first equation, see also A–B splitting, see [53].

The discretized version of the two steps is given as

$$\begin{aligned} & \frac{E_z^{n+1/2}(i, J+1, K) - E_z^{n+1/2}(i, j, k)}{\Delta y} \\ &= -\mu \frac{H_x^{n+1,1}(i, j, k) - H_x^{n,1}(i, j, k)}{\Delta t} \end{aligned} \quad (4.144)$$

where $H_x^{n,1}(i, j, k) = H_x^n(i, j, k)$, and the z direction part is

$$\begin{aligned} & -\frac{E_y^{n+1/2}(i, j, k+1) - E_y^{n+1/2}(i, j, k)}{\Delta z} \\ &= -\mu \frac{H_x^{n+1}(i, j, k) - H_x^{n+1,1}(i, j, k)}{\Delta t}. \end{aligned} \quad (4.145)$$

Remark 4.8 Here, we have an explicit AOS splitting scheme combined with a FDTD method. The discretization scheme is based on the staggered grid idea, while the splitting method is an explicit version.

4.3.8 Practical Formulation of the 3D-AOS-FDTD Method

For more practical reasons, we formulate Eq. (4.99) as

$$B_i(c^n)c_i^{n+1} = c^n, \quad i = 1, \dots, m, \quad (4.146)$$

$$c^{n+1} = \sum_{i=1}^m c_i^{n+1}. \quad (4.147)$$

The Maxwell's equation is given as in Eqs. (4.117)–(4.119). Furthermore, the operator $\nabla \times$ is applied to the equations and we obtain Eqs. (4.120)–(4.122). The equations can be presented in the scalar notation, which is given as in Eqs. (4.123)–(4.131).

In the following, we apply the additive splitting approach to our magnetic field equation, which are derived in the AOS scheme as follows:

- For the scalar field H_x , we have

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t}, \quad (4.148)$$

and the AOS scheme is given as

$$\frac{\partial H_x^*}{\partial t} = -\frac{1}{\mu} \frac{\partial E_z}{\partial y}, \quad H_x^*(t^n) = H_x(t^n), \quad (4.149)$$

$$\frac{\partial H_x^{**}}{\partial t} = \frac{1}{\mu} \frac{\partial E_y}{\partial z}, \quad H_x^{**}(t^n) = H_x^*(t^{n+1}), \quad (4.150)$$

where $H_x(t^{n+1}) = H_x^{**}(t^{n+1})$.

- For the scalar field H_y , we have

$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t}, \quad (4.151)$$

and the AOS scheme is given as

$$\frac{\partial H_y^*}{\partial t} = -\frac{1}{\mu} \frac{\partial E_x}{\partial y}, \quad H_y^*(t^n) = H_y(t^n), \quad (4.152)$$

$$\frac{\partial H_y^{**}}{\partial t} = \frac{1}{\mu} \frac{\partial E_z}{\partial z}, \quad H_y^{**}(t^n) = H_y^*(t^{n+1}), \quad (4.153)$$

where $H_y(t^{n+1}) = H_y^{**}(t^{n+1})$.

- For the scalar field H_z , we have

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t}, \quad (4.154)$$

and the AOS scheme is given as

$$\frac{\partial H_z^*}{\partial t} = -\frac{1}{\mu} \frac{\partial E_y}{\partial y}, \quad H_z^*(t^n) = H_z(t^n), \quad (4.155)$$

$$\frac{\partial H_z^{**}}{\partial t} = \frac{1}{\mu} \frac{\partial E_x}{\partial z}, \quad H_z^{**}(t^n) = H_z^*(t^{n+1}), \quad (4.156)$$

where $H_z(t^{n+1}) = H_z^{**}(t^{n+1})$.

4.3.9 Discretization of the Equations with the AOS

Here, the time and space derivatives are discretized by centred differences and the fields affected by the curl operators are averaged in time. We apply θ -schemes, i.e. the combination of an explicit and implicit time discretization, and can apply such a scheme to the AOS.

For example, we apply AOS Eqs.(4.149)–(4.150) and we have

$$\begin{aligned} & \theta \frac{E_z^{n+1}(i, j+1, k) - E_z^{n+1}(i, j, k)}{\Delta y} + (1-\theta) \frac{E_z^n(i, j+1, k) - E_z^n(i, j, k)}{\Delta y} \\ &= -\mu \frac{H_x^{*,n+1}(i, j, k) - H_x^n(i, j, k)}{\Delta t}, \end{aligned} \quad (4.157)$$

$$\begin{aligned} & -\theta \frac{E_y^{n+1}(i, j, k+1) - E_y^{n+1}(i, j, k)}{\Delta z} - (1-\theta) \frac{E_y^n(i, j, k+1) - E_y^n(i, j, k)}{\Delta z} \\ &= -\mu \frac{H_x^{**,n+1}(i, j, k) - H_x^{*,n+1}(i, j, k)}{\Delta t}, \end{aligned} \quad (4.158)$$

where the results are given as $H_x^{n+1}(i, j, k) = H_x^{**,n+1}(i, j, k)$.

For all Eqs.(4.123)–(4.131) applied to the AOS and the θ -scheme, we have the discretized equations as

$$\begin{aligned} & \theta \left\{ \frac{E_z^{n+1}(i, j+1, k) - E_z^{n+1}(i, j, k)}{\Delta y} - \frac{E_y^{n+1}(i, j, k+1) - E_y^{n+1}(i, j, k)}{\Delta z} \right\} \\ & + (1-\theta) \left\{ \frac{E_z^n(i, j+1, k) - E_z^n(i, j, k)}{\Delta y} - \frac{E_y^n(i, j, k+1) - E_y^n(i, j, k)}{\Delta z} \right\} \\ & = -\mu \frac{H_x^{n+1}(i, j, k) - H_x^n(i, j, k)}{\Delta t}, \end{aligned} \quad (4.159)$$

$$\begin{aligned} & \theta \left\{ \frac{E_x^{n+1}(i, j, k+1) - E_x^{n+1}(i, j, k)}{\Delta z} - \frac{E_z^{n+1}(i+1, j, k) - E_z^{n+1}(i, j, k)}{\Delta x} \right\} \\ & + (1-\theta) \left\{ \frac{E_x^n(i, j, k+1) - E_x^n(i, j, k)}{\Delta z} - \frac{E_z^n(i+1, j, k) - E_z^n(i, j, k)}{\Delta x} \right\} \\ & = -\mu \frac{H_y^{n+1}(i, j, k) - H_y^n(i, j, k)}{\Delta t}, \end{aligned} \quad (4.160)$$

$$\begin{aligned} & \theta \left\{ \frac{E_y^{n+1}(i+1, j, k) - E_y^{n+1}(i, j, k)}{\Delta x} - \frac{E_x^{n+1}(i, j+1, k) - E_x^{n+1}(i, j, k)}{\Delta y} \right\} \\ & + (1-\theta) \left\{ \frac{E_y^n(i+1, j, k) - E_y^n(i, j, k)}{\Delta x} - \frac{E_x^n(i, j+1, k) - E_x^n(i, j, k)}{\Delta y} \right\} \\ & = -\mu \frac{H_z^{n+1}(i, j, k) - H_z^n(i, j, k)}{\Delta t}, \end{aligned} \quad (4.161)$$

$$\begin{aligned} & \theta \left\{ \frac{H_z^{n+1}(i, j, k) - H_z^{n+1}(i, j-1, k)}{\Delta y} - \frac{H_y^{n+1}(i, j, k) - H_y^{n+1}(i, j, k-1)}{\Delta z} \right\} \\ & + (1-\theta) \left\{ \frac{H_z^n(i, j, k) - H_z^n(i, j-1, k)}{\Delta y} - \frac{H_y^n(i, j, k) - H_y^n(i, j, k-1)}{\Delta z} \right\} \\ & = \frac{D_x^{n+1}(i, j, k) - D_x^n(i, j, k)}{\Delta t}, \end{aligned} \quad (4.162)$$

$$\begin{aligned}
& \theta \left\{ \frac{H_x^{n+1}(i, j, k) - H_x^{n+1}(i, j, k-1)}{\Delta z} - \frac{H_z^{n+1}(i, j, k) - H_z^{n+1}(i-1, j, k)}{\Delta x} \right\} \\
& + (1-\theta) \left\{ \frac{H_x^n(i, j, k) - H_x^n(i, j, k-1)}{\Delta z} - \frac{H_z^n(i, j, k) - H_z^n(i-1, j, k)}{\Delta x} \right\} \\
& = \frac{D_y^{n+1}(i, j, k) - D_y^n(i, j, k)}{\Delta t}, \tag{4.163}
\end{aligned}$$

$$\begin{aligned}
& \theta \left\{ \frac{H_y^{n+1}(i, j, k) - H_y^{n+1}(i-1, j, k)}{\Delta x} - \frac{H_x^{n+1}(i, j, k) - H_x^{n+1}(i, j-1, k)}{\Delta y} \right\} \\
& + (1-\theta) \left\{ \frac{H_y^n(i, j, k) - H_y^n(i-1, j, k)}{\Delta x} - \frac{H_x^n(i, j, k) - H_x^n(i, j-1, k)}{\Delta y} \right\} \\
& = \frac{D_z^{n+1}(i, j, k) - D_z^n(i, j, k)}{\Delta t}, \tag{4.164}
\end{aligned}$$

where $\theta = [0, 1]$.

Remark 4.9 If we apply the conductivity as an operator, we have taken into account the averaging of the electric field \mathbf{E} term. Such an idea is done by θ method and afterwards, we can apply the additive operator splitting.

Further, we discretize Eqs. (4.129), (4.130) and (4.131), while the conductivity term and \mathbf{E} term are averaged in time.

We apply then

$$\begin{aligned}
& \frac{D_x^{n+1}(i, j, k) - D_x^n(i, j, k)}{\Delta t} \\
& = \sigma(\theta E_x^{n+1}(i, j, k) + (1-\theta)E_x^n(i, j, k)) + \varepsilon_0 \varepsilon_r \frac{E_x^{n+1}(i, j, k) - E_x^n(i, j, k)}{\Delta t}, \tag{4.165}
\end{aligned}$$

$$\begin{aligned}
& \frac{D_y^{n+1}(i, j, k) - D_y^n(i, j, k)}{\Delta t} \\
& = \sigma(\theta E_y^{n+1}(i, j, k) + (1-\theta)E_y^n(i, j, k)) + \varepsilon_0 \varepsilon_r \frac{E_y^{n+1}(i, j, k) - E_y^n(i, j, k)}{\Delta t}, \tag{4.166}
\end{aligned}$$

$$\begin{aligned}
& \frac{D_z^{n+1}(i, j, k) - D_z^n(i, j, k)}{\Delta t} \\
& = \sigma(\theta E_z^{n+1}(i, j, k) + (1-\theta)E_z^n(i, j, k)) + \varepsilon_0 \varepsilon_r \frac{E_z^{n+1}(i, j, k) - E_z^n(i, j, k)}{\Delta t}, \tag{4.167}
\end{aligned}$$

where $\theta \in [0, 1]$, i.e. $\theta = 1$ is implicit, $\theta = 0$ is explicit and $\theta = 1/2$ is semi-implicit

Then, (4.159)–(4.170) were split into the three direction parts.

For the pure implicit version, which conform with the AOS method, we have $\theta = 1$ and obtain

$$\frac{D_x^{n+1}(i, j, k) - D_x^n(i, j, k)}{\Delta t} = \sigma E_x^{n+1}(i, j, k) + \varepsilon_0 \varepsilon_r \frac{E_x^{n+1}(i, j, k) - E_x^n(i, j, k)}{\Delta t}, \quad (4.168)$$

$$\frac{D_y^{n+1}(i, j, k) - D_y^n(i, j, k)}{\Delta t} = \sigma E_y^{n+1}(i, j, k) + \varepsilon_0 \varepsilon_r \frac{E_y^{n+1}(i, j, k) - E_y^n(i, j, k)}{\Delta t}, \quad (4.169)$$

$$\frac{D_z^{n+1}(i, j, k) - D_z^n(i, j, k)}{\Delta t} = \sigma E_z^{n+1}(i, j, k) + \varepsilon_0 \varepsilon_r \frac{E_z^{n+1}(i, j, k) - E_z^n(i, j, k)}{\Delta t}. \quad (4.170)$$

Also this part can be splitted by applying the AOS scheme for the two operators.

Remark 4.10 At least, the AOS scheme is flexible and we could extend to the implicit version, i.e. $\theta = 1$. Here, we have to deal additional with inversion of the underlying equation system, which is more delicate, but we can skip the CFL conditions as time step conditions. Further additional steps are necessary and are computed implicitly.

4.3.10 Transport Equation Coupled with an Electro-magnetic Field Equations

The following example is discussed in [3], and concluded some of the important multiscale results.

We deal with the two-dimensional advection–diffusion equation and electric field equation:

$$\partial_t u = -v_x(E_z(x, y)) \frac{\partial u}{\partial x} - v_y \frac{\partial u}{\partial y} + D \frac{\partial^2 u}{\partial x^2} + D \frac{\partial^2 u}{\partial y^2}, \quad (4.171)$$

$$(x, y, t) \in \Omega \times (0, T),$$

$$u(x, y, t_0) = u_0(x, y), \quad (4.172)$$

$$\frac{\partial H_x(x, y)}{\partial t} = -\frac{\partial E_z}{\partial y}, \quad (x, y, t) \in \Omega \times (0, T), \quad (4.173)$$

$$\frac{\partial H_y(x, y)}{\partial t} = \frac{\partial E_z}{\partial x}, \quad (x, y, t) \in \Omega \times (0, T), \quad (4.174)$$

$$\frac{\partial E_z(x, y)}{\partial t} = \frac{1}{\varepsilon} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) - J_{source}, \quad (x, y, t) \in \Omega \times (0, T), \quad (4.175)$$

where we have the initial function:

$$u(\mathbf{x}, t_0) = u_a(\mathbf{x}, t_0) = \frac{1}{t_0} \exp \left(-\frac{(\mathbf{x} - \mathbf{v}t_0)^2}{4Dt_0} \right),$$

where $\mathbf{x} = (x, y)^t$ and $\mathbf{v} = (v_x, v_y)^t$, and we have

$$\begin{cases} v_x(E_z(x, y)) = 1, v_y = 1.0, & \text{for } t \in (0, t_0), \\ v_x(E_z(x, y)) = \alpha E_z(x, y), v_y = 0.0, & \text{for } t \geq t_0, \end{cases} \quad (4.176)$$

with $\alpha = 0.001, t_0 = 10.0$. The spatial domain is given as $\Omega = [0, 1] \times [0, 1]$.

The electric field $E_z(x, y)$ has the following line source:

$$J_{source}(x, y) = \sin(t) \text{ where } x = 0, y \in (0, 100).$$

The control of the particle transport is given by the electric field shown in Fig. 4.7.

In the following, we have the line sources with the results given in Fig. 4.8:

Fig. 4.7 Electric field in the apparatus

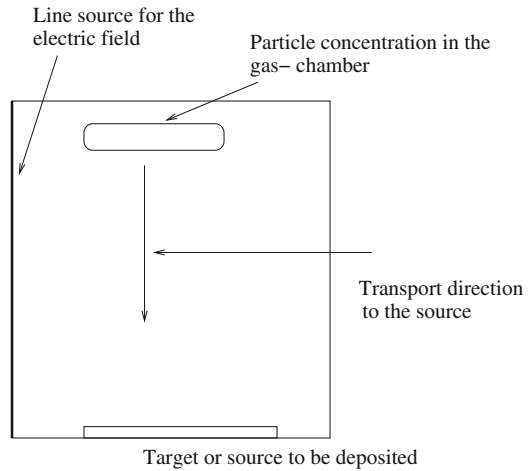
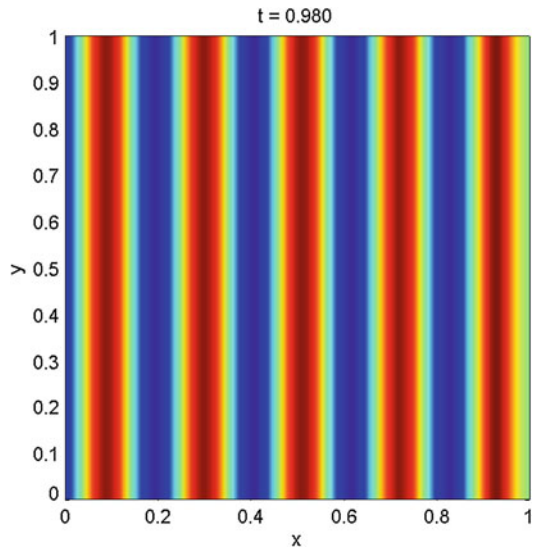


Fig. 4.8 Line source of the electric field in the apparatus



Numerically, we solve the equation, as in the following explicit AOS Algorithm 4.3:

Algorithm 4.3 We have coupled the equations by the following algorithm:

- (1) Initialize convection–diffusion equation, till t_0 .
- (2) Solve the electric field equation with t_{start} and obtain $E_z(x, y)$ for t_0
- (3) Solve convection–diffusion equation with $t_0 + \Delta t$ and use $E_z(x, y)$ for t_{start} for the unknown.
- (4) Do $t_0 = t_0 + \Delta t$ and go to (2) till $t_0 = t_{end}$

The following Figs. 4.9 and 4.10 show the developing concentration under the influence of the electric field, where $\alpha = 0.07$, $t_{start} = 0.5$ and $v_y = 0$ for $t \geq t_{start}$.

Remark 4.11 For spatial and time discretization, it is important to balance such schemes. If we apply an explicit AOS method and assume to have finite difference schemes in time and space, we have taken into account the CFL (Courant-Friedrichs-Levy) condition.

The condition for the explicit scheme is given as

$$\sqrt{\varepsilon} \Delta x \geq \Delta t, \tag{4.177}$$

where Δx and Δt are the spatial and time steps.

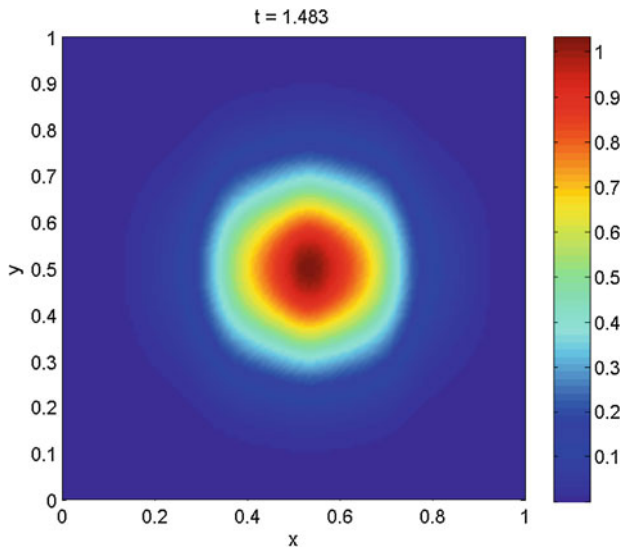
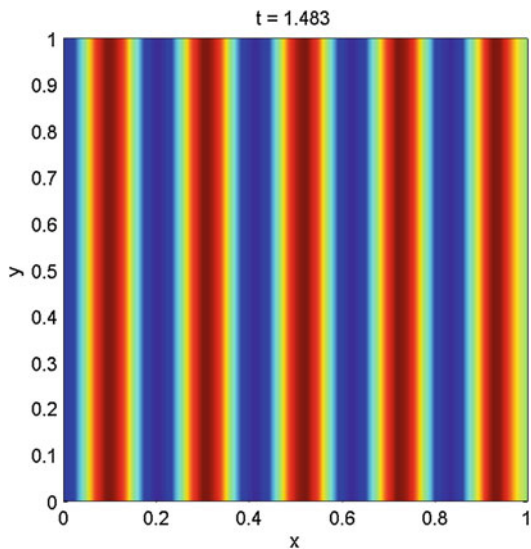


Fig. 4.9 Concentration density of the plasma specie, influenced by the electromagnetic field, in the apparatus at time $t = 1.483$ (the concentration flows from the left lower corner to the center)

Fig. 4.10 Electric field in the apparatus at time $t = 1.483$



Remark 4.12 Another idea is based on the following implicit AOS Algorithm 4.4, while Eqs. (4.171)–(4.175) are discretized as

$$\begin{aligned}
 u^{n+1}(i, j) = & u^n(i, j) \\
 & + \Delta t \left(-v_x(E_z^{n+1}(i, j)) \frac{u^n(i+1, j) - u^n(i, j)}{\Delta x} - v_y \frac{u^n(i, j+1) - u^n(i, j)}{\Delta y} \right. \\
 & \left. + D \frac{u^n(i+1, j) - 2u^n(i, j) + u^n(i-1, j)}{\Delta x^2} + D \frac{u^n(i, j+1) - 2u^n(i, j) + u^n(i, j-1)}{\Delta y^2} \right), \quad (4.178)
 \end{aligned}$$

$$\frac{H_x^{n+1}(i, j) - H_x^n(i, j)}{\Delta t} = - \frac{E_z^{n+1}(i, j+1) - E_z^{n+1}(i, j)}{\Delta y}, \quad (4.179)$$

$$\frac{H_y^{n+1}(i, j) - H_y^n(i, j)}{\Delta t} = \frac{E_z^{n+1}(i+1, j) - E_z^{n+1}(i, j)}{\Delta x}, \quad (4.180)$$

$$\frac{E_z^{n+1,*}(i, j) - E_z^n(i, j)}{\Delta t} = \frac{1}{\varepsilon} \left(\frac{H_y^{n+1}(i+1, j) - H_y^{n+1}(i, j)}{\Delta x} \right) - 0.5 J_{source}(i, j), \quad (4.181)$$

$$\frac{E_z^{n+1}(i, j) - E_z^{n+1,*}(i, j)}{\Delta t} = \frac{1}{\varepsilon} \left(- \frac{H_x^{n+1}(i, j+1) - H_x^{n+1}(i, j)}{\Delta y} \right) - 0.5 J_{source}(i, j), \quad (4.182)$$

where $i, j = 1, \dots, I$ are the spatial discretization points with Δx , and Δy are the spatial steps. Furthermore, Δt is the time step with $n = 0, 1, \dots, N$, which are the time points.

Then the equation system is given as

$$\mathcal{U}^{n+1} = (I - \Delta t A)^{-1} \mathcal{U}^n, \quad (4.183)$$

$$U^{n+1} = U^n + \Delta t B(v_x(\mathcal{E}_z^n), v_y, D)U^n, \quad (4.184)$$

where $U^{n+1} = (u^{n+1}(1, 1), \dots, u^{n+1}(I, I))^t$ is the discretized solution of the transport system, $\mathcal{U}^{n+1} = (\mathcal{H}_x^{n+1}, \mathcal{H}_y^{n+1}, \mathcal{E}_z^{n+1})^t$ is the discretized solution of the electro-magnetic field, with $\mathcal{H}_x^{n+1} = (H_x^{n+1}(1, 1), \dots, H_x^{n+1}(I, I))^t$, $\mathcal{H}_y^{n+1} = (H_y^{n+1}(1, 1), \dots, H_y^{n+1}(I, I))^t$ and $\mathcal{E}_z^{n+1} = (E_z^{n+1}(1, 1), \dots, E_z^{n+1}(I, I))^t$. Furthermore, the matrices $A \in \mathbb{R}^{I \times I}$ and $B \in \mathbb{R}^{3I \times 3I}$ are given and have embedded the boundary conditions.

The algorithmic idea 4.4 is given as follows.

Algorithm 4.4 We have coupled the equations by the following algorithm:

- (1) Initialize convection–diffusion equation, till t_0 and $n = 0$.
- (2) Solve implicitly the electro-magnetic field equation with the time step Δt and obtain \mathcal{E}_z^{n+1} for t_{n+1} .
- (3) Solve explicitly the convection–diffusion equation with Δt and use \mathcal{E}_z^{n+1} for t_{n+1} for the unknown and obtain U^{n+1} .
- (4) Do $t_{n+1} = t_n + \Delta t$ and go to (2) till $t_{n+1} = t_{end}$.

Here, we have the benefit that we are not restricted to the time step of the electro-magnetic field and we could apply the large time step, which is also applied for the convection–diffusion equation.

4.4 Extensions of Particle in Cell Methods for Nonuniform Grids: Multiscale Ideas and Algorithms

Abstract In this section, we discuss ideas to extend uniform particle in cell (PIC) method to nonuniform PIC methods. The ideas are based to modify the so-called PIC cycle parts, which decouple grid-free (particle methods), grid-based (field methods) and couple the parts with interpolation methods. The methodological idea of the PIC method can be seen as a multiscale method, while we deal with different underlying modelling scales, e.g. micro- and macroscopic scale. The different parts of the PIC method can be applied in different scales, e.g. a microscale (particle solver) and a macroscale (field solver). So we have a multiscale behaviour, while the transfer between the micro- and macro-model is done via interpolation or restriction, which is applied in the PIC method as spline approximations, see [54]. Another aspect results of the physical constraints mean that we have to fulfil the mass, momentum and energy conservation of the problem, see [54]. Such a problem can only be fulfilled for a uniform grid steps, while we deal with a primary grid. A modification to an adaptive or nonuniform grid needs to extend the freedom degrees of the underlying

grid, and therefore we have to deal with an additional so-called dual grid. On dual grid or in the logical space, we can extend the uniform grid into an adaptive grid and such a modification allows us to conserve the constraints, e.g. mass, momentum and energy conservation, see also [55, 56]. Both interpolation schemes (particle to grid and grid to particle) and solver methods (macrosolver: Poisson solver and microsolver: time integrator) have to be combined such that the physical constraints are fulfilled and the numerical errors are at least second order, see [57]. Here, we discuss the ideas to develop step-by-step multiscale extension of the PIC cycle. We modify shape function to adaptive shape functions and fit them to the adaptive discretization schemes such that the interpolations are of the same order as in the uniform case. Furthermore, we present some extensions to 2D and deal with simple 1D examples.

4.4.1 Introduction of the Problem

The motivation of the modification arose of a practical application in a propulsion problem. While in the inner or ion thruster part we deal with high density of particle and the outer or plume region, it has only a very low density of particles, see [58].

If we apply uniform PIC methods, we have taken into one spatial step for the full region, i.e. the very small spatial step of the inner region, and we have the problem of very long computational times.

In Fig. 4.11, we present the different spatial scales of the motivation.

Remark 4.13 The multiscale problem is given by the restriction of the time and spatial steps for a fine resolution of the inner part (restriction by the Debye length λ_D , where $\Delta x \leq \lambda_D$ and Δx is the spatial step size of the uniform grid. The Debye length is the distance scale over which significant charge densities can spontaneously exist,

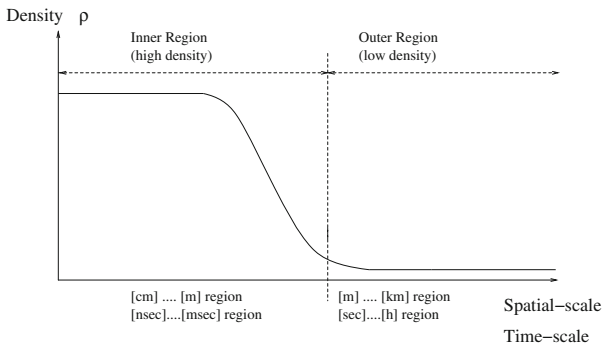


Fig. 4.11 The model problem with *inner* and *outer* region of different spatial and time scales

see [30]. It is therefore the largest scale, which can be resolved by the PIC method, see [54]. Moreover, if we deal with the multiscale problem of the test problem, we have to obtain very small spatial step sizes.

4.4.2 Introduction of the Extended Particle in Cell Method

The Particle in Cell (PIC) method is the well-known method over the last decades. The concept of coupling grid and grid-free methods are applied to accelerate the solver process. While parts of the equations are solved on a grid, e.g. Poisson equation, the transport of particles is done grid-free by computing the trajectories with fast time integrators, see [54, 59].

In recent applications, the flexibility of PIC schemes, with respect to higher order schemes and nonuniform grids, is important (Fig. 4.12).

In the following, we discussed a possible flexibilization of the PIC cycles based on improving all parts of the cycle, see Fig. 4.2.

The following three parts of the PIC can be improved:

- Shape function (higher order spline functions, which fulfil the constraints, e.g. TSC or higher, see [54]).

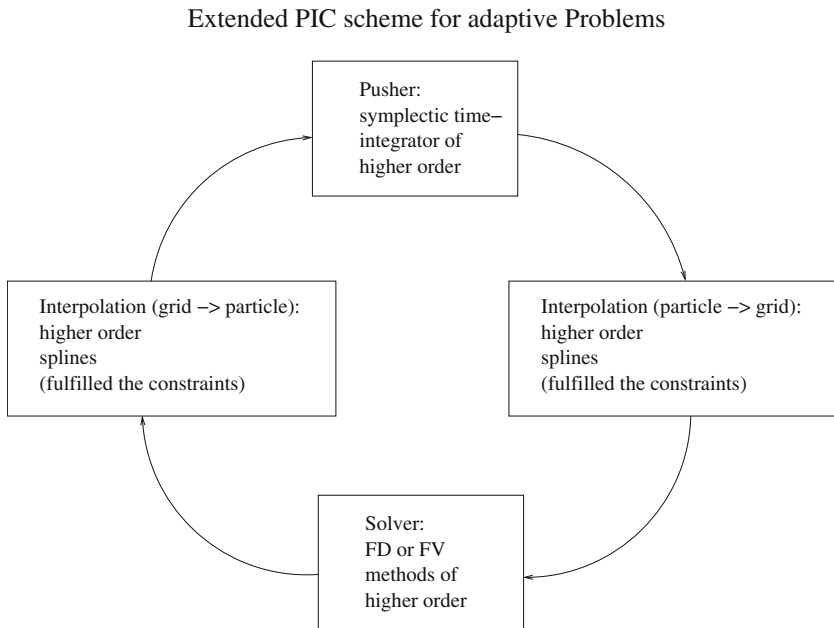


Fig. 4.12 Improved PIC cycles for adaptive PIC

- Solver (higher order discretization schemes, e.g. fourth-order finite difference schemes, see [60]).
- Pusher (higher order symplectic time integrators, e.g. fourth-order symplectic schemes, see [61]).

Remark 4.14 Before improving one part of the PIC cycle, we have to be careful to fulfil the physical constraints of the problem, such that it might be possible, which we have to update all the parts of the PIC cycle for such an extension, see the discussion of an adaptive PIC code in [62].

4.4.3 Mathematical Model

In the following, we discuss the mathematical model, which is based on the Vlasov–Poisson equation, which describe an ideal plasma model.

The Vlasov equation describes the electron distribution f

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{F}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0, \quad (4.185)$$

and the Poisson equation describes the potential to the electrons in the electric field \mathbf{E} :

$$\nabla^2 \phi = -\frac{\rho}{\varepsilon}, \quad (4.186)$$

$$\mathbf{F} = q\mathbf{E} = -q\nabla\phi, \quad (4.187)$$

The positive ions are used as a fixed, neutralizing, background charge density ρ_0 and the total charge density ρ is given as

$$\rho(\mathbf{x}) = q \int f \, d\mathbf{v} + \rho_0, \quad (4.188)$$

We apply the following assumption to the model in the linear case:

- Plasma frequency: $\omega_P = \sqrt{\frac{nq^2}{\varepsilon_0 m_e}}$.
- Debye length: $\lambda_D = \sqrt{\frac{\varepsilon_0 k_B T}{nq^2}}$.

These lengths are important for the explicit numerical schemes, i.e. we have restrictions of the time and spatial step sizes:

- Time step size $\Delta t \ll \frac{2}{\omega_P}$.
- Spatial step size: $\Delta x \leq \lambda_D$.

Furthermore, we have some more conditions:

- Restriction to the length of apparatus L : $\lambda_D \ll L$.
- Number of particle: $N_P \lambda_D \gg L$,

such that we have a sufficient large length of the test apparatus and also to fulfil the number of particle per Debye length which is sufficiently large, where we have from the statistical point of view sufficient dates for the methods, see [54].

4.4.4 Discretization of the Model

To compute the model, we have to apply the idea of a super particle, which allows to decouple into an equation of motion (transport of the particles) and the potential equation (forces to the particles).

We assume that the $x - v$ phase space is divided into a regular array of infinitesimal cells of volume $d\tau = dx dv$, where $d\tau$ is sufficiently small so that only one electron is in it. Then $f(x, v, t)d\tau$ gives the probability that the cell at (x, v) is occupied at time t . We assume that the electron is then shifted to time t' to the cell (x', v') . Due to this assumption, it is also used in the characteristics schemes, see [63].

We have to solve the equation of motions:

$$\frac{dx}{dt} = v, \quad (4.189)$$

$$\frac{dv}{dt} = \frac{qE}{m}, \quad (4.190)$$

and in the time integral form

$$x' = x + \int_t^{t'} v dt, \quad (4.191)$$

$$v' = v + \int_t^{t'} \frac{qE}{m} dt, \quad (4.192)$$

and we can show in general for such a shift: $f(x', v', t') = f(x, v, t)$.

To speed up the computations, we take a sample of points (super particle) $\{x_i, v_i, i = 1, \dots, N_s\}$ and an element i of the phase fluid is corresponding to

$$N_s = \int_i f dx dv, \quad (4.193)$$

The characteristics to the phase space of the super particle points are given by

$$\frac{dx_i}{dt} = v_i, \quad (4.194)$$

$$\frac{dv_i}{dt} = \frac{F(x_i)}{M}, \quad (4.195)$$

$M = N_s m_e$ and m_e is the electron mass.

In the following, we discuss the different extensions to the adaptive PIC methods.

4.4.4.1 1D Adaptive PIC

To understand the parts of the adaptive PIC method, we discuss in the first steps the one-dimensional case. In the following, we describe the different tools for the 1D adaptive PIC:

- 1 D adaptive finite difference (FD) method,
- 1 D adaptive Shape function, and
- Fitting scheme at the interface.

While the 1D FD methods are applied to the micro- and macro-model, we apply also adaptive interpolation/restriction methods, i.e. shape functions, to apply the data transfer between the different scales. Furthermore, we have to deal with a fitting scheme at the interface to fulfil the constraints of the scheme, e.g. conserve the first moments of the shape functions, see [62].

1D Adaptive Finite Difference Discretization for the Poisson Equation

In the following, we have the adaptive scheme, which are based on weighting the central difference scheme for the underlying model problem, i.e. here the Poisson's equation.

We discuss the adaptive grid of finite difference schemes, see [64], for the Poisson equation in one dimension:

$$\frac{d^2\phi}{dx^2} = -\frac{1}{\varepsilon_0}\rho(x_i), \quad x_i \in [0, L], \quad (4.196)$$

$$\phi(0) = 0, \phi(L) = 0, \quad (4.197)$$

where x_i give the coordinates of a super particle i .

The finite difference scheme after Shortly and Weller [65], which is given with a three-point stencil, see also [66], and the difference quotient are given as

$$-D_{\Delta x}^2 \phi = \frac{2}{\Delta x^2} \left[\frac{1}{s_r(s_r + s_l)} \phi(x + s_r \Delta x) + \frac{1}{s_l(s_r + s_l)} \phi(x - s_l \Delta x) - \frac{1}{s_r s_l} \phi(x) \right], \quad (4.198)$$

where Δx is the mesh size of the grid and $s_r, s_l \in (0, 1]$ are the scaled factors of the finer grid. Furthermore, $D_{\Delta x}^2 = \partial_{s_r \Delta x}^+ \partial_{s_l \Delta x}^-$ is the difference quotient with

$$\partial_{s_l \Delta x}^+ \phi = \frac{\phi(x) - \phi(x - s_l \Delta x)}{s_l \Delta x}, \quad (4.199)$$

$$\partial_{s_r \Delta x}^- \phi = \frac{\phi(x + s_r \Delta x) - \phi(x)}{s_r \Delta x}. \quad (4.200)$$

The consistency error is given for the boundary points also as a second-order method, see [66]:

$$\|\phi(x) - \phi_{\Delta x}(x)\| \leq \Delta x^2 \left(\frac{1}{48} d^2 \|\phi\|_{C^{3,1}(\bar{\Omega})} + \frac{2}{3} \|\phi\|_{C^{2,1}(\bar{\Omega})} \right), \quad (4.201)$$

where $d \leq \Delta x$.

Remark 4.15 For a different notation, we apply

$$\begin{aligned} D_{\Delta x} D_{\Delta \tilde{x}} \phi &= -\frac{2}{\Delta x(\Delta x + \Delta \tilde{x})} \phi(x + \Delta x) + \frac{2}{\Delta x \Delta \tilde{x}} \phi(x) \\ &\quad - \frac{2}{\Delta \tilde{x}(\Delta x + \Delta \tilde{x})} \phi(x - \Delta \tilde{x}), \end{aligned} \quad (4.202)$$

while Δx is the grid size on the left-hand side and $\Delta \tilde{x}$ is the grid size on the right-hand side.

Adaptive Shape Functions

In the following, we derive adaptive higher order shape functions.

We have the following underlying steps for the construction:

1. 1D Interpolation and Shape functions.
2. 1D uniform shape functions.
3. Adaptive Linear Splines (adaptive CIC).
4. Construction of higher order Splines.

The steps are discussed in the following outlined points.

1. *1D Interpolation and Shape functions:*

In the following, we discuss the shape functions that are need to map the charge densities on a grid.

We deal with CIC (Cloud in Cell) shape functions, see [54], which are the linear shape functions $S(x_i - X_j)$, where X_j implements the grid point and x_i is the position of the particle i .

The density at the grid point of the particles is weighted by the weighting function:

$$\rho_j = \sum_{i=1}^N q_i S(x_i - X_j), \quad (4.203)$$

where q_i is the i th charge.

In standard application, this function is symmetry and fulfils the isotropy of space, charge conservation and condition to avoid self forces, see [54, 67].

For the consistency of the uniform and nonuniform shape functions, we have the following restriction:

$$\sum_{i=1}^N S(x - X_i) = 1, \quad (4.204)$$

where all the weights are $q_i = 1$ and x is the position of the particle and X_i the grid point at position i .

In the following, we see the construction on a non-symmetric mesh, see Fig. 4.13.

2. 1D uniform shape function

We deal with the following uniform shape functions:

- NGP: nearest grid point and
- CIC: Cloud in Cell.

The NGP uniform shape functions is given as

$$S(x - X) = \begin{cases} 1, & \text{when } |x - X| \leq \frac{\Delta x}{2}, \\ 0, & \text{else,} \end{cases} \quad (4.205)$$

where we have a uniform grid size of Δx in the domain $\Omega = [0, L]$.

The CIC uniform shape functions is given as

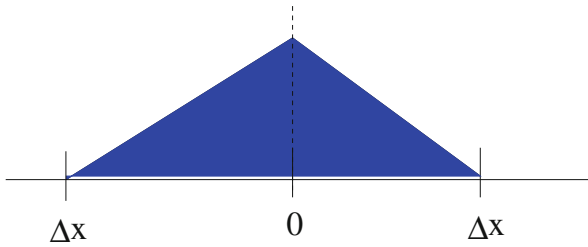
$$S(x - X) = \begin{cases} 1 - \frac{|x-X|}{\Delta x}, & \text{when } |x - X| < \Delta x, \\ 0, & \text{else,} \end{cases} \quad (4.206)$$

where we have a uniform grid size of Δx in the domain $\Omega = [0, L]$.

For the uniform mesh function, we have to fulfil the consistency (mass conservation) (4.204).

Theorem 4.5 *For the uniform shape function (4.206), we fulfil the consistency (4.204).*

Uniform Shape Function



Adaptive Shape Function

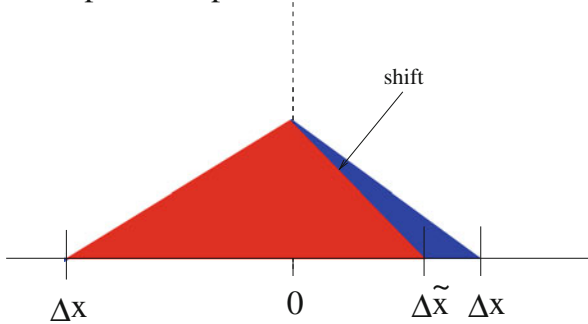


Fig. 4.13 Adaptive shape function

Proof It is sufficient to proof the function for the following situation for one particle x and the two grid points X and $X - \Delta x$, based on the symmetry, and one can do it for all particles:

$$1 - \frac{x - (X - \Delta x)}{\Delta x} + 1 - \frac{(X - x)}{\Delta x} = 1, \tag{4.207}$$

$$2 - \frac{x}{\Delta x} - \frac{X}{\Delta x} - 1 - \frac{X}{\Delta x} + \frac{x}{\Delta x} = 1, \tag{4.208}$$

this is fulfilled.

3. Adaptive Linear Splines (adaptive CIC)

In the following, we discuss the adaptive shape functions.

We assume the domain $\Omega = [0, L]$ and Δx is operating in the domain $[0, L_1]$, while $\Delta \tilde{x}$ is operating in the domain $[L_1, L]$.

The grid point X is not at the boundary L_1 , i.e. $x < L_1 - \Delta x$ or $x > L_1 + \Delta \tilde{x}$:

$$S(x - X) = \begin{cases} 1 - \frac{|x-X|}{\Delta x}, & \text{when } |x - X| < \Delta x, \text{ and } x \in [0, L_1], \\ 1 - \frac{|x-X|}{\Delta \tilde{x}}, & \text{when } |x - X| < \Delta \tilde{x}, \text{ and } x \in [L_1, L], \\ 0, & \text{else,} \end{cases} \quad (4.209)$$

where we assume to have a nonuniform grid size, while of Δx is the domain $\Omega = [0, L_1]$ and $\Delta \tilde{x}$ is the domain $\Omega = [L_1, L]$.

For the nonuniform mesh function, we have to fulfil the consistency (mass conservation) (4.204).

Theorem 4.6 *For the nonuniform shape function (4.285), we fulfil the consistency (4.204).*

Proof It is sufficient to prove that the shape functions based on each different domain fulfil the condition.

For domain $\Omega_1 = [0, L_1]$, we have

$$1 - \frac{x - (X - \Delta x)}{\Delta x} + 1 - \frac{(X - x)}{\Delta x} = 1, \quad (4.210)$$

and when it is fulfilled also for domain $\Omega_2 = [L_1, L]$, we have

$$1 - \frac{x - (X - \Delta \tilde{x})}{\Delta \tilde{x}} + 1 - \frac{(X - x)}{\Delta \tilde{x}} = 1. \quad (4.211)$$

Remark 4.16 The idea of the adaptive shape functions can also be extended to higher order shape functions, e.g. [54]. An example is given in the appendix.

4. Construction of the higher order Spline

We have the following situation of the shape functions, see in Fig. 4.14.

Following [54], we have the following constraints to derive the higher shape functions:

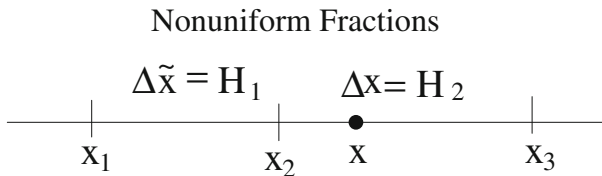


Fig. 4.14 Nonuniform fractions for the shape functions (nonuniform TSC function)

$$\sum_{p=1}^m W_p(x) = 1, \quad (4.212)$$

$$\sum_{p=1}^m W_p(x)(x - x_p)^n = \text{const}. \quad (4.213)$$

The additional obtained freedom degrees can be used to approximate to the correct potential ϕ_c .

We improve the interpolation by the fact that

$$\phi(x') = G(x' - x) + \frac{C}{2} \frac{d^2 G(x' - x)}{dx^2} + O(\Delta^3), \quad (4.214)$$

where G is the Greens function and $\phi(x')$ is the correct potential at x' .

Later, we could apply the freedom degree with C to the spline fitting of the adaptive grids.

Due to the fact that ϕ and G are even functions, we have the following restriction of our constraint:

$$\sum_{p=1}^m W_p(x)(x - x_p)^n = \begin{cases} 0 & n \text{ odd} \\ \text{const}, & n \text{ even} \end{cases} \quad (4.215)$$

where

$$W_p = 0, \quad p \neq 1, 2, \dots, n. \quad (4.216)$$

Furthermore, the displacement invariance property is given as

$$W_p(x) = W(x - x_p). \quad (4.217)$$

Example 4.2 In the following, we derive the uniform and nonuniform shape functions.

1. Uniform Case

We derive the case of $n = 2$ and $n = 3$.

For $n = 2$, we have three constraint equations:

$$W_1 + W_2 + W_3 = 1, \quad (4.218)$$

$$W_1 x_1 + W_2 x_2 + W_3 x_3 = x, \quad (4.219)$$

$$W_1 x_1^2 + W_2 x_2^2 + W_3 x_3^2 = C + x^2, \quad (4.220)$$

$$W_p = 0 \text{ for } p \neq 1, 2, 3, \quad (4.221)$$

additional, we have to apply to derive the constant C :

$$\phi(x') = G(x' - x) + \frac{C}{2} \frac{d^2 G(x' - x)}{dx^2} + O(\Delta^3), \quad (4.222)$$

where G is the Greens function and $\phi(x')$ is the correct potential at x' .

For solving the linear equation system, we applied program-code Maxima [68] and we obtain

$$W_1 = \frac{C + (x_2 - x) x_3 - x x_2 + x^2}{(x_2 - x_1) x_3 - x_1 x_2 + x_1^2}, \quad (4.223)$$

$$W_2 = -\frac{C + (x_1 - x) x_3 - x x_1 + x^2}{(x_2 - x_1) x_3 - x_2^2 + x_1 x_2}, \quad (4.224)$$

$$W_3 = \frac{C + (x_1 - x) x_2 - x x_1 + x^2}{x_3^2 + (-x_2 - x_1) x_3 + x_1 x_2}. \quad (4.225)$$

Using the displacement invariance property (4.217) and Eq. (4.216), we obtain

$$W(x) = \begin{cases} \frac{x^2+3 H x+2 H^2+C}{2 H^2}, & -\frac{3}{2} H \leq x < -\frac{1}{2} H, \\ \frac{1-(x^2+C)}{H^2}, & -\frac{1}{2} H \leq x < \frac{1}{2} H, \\ \frac{x^2-3 H x+2 H^2+C}{2 H^2}, & \frac{H}{2} < x < \frac{3 H}{2}, \\ 0, & \text{else.} \end{cases} \quad (4.226)$$

2. For $n = 3$, we have three constraint equations:

$$W_1 + W_2 + W_3 + W_4 = 1, \quad (4.227)$$

$$W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_4 = x, \quad (4.228)$$

$$W_1 x_1^2 + W_2 x_2^2 + W_3 x_3^2 + W_4 x_4^2 = C + x^2, \quad (4.229)$$

$$W_1 x_1^3 + W_2 x_2^3 + W_3 x_3^3 + W_4 x_4^3 = 3x C + x^3, \quad (4.230)$$

$$W_p = 0 \text{ for } p \neq 1, 2, 3, 4, \quad (4.231)$$

additionally, we have to apply to derive the constant C :

$$\phi(x') = G(x' - x) + \frac{C}{2} \frac{d^2 G(x' - x)}{dx^2} + O(\Delta^3), \quad (4.232)$$

where G is the Greens function and $\phi(x')$ is the correct potential at x' .

For solving the linear equation system, we applied program-code Maxima [68] and we obtain

$$\begin{aligned}
W_1 &= \frac{x_4 (C + (x_2 - x) x_3 - x x_2 + x^2) + x_3 (C - x x_2 + x^2) + x_2 (C + x^2) - 3 x C - x^3}{((x_2 - x_1) x_3 - x_1 x_2 + x_1^2) x_4 + (x_1^2 - x_1 x_2) x_3 + x_1^2 x_2 - x_1^3}, \\
W_2 &= -\frac{x_4 (C + (x_1 - x) x_3 - x x_1 + x^2) + x_3 (C - x x_1 + x^2) + x_1 (C + x^2) - 3 x C - x^3}{((x_2 - x_1) x_3 - x_2^2 + x_1 x_2) x_4 + (x_1 x_2 - x_2^2) x_3 + x_2^3 - x_1 x_2^2}, \\
W_3 &= \frac{x_4 (C + (x_1 - x) x_2 - x x_1 + x^2) + x_2 (C - x x_1 + x^2) + x_1 (C + x^2) - 3 x C - x^3}{(x_3^2 + (-x_2 - x_1) x_3 + x_1 x_2) x_4 - x_3^3 + (x_2 + x_1) x_3^2 - x_1 x_2 x_3}, \\
W_4 &= -\frac{x_3 (C + (x_1 - x) x_2 - x x_1 + x^2) + x_2 (C - x x_1 + x^2) + x_1 (C + x^2) - 3 x C - x^3}{x_4^3 + (-x_3 - x_2 - x_1) x_4^2 + ((x_2 + x_1) x_3 + x_1 x_2) x_4 - x_1 x_2 x_3}.
\end{aligned}$$

Using the displacement invariance property (4.217) and Eq. (4.216), we obtain

$$W(x) = \begin{cases} \frac{-x^3 + 6H(x^2 + C) - 11H^2x - 3Cx + 6H^3}{6H^3}, & -2H \leq x < -H, \\ \frac{x^3 - 2H(x^2 + C) - H^2x + 3Cx + 2H^3}{2H^3}, & -H \leq x < 0, \\ \frac{-x^3 - 2H(x^2 + C) + H^2x - 3Cx + 2H^3}{2H^3}, & 0 \leq x < H, \\ \frac{x^3 + 6H(x^2 + C) + 11H^2x + 3Cx + 6H^3}{6H^3}, & H \leq x < 2H, \\ 0, & \text{else.} \end{cases} \quad (4.233)$$

A simpler notation because of the symmetry is given as

$$W(x) = \begin{cases} \frac{-|x|^3 - 2H(x^2 + C) + H^2|x| - 3C|x| + 2H^3}{2H^3}, & |x| < H, \\ \frac{|x|^3 + 6H(x^2 + C) + 11H^2|x| + 3C|x| + 6H^3}{6H^3}, & H \leq |x| < 2H, \\ 0, & \text{else.} \end{cases} \quad (4.234)$$

Such shape function can be applied for the higher order interpolations between grid-free (pusher) and grid parts (solver).

2. Nonuniform Case

We derive the cases for $n = 2$, and the same idea is also applied for $n = 3$.

For $n = 2$, we have three constraint equations:

$$W_1 + W_2 + W_3 = 1, \quad (4.235)$$

$$W_1x_1 + W_2x_2 + W_3x_3 = x, \quad (4.236)$$

$$W_1x_1^2 + W_2x_2^2 + W_3x_3^2 = C + x^2, \quad (4.237)$$

$$W_p = 0 \text{ for } p \neq 1, 2, 3, \quad (4.238)$$

additionally, we have to apply to derive the constant C :

$$\phi(x') = G(x' - x) + \frac{C}{2} \frac{d^2 G(x' - x)}{dx d\tilde{x}} + O(\Delta^3), \quad (4.239)$$

where G is the Greens function and $\phi(x')$ is the correct potential at x' . The adaptive Laplacian is given as $\frac{d^2}{dx d\tilde{x}}$, as given in Eq.(4.202).

We deal with the discussion of the smoothness constraint, which we have as an upper bound of our C . For the uniform grid, the discussion is done in [54].

The effects of the charge assignment are given as

$$\phi_p = \frac{1}{8} \frac{2H_1}{H_1 + H_2} G_{p+H_2} + \frac{3}{4} G_p + \frac{1}{8} \frac{2H_2}{H_1 + H_2} G_{p-H_1}, \quad (4.240)$$

and we obtain

$$\begin{aligned} \phi_p &= G_p + \frac{H_1 H_2}{8} \left(\frac{2H_2}{H_1 + H_2} G_{p+H_2} - \frac{2}{H_1 H_2} G_p + \frac{2H_1}{H_1 + H_2} G_{p-H_1} \right), \\ \phi_p &= G_p + \frac{C}{2} \frac{d}{dx} \frac{d}{d\tilde{x}} G_p, \end{aligned} \quad (4.241)$$

and we obtain $C = \frac{H_1 H_2}{4}$.

Next, we solve the linear equation system that we applied program-code Maxima [68] and we obtain

$$W_1 = \frac{C + (x_2 - x) x_3 - x x_2 + x^2}{(x_2 - x_1) x_3 - x_1 x_2 + x_1^2}, \quad (4.242)$$

$$W_2 = -\frac{C + (x_1 - x) x_3 - x x_1 + x^2}{(x_2 - x_1) x_3 - x_2^2 + x_1 x_2}, \quad (4.243)$$

$$W_3 = \frac{C + (x_1 - x) x_2 - x x_1 + x^2}{x_3^2 + (-x_2 - x_1) x_3 + x_1 x_2}. \quad (4.244)$$

Using the displacement invariance property (4.217) and Eq.(4.216), we obtain

$$W(x) = \begin{cases} \frac{x^2 - 2H_1x + H_2(H_1 - x) + H_1^2 + C}{H_1(H_1 + H_2)}, & -\frac{3H_1}{2} < x < -\frac{H_1}{2}, \\ \frac{-x^2 + H_2(x + H_1) - H_1x - C}{H_1H_2}, & -\frac{H_1}{2} < x < -\frac{H_2}{2}, \\ \frac{x^2 + 2H_2x + H_1(H_2 + x) + H_2^2 + C}{H_2(H_1 + H_2)}, & \frac{H_2}{2} < x < \frac{3H_2}{2}, \\ 0, & \text{else,} \end{cases} \quad (4.245)$$

where $H_1 = x_2 - x_1$, $H_2 = x_3 - x_2$, $H_1 + H_2 = x_3 - x_1$ and $C \in [0, \frac{H_1H_2}{4}]$. So we deal with an adaptive interface with grid lengths H_1 and H_2 .

4.4.4.2 Correction of the Shape Function

For the physical constraints, it has to be fulfilled in the shape functions, and therefore we have additional algorithms to correct the derived shape functions:

- Algorithm 1 (Multigrid idea) for corrected shape function,
- Algorithm 2 (Fixpoint idea) for corrected shape function,
- Improved Pusher: Velocity Verlet,
- Momentum conserved constraint, and
- Spline fitting to fulfil the momentum conservation.

Algorithm 1 (Multigrid Idea) for Corrected Shape Function

In the following, we present the algorithm of the corrected shape function. This is an initialization process, which we have to do first one and afterwards we have at the interface such a corrected shape function.

Algorithm 4.7 (1) Compute the corrected potential at the interface with the fine grid:

$$\phi_{fine}(x') = W_{2,fine}(x)G(x' - x), \quad (4.246)$$

where G is the Greens function (which is given locally in 2D or 3D) and $\rho(x) = 1$, i.e. we assume $W_{2,fine}(x) = 1$.

(2) Compute the uncorrelated potential at the interface with the coarse-grid local (quadratic spline with an assumed $C = C_{uncorrelated}$, e.g. $C_{uncorrelated} = (\Delta x/2)^2$.)

$$\phi_{coarse,uncorrelated}(x') = W_{2,coarse}(x)G(x' - x), \quad (4.247)$$

where G is the Greens function (which is given locally in 2D or 3D) and $\rho(x) = 1$, but we have a different shape function based on the adaptation $W_{2,coarse}(x) \neq W_{2,fine}(x) = 1$.

(3) Compute the corrected adaptive shape function (compute the parameter C):

$$\begin{aligned}
 \phi_{coarse,correlated} &= \frac{C}{2} W_{2,coarse}(x) \frac{\partial}{\partial x} \frac{\partial}{\partial \bar{x}} G + W_{2,coarse}(x) G \\
 &= \frac{C}{2} W_{2,coarse}(x) \frac{\partial}{\partial x} \frac{\partial}{\partial \bar{x}} G + \phi_{coarse,uncorrelated}(x') \\
 &= \phi_{fine}(x'), \tag{4.248}
 \end{aligned}$$

and we have

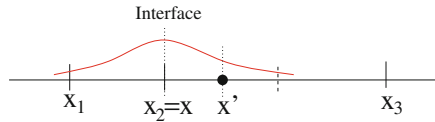
$$C = 2 \frac{\phi_{fine}(x') - \phi_{coarse,uncorrelated}(x')}{W_{2,coarse}(x) \frac{\partial}{\partial x} \frac{\partial}{\partial \bar{x}}}. \tag{4.249}$$

For the initialization of the interface, we have first computed this corrected shape function, and if we do not change the interface afterwards, we could use the fitted spline for all the particles.

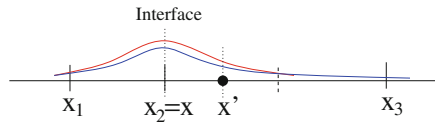
In the following, we describe the spline fitting algorithm in Fig. 4.15.

Spline-Fitting

1.) Computation at Interface with fine shape function (red: correct Potential)



2.) Computation at Interface with coarse shape function (blue: uncorrelated Potential)



3.) Computation at Interface with corrected coarse shape function (green: correlated Potential)

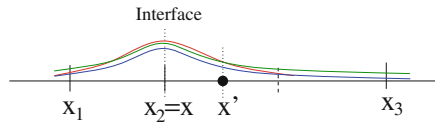
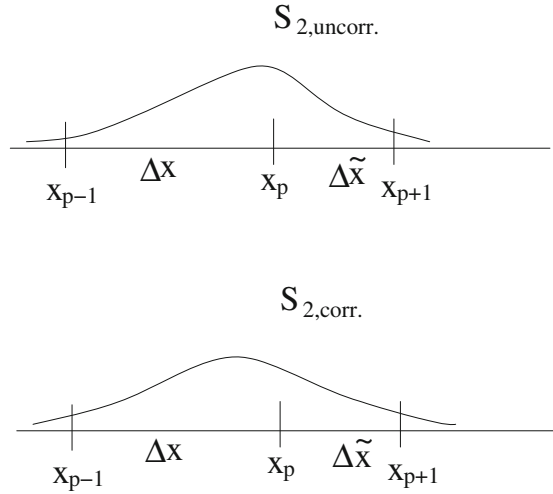


Fig. 4.15 Spline fitting: fine-coarse interface at interface point x

Correlation shape function and discretization



Correlation shape function with respect to the adaptive grid

Fig. 4.16 Correlated shape functions to the adaptive discretization scheme (adapted TSC function)

In Fig. 4.16, we present the correlated shape function with respect to the adaptive discretization scheme.

Algorithm 2 (Fixpoint Idea) for Corrected Shape function

In the following, we present an alternative algorithm of the corrected shape function based on forward and backward computations at the interface, which can be formulated to a fixpoint scheme.

The algorithm is given as follows.

Algorithm 4.8 We start with known $x_i^n, x_p, v_i^{n-1/2}$ and $C_0 = 0$

- (1) Forward PIC algorithm starting with x_i^n and $+q$ (positive charge)

$$x_i^n \rightarrow \rho_p^n \rightarrow \phi_p^n \rightarrow E_p^n \rightarrow F_i^n \rightarrow v_i^{n+1/2} \rightarrow x_i^{n+1} \tag{4.250}$$

- (2) Backward PIC algorithm starting with x_i^{n+1} and $-q$ (negative charge)

$$x_i^{n+1} \rightarrow -\rho_p^{n+1} \rightarrow -\phi_p^{n+1} \rightarrow -E_p^{n+1} \rightarrow -F_i^{n+1} \rightarrow -\tilde{v}_i^{n+1/2} \rightarrow \tilde{x}_i^n \tag{4.251}$$

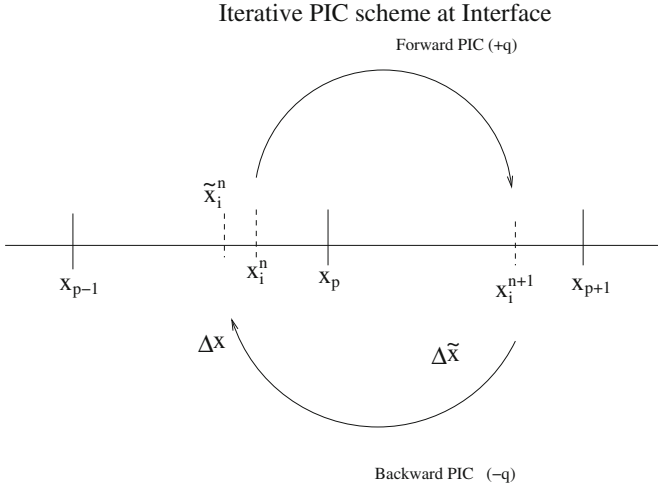


Fig. 4.17 Iterative PIC (forward and backward computations with PIC at the interface)

(3) Difference Forward PIC and Backward PIC algorithm

$$\Delta x_i = x_i^n - \tilde{x}_i^n, \tag{4.252}$$

(4) Adaptation of parameter C_j

We compute the error of the schemes (forward, backward)

$$|W(x_i^n - x_p, C_{j-1}) - W(\tilde{x}_i^n - x_p, C_{j-1})| = \delta W, \tag{4.253}$$

if $\Delta W \leq error$, we are done and C_{j-1} is our novel parameter for the shape function else we compute C_j with

$$W(x_i^n - x_p, C_j) - W(\tilde{x}_i^n - x_p, C_{j-1}) = 0, \tag{4.254}$$

and go to step(1)

In Fig. 4.17, we see the idea of the iterative forward and backward PIC scheme.

Improved Pusher: Velocity Verlet

For the backward PIC, we have a problem in computing the backward velocity $\tilde{v}_i^{n+1/2}$, with the simple leap-frog algorithm, see [69, 70], and we have to apply $v_i^{n+3/2}$ which is not given.

Here, an improved second-order scheme to compute also backward a PIC algorithm.

We have to apply the velocity Verlet, which is given as a forward scheme $x_n \rightarrow x_{n+1}$ (x_n, v_n is known)

$$v_{n+1/2} = v_n + \frac{1}{2} \Delta t F(x_n), \tag{4.255}$$

$$x_{n+1} = x_n + \Delta t v_{n+1/2}, \tag{4.256}$$

$$v_{n+1} = v_{n+1/2} + \frac{1}{2} \Delta t F(x_{n+1}), \tag{4.257}$$

or a backward scheme $\tilde{x}_{n+1} \rightarrow \tilde{x}_n$ ($\tilde{x}_{n+1}, \tilde{v}_{n+1}$ is known):

$$\tilde{v}_{n+1/2} = \tilde{v}_{n+1} - \frac{1}{2} \Delta t F(\tilde{x}_{n+1}), \tag{4.258}$$

$$x_n = \tilde{x}_{n+1} - \Delta t \tilde{v}_{n+1/2}, \tag{4.259}$$

$$\tilde{v}_n = \tilde{v}_{n+1/2} - \frac{1}{2} \Delta t F(\tilde{x}_n). \tag{4.260}$$

Remark 4.17 Higher order schemes with respect to magnetic and electric field can be obtained by extrapolation schemes [71] or cyclotronic integrators [61].

Momentum Conserved Constraint

Idea of spline fitting, see [54], reduces the spatially localized errors based on the adaption at the interface.

We are motivated to embed higher order shape and discretization functions to reduce the local error at the interface of the adaptation.

In book of Hockney [54], the higher order shape functions are introduced to fit at the long-range constraints, and we apply them as a freedom degree to the adaptive grids, see Fig. 4.18.

The full PIC cycle is given as follows (discrete model):

- (1) Charge assignment (Method: Spline functions):

$$\rho_p^n = \frac{q}{H} \sum_{i=1}^{N_p} W(x_i - x_p) \tag{4.261}$$

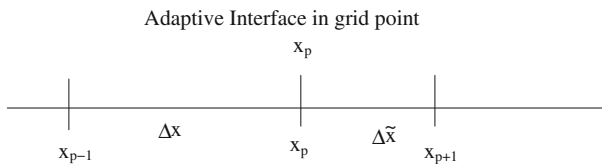


Fig. 4.18 Adaptive interface

(2) Field equation (Method: Solver)

We have to solve

$$\nabla^2 \phi_p = -\frac{\rho_p}{\varepsilon_0} \quad (4.262)$$

We obtain the notation with the Greens function:

$$\phi_{p,h} = \sum_j \rho_{p',h} G_{h,H}(p, p') \quad (4.263)$$

while we have a discrete Greens function, see the idea of the composite grids, [54].

The discrete analogue of the Greens function to the adaptive finite difference scheme is given as

$$G_{h,H}(\cdot, e_{\Gamma_h^*}) = A_{h,H}^{-1} e_{\Gamma_h^*} \quad (4.264)$$

where $\Gamma_h^* = \Gamma - h$.

If we have not a translation invariant matrix, we have also a non-translation invariant inverse matrix, and therefore also the discrete Greens function is not translation invariant.

The we discretize the electric field with

$$E_{p,h} = \sum_s a_s (\phi_{p+s,h} - \phi_{p-s,h}) \quad (4.265)$$

where a_s is the coefficient for the finite difference discretization.

(3) Force interpolation (Method: Spline functions)

$$F(x_i) = \sum_p W(x_i - x_p) F_p \quad (4.266)$$

(4) Equation of motion (Method: Pusher)

$$v_{i+1/2} = v_{i-1/2} + \frac{\delta t}{2} F(x_i), \quad (4.267)$$

$$x_{i+1} = x_i + \delta t v_{i+1/2}, \quad (4.268)$$

$$v_{i+1} = v_{i+1/2} + \frac{\delta t}{2} F(x_{i+1}). \quad (4.269)$$

Based on the PIC cycle, we fulfil the following constraints and conserve the momentum.

Based on the ideas of [54, 57], the conditions

- Identical charge assignment, and
- Correctly space-centred finite difference approximations, while we have the condition

$$d(x_p; x_{p'}) = -d(x_{p'}, x_p) \quad (4.270)$$

are sufficient to fulfil the self-force and inter-particle force, and therefore the momentum constraint.

While we deal with adaptive grids, the constraint 2 (4.270) is only fulfilled with uniform grids.

We propose the following constraint, which is a combination of constraints 1 and 2, while we balance between the freedom degree of the shape functions:

$$d(x_p - x_{p'})W(x_p - x_{p'}, C_{opt}) = -d(x_{p'} - x_p)W(x_{p'} - x_p, C_{opt}) \quad (4.271)$$

$$C_{opt} \in [0, H^2/4].$$

Then the momentum conservation is given with respect to the self-force and inter-particle force.

Spline Fitting to Fulfil the Momentum Conservation

We obtain the following approach to the Greens function:

$$\begin{aligned} G_{ex,q}^p &= G_q^p + C_{1,exact} \Delta G_q^p \phi(x') \\ &= G_{q-e^d h_l}^{p-e^d h_l} + C_{2,exact} \Delta G_{q-e^d h_l}^{p-e^d h_l} = G_{ex,q-e^d h_l}^{p-e^d h_l}, \end{aligned} \quad (4.272)$$

where $C_{1,exact} = |p - q|/2$, $C_{2,exact} = |(p - e^d h_l) - (q - e^d h_l)|/2$.

To obtain the translation invariant $G_{ex,q}^p = G_{ex,q-e^d h_l}^{p-e^d h_l}$, we have to fit

$$G_q^p + C_1 \Delta G_q^p \phi(x') = G_{q-e^d h_l}^{p-e^d h_l} + C_2 \Delta G_{q-e^d h_l}^{p-e^d h_l}, \quad (4.273)$$

$$G_{q-e^d h_l}^{p-e^d h_l} = G_q^p + C_1 \Delta G_q^p \phi(x') - C_2 \Delta G_{q-e^d h_l}^{p-e^d h_l} \quad (4.274)$$

and we can fit C_1 and C_2 to have a translation invariant function G_q^p . Furthermore, C_1 and C_2 have fulfilled the adaptive higher order discretization scheme.

Remark 4.18 Here, we apply the similar ideas as in [72] for AMR (adaptive mesh refinement). While we are only approaching to one interface and we deal with higher order shape functions, we are more flexible to derive the constants C_1 and C_2 .

4.4.5 2D Adaptive PIC

In the following, we discuss the extension to the two-dimensional particle in cell method based on adaptive schemes. Here, we have the influence of the higher dimensions to the discretization and shape functions.

In the following, we describe the different tools for the 2d adaptive PIC:

- 2D discretization scheme based on finite difference methods for the different equations (e.g. Maxwell and Newton equation),
- Shape functions:
 - 2D Shape functions (general introduction),
 - 2D adaptive Shape function (linear functions), and
 - 2D adaptive Shape function (quadratic functions).

Remark 4.19 The discretization and solver schemes are similar to the 1D problem. Based on the FD method, we have only to increase the standard method to a two-dimensional scheme, see [64], and apply the linear equation systems to the solver methods, see [54]. More important are the modifications related to the shape functions, which connect the different models (microscopic and macroscopic model).

In the following, we concentrate on the 2D shape functions.

2D Shape Functions (General Introduction)

In the following, we describe higher order shape function for 2D problems.

We can extend the idea of the derivation of the shape function to higher dimensions, in the following, we discuss the 2D shape functions.

Constraints for the two-dimensional shape functions n th order

For n th order, we have $n + 1$ constraint equations:

$$\sum_{\mathbf{P}} W_{\mathbf{P}} = 1, \text{ charge conservation,} \quad (4.275)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_i = 0, \text{ first order,} \quad (4.276)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_i \Delta_j = C_1 \delta_{ij}, \text{ second order,} \quad (4.277)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_i \Delta_j \Delta_k = 0, \text{ third order,} \quad (4.278)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_{i_1} \Delta_{i_2} \Delta_{i_3} \Delta_{i_4} = C_2 \delta_{i_1, i_2, i_3, i_4}, \text{ fourth order,} \quad (4.279)$$

$$\vdots, \quad (4.280)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_{i_1} \Delta_{i_2} \dots \Delta_{i_n} = 0, \text{ nth order (n odd),} \quad (4.281)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_{i_1} \Delta_{i_2} \dots \Delta_{i_n} = C_{n/2} \delta_{i_1, i_2, \dots, i_n}, \text{ nth order (n even),} \quad (4.282)$$

where $\mathbf{p} = (p_1, p_2)$ is a pair labelling the mesh point \mathbf{p} at position $\mathbf{x}_{\mathbf{p}}$. The expansion

of the additional constant C is given as

$$\phi(\mathbf{x}') = \sum_{\mathbf{P}} W_{\mathbf{P}}(\mathbf{x}) \sum_{r,s=0}^{\infty} \frac{\Delta_1^r \Delta_2^s}{r! s!} \frac{\partial^{r+s} G}{\partial x^r \partial y^s}, \tag{4.283}$$

where G is the Greens function and $\phi(x')$ is the correct potential at x' .

2D Adaptive Shape Function (Linear Function), Linear Spline $n = 1$, CIC Adaptive for 2D

In the following, we discuss the two ideas to create 2D shape functions for the two-dimensional case:

- Local one-dimensional (splitting in the locally dimensions).
- Full two-dimensional (non-splitting of the locally dimensions).

We discuss the different approximations.

- One-dimensional Local

In the following, we discuss the adaptive shape functions.

Assumption 4.9 We assume that the dimensions can be separated and the shape functions can be constructed as locally one-dimensional problems:

$$W(x, y) = P(x)P(y) \tag{4.284}$$

We assume a four-point stencil for the adaptive finite difference scheme.

We assume the domain $\Omega = [0, L_1] \times [0, L_2]$. In the adaptive grid, we assume that Δx is operating in the domain $[0, L_{1,1}] \times [0, L_2]$, while $\Delta \tilde{x}$ is operating in the domain $[L_{1,1}, L_1] \times [0, L_2]$. Furthermore, we assume that Δy is operating in the domain $[0, L_1] \times [0, L_{2,1}]$, while $\Delta \tilde{y}$ is operating in the domain $[0, L_1] \times [L_{2,1}, L_2]$. We have the following shape function:

$$S(\mathbf{x} - \mathbf{X}) = \begin{cases} \left(1 - \frac{|x-X|}{\Delta x}\right) \left(1 - \frac{|y-Y|}{\Delta y}\right), & \text{when } |x-X| < \Delta x, |y-Y| < \Delta y \\ & \text{and } (x, y) \in [0, L_{1,1}] \times [0, L_{2,1}], \\ \left(1 - \frac{|x-X|}{\Delta \tilde{x}}\right) \left(1 - \frac{|y-Y|}{\Delta y}\right), & \text{when } |x-X| < \Delta \tilde{x}, |y-Y| < \Delta y \\ & \text{and } (x, y) \in [L_{1,1}, L_1] \times [0, L_{2,1}], \\ \left(1 - \frac{|x-X|}{\Delta x}\right) \left(1 - \frac{|y-Y|}{\Delta \tilde{y}}\right), & \text{when } |x-X| < \Delta x, |y-Y| < \Delta \tilde{y} \\ & \text{and } (x, y) \in [0, L_{1,1}] \times [L_{2,1}, L_2], \\ \left(1 - \frac{|x-X|}{\Delta \tilde{x}}\right) \left(1 - \frac{|y-Y|}{\Delta \tilde{y}}\right), & \text{when } |x-X| < \Delta \tilde{x}, |y-Y| < \Delta \tilde{y} \\ & \text{and } (x, y) \in [L_{1,1}, L_1] \times [L_{2,1}, L_2], \\ 0, & \text{else,} \end{cases} \tag{4.285}$$

where we have $\mathbf{x} = (x, y)^t$.

For the nonuniform mesh function, we have to fulfil the consistency (mass conservation) (4.204).

Theorem 4.10 *For the nonuniform shape function (4.285), we fulfil the consistency (4.204).*

Proof It is sufficient to prove that the shape functions based on each different domain fulfil the condition.

While we can separate to local one-dimensional problem and each dimension is fulfil, see Sect. 4.4.4.1 and we are done.

- Two-dimensional

For $n = 1$, we have three constraint equations:

$$\sum_{\mathbf{P}} W_{\mathbf{P}} = 1, \quad (4.286)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_i = 0, \quad (4.287)$$

where $\mathbf{p} = (p_1, p_2)$ is a pair labelling the mesh point \mathbf{p} at position $\mathbf{x}_{\mathbf{p}}$. We have the following equations:

$$W_1 + W_2 + W_3 = 1, \quad (4.288)$$

$$W_1 x_1 + W_2 x_2 + W_3 x_3 = x, \quad (4.289)$$

$$W_1 y_1 + W_2 y_2 + W_3 y_3 = y. \quad (4.290)$$

By solving Eqs. (4.317)–(4.320), we obtain (using program-code Maxima [68])

$$W_1 = \frac{x (y_3 - y_2) + x_2 (y - y_3) + x_3 (y_2 - y)}{x_1 (y_3 - y_2) + x_2 (y_1 - y_3) + x_3 (y_2 - y_1)}, \quad (4.291)$$

$$W_2 = -\frac{x (y_3 - y_1) + x_1 (y - y_3) + x_3 (y_1 - y)}{x_1 (y_3 - y_2) + x_2 (y_1 - y_3) + x_3 (y_2 - y_1)}, \quad (4.292)$$

$$W_3 = \frac{x (y_2 - y_1) + x_1 (y - y_2) + x_2 (y_1 - y)}{x_1 (y_3 - y_2) + x_2 (y_1 - y_3) + x_3 (y_2 - y_1)}, \quad (4.293)$$

$$\text{for } -\frac{\mathbf{H}_2}{2} \leq \mathbf{x} \leq \mathbf{H}_1. \quad (4.294)$$

where $\mathbf{x} = (x, y)^t$, $\mathbf{H}_1 = (H_{11}, H_{12})^t$ and $\mathbf{H}_2 = (H_{21}, H_{22})^t$.

Using the displacement invariance property (4.217) and Eq. (4.216), we obtain

$$W(x) = \begin{cases} 1 - \frac{x}{H_{11}} - \frac{y}{H_{21}}, & 0 < x < \frac{H_{11}}{2}, 0 < y < \frac{H_{21}}{2}, \\ 1 - \frac{x}{H_{11}} + \frac{y}{H_{22}}, & 0 < x < \frac{H_{11}}{2}, -\frac{H_{22}}{2} < y < 0, \\ 1 + \frac{x}{H_{12}} - \frac{y}{H_{21}}, & -\frac{H_{12}}{2} < x < 0, 0 < y < \frac{H_{21}}{2}, \\ 1 + \frac{x}{H_{12}} + \frac{y}{H_{22}}, & -\frac{H_{12}}{2} < x < 0, -\frac{H_{22}}{2} < y < 0, \\ \\ 1 - \frac{x}{H_{11}}, & \frac{H_{11}}{2} < x < \frac{3H_{11}}{2}, 0 < y < \frac{H_{21}}{2}, \\ 1 - \frac{x}{H_{11}}, & \frac{H_{11}}{2} < x < \frac{3H_{11}}{2}, -\frac{H_{22}}{2} < y < 0, \\ 1 + \frac{x}{H_{12}}, & -\frac{3H_{12}}{2} < x < -\frac{H_{12}}{2}, 0 < y < \frac{H_{21}}{2}, \\ 1 + \frac{x}{H_{12}}, & -\frac{3H_{12}}{2} < x < -\frac{H_{12}}{2}, -\frac{H_{22}}{2} < y < 0, \\ \\ 1 - \frac{y}{H_{21}}, & 0 < x < \frac{H_{11}}{2}, \frac{H_{21}}{2} < y < \frac{3H_{21}}{2}, \\ 1 - \frac{y}{H_{21}}, & -\frac{H_{12}}{2} < x < 0, \frac{H_{21}}{2} < y < \frac{3H_{21}}{2}, \\ 1 + \frac{y}{H_{22}}, & 0 < x < -\frac{H_{11}}{2}, -\frac{3H_{22}}{2} < y < -\frac{H_{22}}{2}, \\ 1 + \frac{y}{H_{22}}, & -\frac{H_{12}}{2} < x < 0, -\frac{3H_{22}}{2} < y < -\frac{H_{22}}{2}, \\ \\ 0, & \text{else,} \end{cases} \quad (4.295)$$

where $\mathbf{H}_1 = (H_{11}, H_{21})^t$ and $\mathbf{H}_2 = (H_{21}, H_{22})^t$. We deal with an adaptive interface with grid lengths \mathbf{H}_1 and \mathbf{H}_2 , given in Fig. 4.19.

2D Adaptive Shape Function (Quadratic Function), Quadratic Splines $n = 2$, CIC Adaptive for 2D

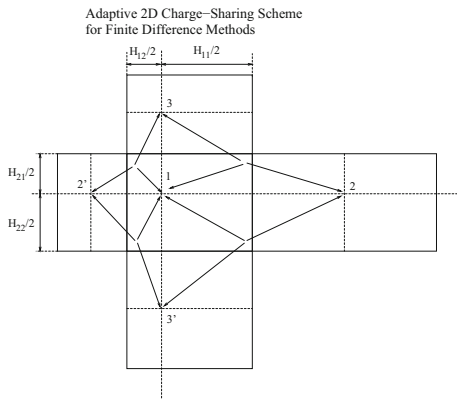
In the following, we discuss the adaptive shape functions.

Assumption 4.11 We assume that the dimensions can be separated and the shape functions can be constructed as locally one-dimensional problems:

$$W(x, y) = P(x)P(y). \quad (4.296)$$

We assume the domain $\Omega = [0, L_1] \times [0, L_2]$. In the adaptive grid, we assume that Δx is operating in the domain $[0, L_{1,1}] \times [0, L_2]$, while $\Delta \tilde{x}$ is operating in the

Fig. 4.19 Two-dimensional adaptive five-point charge-sharing scheme assigns charge to the nearest grid point (labelled 1) and the next-nearest grid points in the east–west direction (labelled 2 and 2') and in the north–south direction (labelled 3 and 3')



domain $[L_{1,1}, L_1] \times [0, L_2]$. Furthermore, we assume that Δy is operating in the domain $[0, L_1] \times [0, L_{2,1}]$, while $\Delta \bar{y}$ is operating in the domain $[0, L_1] \times [L_{2,1}, L_2]$.

We have the following pair of equations for the shape functions.

We have the following equations:

$$P_{1,x} + P_{2,x} + P_{3,x} = 1, \tag{4.297}$$

$$P_{1,x}x_1 + P_{2,x}x_2 + P_{3,x}x_3 = x, \tag{4.298}$$

$$P_{1,x}x_1^2 + P_{2,x}x_2^2 + P_{3,x}x_3^2 = x^2 + C_x, \tag{4.299}$$

and

$$P_{1,y} + P_{2,y} + P_{3,y} = 1, \tag{4.300}$$

$$P_{1,y}y_1 + P_{2,y}y_2 + P_{3,y}y_3 = y, \tag{4.301}$$

$$P_{1,y}y_1^2 + P_{2,y}y_2^2 + P_{3,y}y_3^2 = y^2 + C_y. \tag{4.302}$$

The locally one-dimensional shape functions are given as

$$P_x(x) = \begin{cases} P_{x,1}(x) = \frac{x^2 - 2H_{12}x + H_{11}(H_{12} - x) + H_{12}^2 + C_x}{H_{12}(H_{12} + H_{11})}, & -\frac{3H_{12}}{2} < x < -\frac{H_{12}}{2}, \\ P_{x,2}(x) = \frac{-x^2 + H_{11}(x + H_{12}) - H_{12}x - C_x}{H_{12}H_{11}}, & -\frac{H_{12}}{2} < x < -\frac{H_{11}}{2}, \\ P_{x,3}(x) = \frac{x^2 + 2H_{11}x + H_{12}(H_{11} + x) + H_{11}^2 + C_x}{H_{11}(H_{12} + H_{11})}, & \frac{H_{11}}{2} < x < \frac{3H_{11}}{2}, \\ P_{x,4}(x) = 0, & \text{else} \end{cases} \tag{4.303}$$

where $H_{12} = x_2 - x_1$, $H_{11} = x_3 - x_2$, $H_{11} + H_{12} = x_3 - x_1$ and $C_x \in [0, \frac{H_{12}H_{11}}{4}]$. So we deal with an adaptive interface in x direction with grid length H_{11} and H_{12} , see also Fig. 4.20. Furthermore, we have

$$P_y(y) = \begin{cases} P_{y,1}(y) = \frac{y^2 - 2H_{22}y + H_{21}(H_{22} - y) + H_{22}^2 + C_y}{H_{22}(H_{22} + H_{21})}, & -\frac{3H_{22}}{2} < y < -\frac{H_{22}}{2}, \\ P_{y,2}(y) = \frac{-y^2 + H_{21}(y + H_{22}) - H_{22}y - C_y}{H_{22}H_{21}}, & -\frac{H_{22}}{2} < y < -\frac{H_{21}}{2}, \\ P_{y,3}(y) = \frac{y^2 + 2H_{21}y + H_{22}(H_{21} + y) + H_{21}^2 + C_y}{H_{21}(H_{22} + H_{21})}, & \frac{H_{21}}{2} < y < \frac{3H_{21}}{2}, \\ P_{y,4}(y) = 0, & \text{else} \end{cases} \tag{4.304}$$

where $H_{22} = y_2 - y_5$, $H_{21} = y_4 - y_2$, $H_{21} + H_{22} = y_4 - y_5$ and $C_y \in [0, \frac{H_{21}H_{22}}{4}]$. So we deal with an adaptive interface in y direction with grid length H_{21} and H_{22} , see also Fig. 4.20.

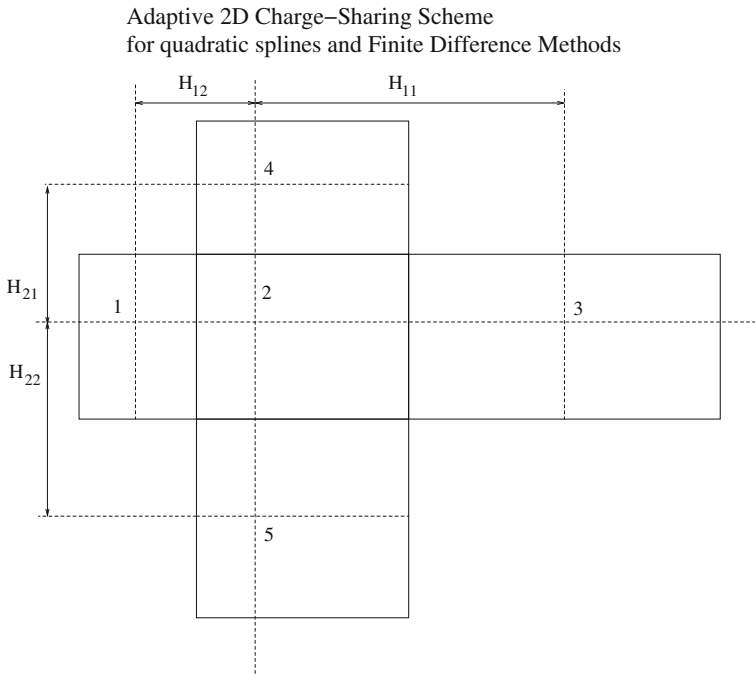


Fig. 4.20 Two-dimensional adaptive five-point charge-sharing scheme assigns charge to the nearest and the next-nearest grid points

Finally, we obtain the 2D shape function with locally one-dimensional shape functions:

$$W(x, y) = \left\{ \begin{array}{l} P_{x,1}(x)P_{y,1}(y), \\ P_{x,2}(x)P_{y,1}(y), \\ P_{x,3}(x)P_{y,1}(y), \\ P_{x,4}(x)P_{y,1}(y), \\ P_{x,1}(x)P_{y,2}(y), \\ P_{x,2}(x)P_{y,2}(y), \\ P_{x,3}(x)P_{y,2}(y), \\ P_{x,4}(x)P_{y,2}(y), \\ P_{x,1}(x)P_{y,3}(y), \\ P_{x,2}(x)P_{y,3}(y), \\ P_{x,3}(x)P_{y,3}(y), \\ P_{x,4}(x)P_{y,3}(y), \\ P_{x,1}(x)P_{y,4}(y), \\ P_{x,2}(x)P_{y,4}(y), \\ P_{x,3}(x)P_{y,4}(y), \\ P_{x,4}(x)P_{y,4}(y). \end{array} \right\} \quad (4.305)$$

For the nonuniform mesh function, we have to fulfil the consistency (mass conservation) (4.204).

Theorem 4.12 *For the nonuniform shape function (4.285), we fulfil the consistency (4.204).*

Proof It is sufficient to prove that the shape functions based on each different domain fulfil the condition.

While we can separate to local one-dimensional problem and each dimension is fulfil, see Sect. 4.4.4.1 and we are done.

4.4.6 Application: Multidimensional Finite Difference Method

In the following, we discuss the multidimensional discretization of the Poisson and electric field equation.

The Poisson equations is given as

$$\Delta\phi(X_{i,j,k}) = -\frac{1}{\epsilon_0}\rho(X_{i,j,k}), \quad X_{i,j,k} \in [0, L]^3 = \Omega, \quad (4.306)$$

$$\phi(X_{i,j,k}) = 0, \quad X_{i,j,k} \in \partial\Omega, \quad (4.307)$$

where $X_{i,j,k} = (x_i, y_j, z_k)^t$ is the three-dimensional coordinate of the particle (i, j, k) .

The electric field is given as

$$E_{i,j,k} = -\nabla\phi(X_{i,j,k}), \quad (4.308)$$

where $E_{i,j,k}$ is the electric field in grid point $X_{i,j,k}$.

The multidimensional finite difference equations are given as

$$\begin{aligned} & \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2} + \frac{\phi_{i,j+1,k} - 2\phi_{i,j,k} + \phi_{i,j-1,k}}{\Delta y^2} \\ & + \frac{\phi_{i,j,k+1} - 2\phi_{i,j,k} + \phi_{i,j,k-1}}{\Delta z^2} = -\frac{1}{\varepsilon_0}\rho_{i,j,k} \in [0, L], \end{aligned} \quad (4.309)$$

$$\phi(0, 0, 0) = 0, \quad \phi(L, 0, 0) = 0, \quad \phi(0, L, 0), \quad \dots, \quad \phi(L, L, L) = 0, \quad (4.310)$$

where $\phi(x_i, y_j, z_k) = \phi_{i,j,k}$

The electric fields are given as

$$E_{i+1/2,j,k} = -\frac{\phi_{i,j,k} - \phi_{i+1,j,k}}{\Delta x}, \quad (4.311)$$

$$E_{i,j+1/2,k} = -\frac{\phi_{i,j,k} - \phi_{i,j+1,k}}{\Delta y}, \quad (4.312)$$

$$E_{i,j,k+1/2} = -\frac{\phi_{i,j,k} - \phi_{i,j,k+1}}{\Delta z}, \quad (4.313)$$

where we called such a discretization “staggered grids”, see [64].

4.4.7 Application: Shape Functions for the Multidimensional Finite Difference Method

In the following subsection, we modify the shape functions to the previous introduced multidimensional finite difference method.

For $n = 1$, we have three constraint equations (additional we need one constraint for the second momentum):

$$\sum_{\mathbf{P}} W_{\mathbf{P}} = 1, \quad (4.314)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} \Delta_i = 0, \quad (4.315)$$

$$\sum_{\mathbf{P}} W_{\mathbf{P}} x_p y_p = xy, \quad (4.316)$$

where $\mathbf{p} = (p_1, p_2)$ is a pair labelling the mesh point \mathbf{p} at position $\mathbf{x}_{\mathbf{p}}$.

We have the following equations:

$$W_1 + W_2 + W_3 + W_4 = 1, \quad (4.317)$$

$$W_1x_1 + W_2x_2 + W_3x_3 + W_4x_4 = x, \quad (4.318)$$

$$W_1y_1 + W_2y_2 + W_3y_3 + W_4y_4 = y, \quad (4.319)$$

$$W_1x_1y_1 + W_2x_2y_2 + W_3x_3y_3 + W_4x_4y_4 = xy. \quad (4.320)$$

By solving Eqs. (4.317)–(4.320), we obtain (using program-code Maxima [68])

$$W_1 = \frac{W_{11}}{W_{12}}, \quad (4.321)$$

$$\begin{aligned} W_{11} = & x_3 (x_2 ((y_3 - y_2) y_4 - y y_3 + y y_2) + x ((y - y_3) y_4 + y_2 y_3 - y y_2)) \\ & + x_4 (x ((y_3 - y_2) y_4 - y y_3 + y y_2) + x_2 ((y - y_3) y_4 + y_2 y_3 - y y_2) \\ & + x_3 ((y_2 - y) y_4 + (y - y_2) y_3)) + x x_2 ((y_2 - y) y_4 + (y - y_2) y_3), \end{aligned} \quad (4.322)$$

$$\begin{aligned} W_{12} = & x_3 (x_2 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_1 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2)) \\ & + x_4 (x_1 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_2 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2) \\ & + x_3 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3)) + x_1 x_2 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3), \end{aligned} \quad (4.323)$$

$$W_2 = -\frac{W_{21}}{W_{22}}, \quad (4.324)$$

$$\begin{aligned} W_{21} = & x_3 (x_1 ((y_3 - y_1) y_4 - y y_3 + y y_1) + x ((y - y_3) y_4 + y_1 y_3 - y y_1)) \\ & + x_4 (x ((y_3 - y_1) y_4 - y y_3 + y y_1) + x_1 ((y - y_3) y_4 + y_1 y_3 - y y_1) \\ & + x_3 ((y_1 - y) y_4 + (y - y_1) y_3)) + x x_1 ((y_1 - y) y_4 + (y - y_1) y_3) \end{aligned} \quad (4.325)$$

$$\begin{aligned} W_{22} = & x_3 (x_2 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_1 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2)) \\ & + x_4 (x_1 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_2 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2) \\ & + x_3 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3)) + x_1 x_2 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3), \end{aligned} \quad (4.326)$$

$$W_3 = \frac{W_{31}}{W_{32}} \quad (4.327)$$

$$\begin{aligned} W_{31} = & x_2 (x_1 ((y_2 - y_1) y_4 - y y_2 + y y_1) + x ((y - y_2) y_4 + y_1 y_2 - y y_1)) \\ & + x_4 (x ((y_2 - y_1) y_4 - y y_2 + y y_1) + x_1 ((y - y_2) y_4 + y_1 y_2 - y y_1) \\ & + x_2 ((y_1 - y) y_4 + (y - y_1) y_2)) + x x_1 ((y_1 - y) y_4 + (y - y_1) y_2) \end{aligned} \quad (4.328)$$

$$\begin{aligned} W_{32} = & x_3 (x_2 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_1 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2)) \\ & + x_4 (x_1 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_2 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2) \\ & + x_3 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3)) + x_1 x_2 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3), \end{aligned} \quad (4.329)$$

$$W_4 = -\frac{W_{41}}{W_{42}}, \quad (4.330)$$

$$\begin{aligned} W_{41} = & x_2 (x_1 ((y_2 - y_1) y_3 - y y_2 + y y_1) + x ((y - y_2) y_3 + y_1 y_2 - y y_1)) \\ & + x_3 (x ((y_2 - y_1) y_3 - y y_2 + y y_1) + x_1 ((y - y_2) y_3 + y_1 y_2 - y y_1) \\ & + x_2 ((y_1 - y) y_3 + (y - y_1) y_2)) + x x_1 ((y_1 - y) y_3 + (y - y_1) y_2) \end{aligned} \quad (4.331)$$

$$\begin{aligned}
W_{42} = & x_3 (x_2 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_1 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2)) \\
& + x_4 (x_1 ((y_3 - y_2) y_4 - y_1 y_3 + y_1 y_2) + x_2 ((y_1 - y_3) y_4 + y_2 y_3 - y_1 y_2)) \\
& + x_3 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3) + x_1 x_2 ((y_2 - y_1) y_4 + (y_1 - y_2) y_3),
\end{aligned} \tag{4.332}$$

$$\text{for } -\frac{\mathbf{H}}{2} \leq \mathbf{x} \leq \frac{\mathbf{H}}{2}. \tag{4.333}$$

where $\mathbf{x} = (x, y)^t$, $\mathbf{H} = (H_x, H_y)^t$.

Using the displacement invariance property (4.217) and Eq.(4.216), we obtain

$$W(x) = \begin{cases} \frac{(3H_x + 3|x|)H_y - 3|y|H_x - 8|x||y|}{3H_x H_y}, & -\frac{\mathbf{H}}{2} < \mathbf{x} < \frac{\mathbf{H}}{2}, \\ \frac{-3H_x|y| + 2|x||y| + (3H_x - 2|x|)H_y}{3H_x H_y}, & -\frac{H_x}{2} < x < \frac{H_x}{2}, \frac{H_y}{2} < |y| < \frac{3H_y}{2}, \\ \frac{6H_x|y| + x(2H_y - 4|y|) - 3H_x H_y}{3H_x H_y}, & \frac{H_x}{2} < x < \frac{3H_x}{2}, -\frac{H_y}{2} < y < \frac{H_y}{2}, \\ \frac{6H_x y + x(2H_y - 4y) - 3H_x H_y}{3H_x H_y}, & \frac{H_x}{2} < x < \frac{3H_x}{2}, \frac{H_y}{2} < y < \frac{3H_y}{2}, \\ \frac{H_y(x + H_x) - 2|y|x - 2|y|H_x}{H_x H_y}, & -\frac{3H_x}{2} < x < -\frac{H_x}{2}, -\frac{H_y}{2} < y < \frac{H_y}{2}, \\ 0, & \text{else,} \end{cases} \tag{4.334}$$

where $y_2 - y_1 = H_y$, $x_3 - x_1 = \frac{3}{2}H_x$, $x_4 - x_1 = -H_x$, $y_3 - y_1 = \frac{1}{2}H_y$, $y_2 - y_1 = H_y$. We deal with an adaptive interface with grid length $\mathbf{H} = (H_x, H_y)^t$ and $\mathbf{H}_{coarse} = 2\mathbf{H} = (2H_x, 2H_y)^t$, given in Fig.4.21.

4.4.8 Simple Test Example: Plume Computation of Ion Thruster with 1D PIC Code

In the following, we present a real-life experiment of an ion thruster with plume computations in 1D, see also the work in [55, 56].

In the following, we present a many particle experiment, which is closer to real numerical applications. The experiment is a simplified thruster model in one space dimension and three velocity dimensions, including the channel and the plume region. Referred to [73], we took the following physics parameters:

Adaptive 2D Charge-Sharing Scheme
for Finite Volume Methods

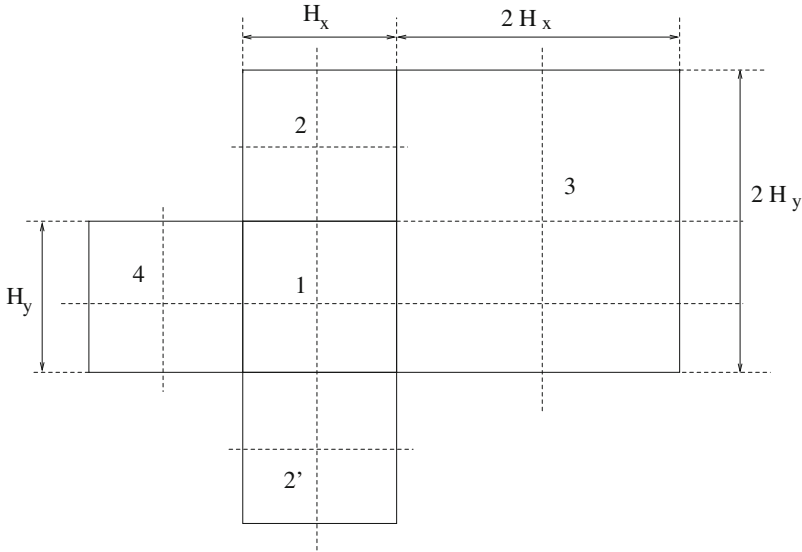


Fig. 4.21 Two-dimensional adaptive five-point charge-sharing scheme assigns charge to the nearest grid point (labelled 1) and the next-nearest grid points in the east–west direction (labelled 2 and 2') and the further north–south direction (labelled 3 and 4)

- Potential at the thruster anode was $\Phi A = 400 \text{ V}$, while the potential at the simulated plume end was taken as zero.
- A static neutral background (here Argon), exponentially decaying in space, was taken for the channel region, with a total density of $n_n = 5.0 \times 10^{18} \text{ m}^{-3}$.
- An electron gun was placed in front of the channel exit ($x \in [300\lambda_{De}; 320\lambda_{De}]$) with an injection flux of $f_e = 2.82 \times 10^{11} \text{ s}^{-1}$. The injected particles had an Gaussian-distributed velocity, due to the thermal velocity $v_{th,e} = 1.03 \times 10^6 \text{ m/s}$. The initial electron temperature was taken as $T_e = 6 \text{ eV}$.
- The implemented reactions are as follows: ionization of Ar with $Ar + e \rightarrow Ar^+ + 2e$ and elastic collisions of electrons and neutrals.

In the 1D model as well as in the real-life thruster, the emitted electrons are getting accelerated by the potential of the anode. These electrons are ionizing the Argon neutrals in the channel, and a plasma is building up, as can be seen in Fig. 4.22. In the real thruster, a configuration of the magnetic field over the whole domain, as well as the resulting in particle–wall interaction, is keeping the plasma in the channel and producing a flat potential, which has a steep decrease at the thruster exit, which accelerates the ions and gives the thrust. While our model is only one dimension, we adapted the magnetic field to the simplified model and took a weak

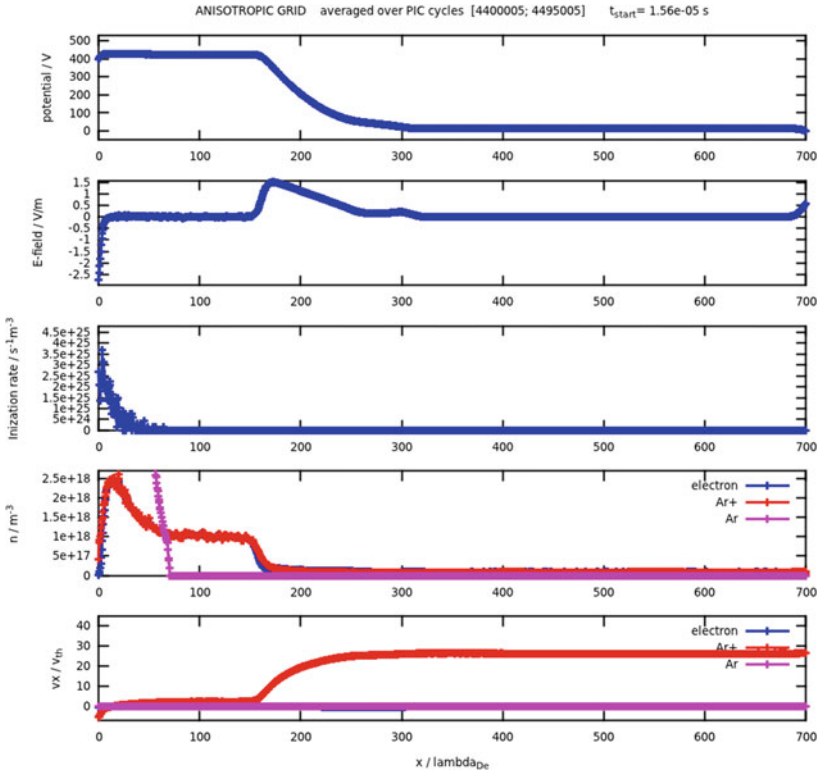


Fig. 4.22 Stable situation in the plume region with potential, electrical field, particle density and particle velocity plotted over the spatial grid length L

magnetic field in the thruster exit region, perpendicular to our space axis x . In this region ($x \in [150\lambda_{De}; 20\lambda_{De}]$), the electron velocity in x direction gets weakened, so that electrons can only pass via collisions. With this configuration, we were able to simulate a simple 1D thruster model, which gets steady state after about $1.5 \times 10^6 \text{ PICsteps} = 5.3 \times 10^6 \text{ s}$, as can be seen in Fig. 4.23.

More computation parameters and the steady-state particle parameters are given in Table 4.2.

Remark 4.20 The test results are produced with uniform and nonuniform grids. In both results, we could achieve the same one-dimensional behaviours. At least, the numerical results validate the behaviour of the steep gradient on the potential, see Fig. 4.22, that decouples the inner and outer part of the ion thruster.

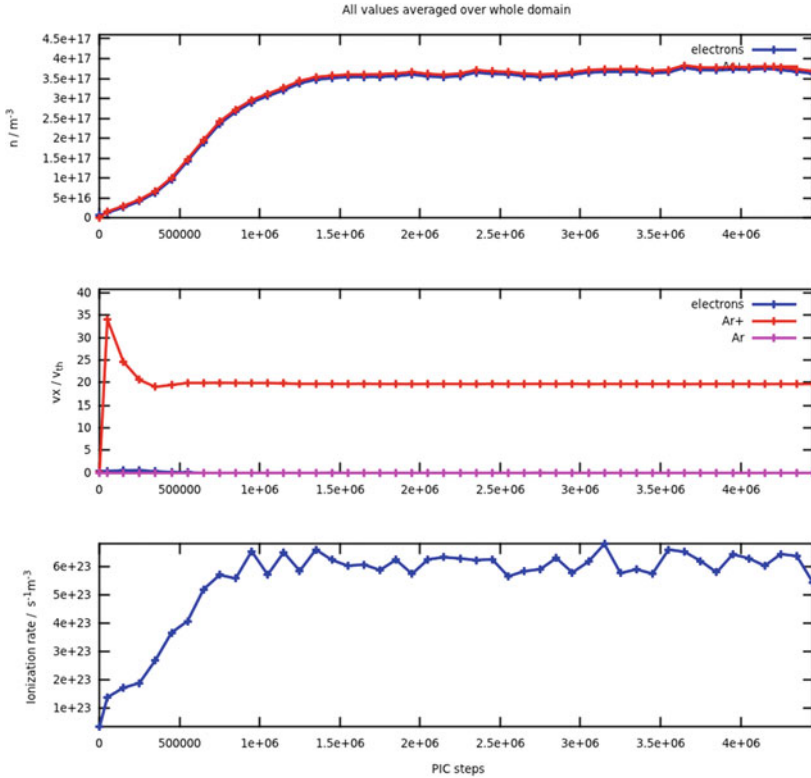


Fig. 4.23 Averaged species in the domain over the computed time

Table 4.2 Parameters for the plume computation

Electrons	$T_e = 6.00 \text{ eV} (69627 \text{ K})$
Superparticles (electrons, PIC)	$N_{db} \times N_{sp} = 100 \times 6.04 \times 10^1 = 6.038 \times 10^3$ $ne = 1.0 \times 10^{12} \text{ cm}^{-3} (1.0 \times 10^{18} \text{ m}^{-3})$ $v_{th,e} = 1.027274 \text{ e} + 06 \text{ m/s}$
Scaling factors	$\omega_{pe} = 5.64146 \times 10^{10} \text{ Hz}$ $\lambda_{De} = 1.820937 \times 10^{-5} \text{ m}$
Ions (Ar+)	$v_{th,Ar} = 8.474025 \times 10^2 \text{ m/s}$
Neutrals (Ar)	$n_n = 5.000000 \text{ e} + 18 \text{ m}^{-3}$
Output of the computations	
Time step	$dt = 3.545181 \times 10^{-12} \text{ s}$
Averaging time	$3.545 \times 10^{-3} \text{ ns} - 0.0 \text{ ns}$
Spatial length	$L_{system} = 1.274656 \times 10^1 \text{ mm}$

4.4.9 Conclusion

We have derived an extension of uniform particle in cell method to nonuniform grids for 1D and 2D equations. The multiscale method, which is given with the parts pusher (microscopic level), solver (macroscopic level) and interpolation/Restriction (complying microscopic and macroscopic level), can be extended with respect to an adaptive scheme. The extensions have been done for the solver, pusher and interpolation functions, which coupled the microscopic and macroscopic model equations. The problem is to modify all parts of the cycle to achieve an extension of the adaptive or nonuniform grids. At least, we can accelerate a simple real-life problem, which has a gap between the high-density (apparatus) and low-density (plume) area, such that adaptive schemes can overcome the uniform step sizes and modify to each disparate spatial and time scales, see [56, 74].

References

1. E. Weinan, *Principle of Multiscale Modelling* (Cambridge University Press, Cambridge, 2010)
2. J. Geiser, in *Coupled Systems: Theory, Models and Applications in Engineering*. Numerical Analysis and Scientific Computing Series, ed. by F. Magoules, C.H. Lai (CRC Press, Chapman & Hall/CRC, 2014)
3. J. Geiser, Operator splitting method for coupled problems: transport and Maxwell equations. *Am. J. Comput. Math. Sci. Res. Publ. USA* **1**, 163–175 (2011)
4. H. Kim, *Multiscale and Multiphysics Computational Frameworks for Nano- and Bio-systems*. Springer Theses (Springer, Heidelberg, 2011)
5. J. Heinlin, G. Morfill, M. Landthaler, W. Stolz, G. Isbary, J.L. Zimmermann, T. Shimizu, S. Karrer, Plasma medicine: possible applications in dermatology. *JDDG, J. Ger. Soc. Dermatol.* (2010). ISSN 1610-0379
6. J. Liebmann, J. Scherer, N. Bibinov, P. Rajasekaran, R. Kovacs, R. Gesche, P. Awakowicz, V. Kolb-Bachofen, Biological effects of nitric oxide generated by an atmospheric pressure gas-plasma on human skin cells. *Nitric Oxide* **24**(1), 8–16 (2011)
7. K.H. Becker, U. Kogelschatz, K.H. Schoenbach, R.J. Barker, *Non-Equilibrium Air Plasmas at Atmospheric Pressure*. Series in Plasma Physics (Taylor and Francis, London, 2004)
8. J. Meichsner, M. Schmidt, R. Schneider, H.-E. Wagner, *Nonthermal Plasma Chemistry and Physics* (CRC Press, Taylor and Francis, Boca Raton, 2012)
9. T.K. Senega, R.P. Brinkmann, Generalized transport coefficients of multicomponent low-temperature plasmas. *IEEE Trans. Plasma Sci.* **35**(5), 1196–1203 (2007)
10. T.K. Senega, *Schwereteilchen-Transport in Niedertemperatur-Plasmen: Modellierung Technisch Relevanter Plasmen* (Logos, Berlin, 2007)
11. W. Dobrygin, *Modelling and Simulation of a Plasmajet*. Diplomarbeit, Theoretische Elektrotechnik (Ruhr-Universität Bochum, 2014)
12. W. Dobrygin, J. Trischmann, T. Hemke, R.P. Brinkmann, *Simulation und Modellierung der Strömungsdynamik eines nicht-thermischen Plasmajets in Atmosphärendruck*. Vortrag, Theoretische Elektrotechnik (Ruhr-Universität Bochum, Bochum, 2014)
13. J. Schäfer, F. Sigeneger, R. Foest, D. Loffhagen, K.-D. Weltmann, On plasma parameters of a self-organized plasma jet at atmospheric pressure. *Eur. Phys. J. D* **60**, 531–538 (2010)
14. T.K. Senega, R.P. Brinkmann, A multi-component transport model for non-equilibrium low-temperature low-pressure plasmas. *J. Phys. D: Appl. Phys.* **39**, 1606–1618 (2006)
15. K.-H. Spatschek, *Theoretische Plasmaphysik* (Teubner Studienbücher, 1990)

16. D. Bothe, On the Maxwell-Stefan approach to multicomponent diffusion. *Parabol. Probl. Progr. Nonlinear Differ. Equ. Appl.* **80**, 81–93 (2011)
17. J.A. Wesselingh, R. Krishna, *Mass Transfer in Multicomponent Mixtures*. VSSD, 1st edn. (Delft, The Netherlands, 2000–2006)
18. K. Böttcher, Numerical solution of a multi-component species transport problem combining diffusion and fluid flow as engineering benchmark. *Int. J. Heat Mass Transf.* **53**, 231–240 (2010)
19. M. Herberg, M. Meyries, J. Prüss, M. Wilke, Reaction-diffusion systems of Maxwell-Stefan type with reversible mass-action kinetics. Preprint, eprint [arXiv:1310.4723](https://arxiv.org/abs/1310.4723) (2013)
20. J. Geiser, in *Iterative Splitting Methods for Differential Equations*. Numerical Analysis and Scientific Computing Series, ed. by F. Magoules, C.H. Lai (Chapman & Hall/CRC, 2011)
21. J. Geiser, in *Modelling and Simulation in Engineering with Multi-physics and Multiscale Methods: Theory and Application*. Numerical Analysis and Scientific Computing Series, ed. by F. Magoules, C.H. Lai (Chapman & Hall/CRC, 2014)
22. D. Fang, M. Hieber, R. Zi, Global existence results for Oldroyd-B fluids in exterior domains: the case of non-small coupling parameters. *Mathematische Annalen* **357**(2), 687–709 (2013)
23. V. Giovangigli, *Multicomponent Flow Modeling* (Birkhäuser, Basel, 1999)
24. R. Krishna, R. Taylor, Multicomponent mass transfer theory and applications, in *Handbook for Heat and Mass Transfer*, vol. 2, Chap. 7, ed. by N. Cheremisinoff (Gulf, Houston, 1986)
25. R. Krishna, J. Wesselingh, The Maxwell-Stefan approach to mass transfer. *Chem. Eng. Sci.* **52**, 861911 (1997)
26. J. Bear, Y. Bachmat, *Introduction to Modeling of Transport Phenomena in Porous Media* (Kluwer Academic Publishers, Dordrecht, 1991)
27. D.J. Acheson, *Elementary Fluid Dynamics*. Oxford Applied Mathematics & Computing Science Series (2002)
28. R. Balescu, *Transport Processes in Plasma: Classical Transport*, vol. 1 (North Holland Publishing, Amsterdam, 1988)
29. B.V. Alexeev, *Generalized Boltzmann Physical Kinetics*, 1st edn. (Elsevier Science, Amsterdam, 2004)
30. M.A. Lieberman, A.J. Lichtenberg, *Principle of Plasma Discharges and Materials Processing*, 2nd edn. (Wiley, New York, 2005)
31. K.N. Kulkarni, Multicomponent diffusion in ternary and quaternary diffusion couples and in multilayered assemblies. Ph.D. thesis, Purdue University, 2008
32. A. Spille-Kohoff, E. Preus, K. Böttcher, Numerical solution of multi-component species transport in gases at any total number of components. *Int. J. Heat Mass Transf.* **55**, 5373–5377 (2012)
33. J. Bear, *Dynamics of Fluids in Porous Media* (American Elsevier, New York, 1972)
34. J. Geiser, *Discretization and Simulation of Systems for Convection-Diffusion-Dispersion Reactions with Applications in Groundwater Contamination*. Monograph, Series: Groundwater Modelling, Management and Contamination (Nova Science Publishers, Inc., New York, 2008)
35. R.E. Ewing, Up-scaling of biological processes and multiphase flow in porous media. *IIMA Volumes in Mathematics and its Applications*, vol. 295 (Springer, New York, 2002), pp. 195–215
36. E. Fein, Software package r^3t : model for transport and retention in porous media. Final report, GRS-192, Braunschweig (2004)
37. M. Genuchten, Convective-dispersive transport of solutes involved in sequential first-order decay reactions. *Comput. Geosci.* **11**(2), 129–147 (1985)
38. S. Larsson, V. Thomee, *Partial Differential Equations with Numerical Methods*. Text in Applied Mathematics, vol. 45 (Springer, Heidelberg, 2003)
39. P. Knabner, L. Angerman, *Numerical Methods for Elliptic and Parabolic Partial Differential Equations: An Applications-oriented Introduction*. Texts in Applied Mathematics (Springer, Heidelberg, 2003)
40. R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics (Cambridge University Press, Cambridge, 2002)

41. S.V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Taylor and Francis, 1980)
42. J. Geiser, Discretization methods with analytical solutions for convection-diffusion dispersion-reaction equations and applications. *J. Eng. Math.* **57**(1), 79–98 (2007)
43. J. Geiser, Iterative operator-splitting methods with higher order time-integration methods and applications for parabolic partial differential equations. *J. Comput. Appl. Math.* Elsevier, Amsterdam, The Netherlands **217**, 227–242 (2008)
44. J. Geiser, Recent advances in splitting methods for multiphysics and multiscale: theory and applications. *J. Algorithms Comput. Technol.* Multi-Sci. Brentwood, Essex, UK, accepted August 2014 (to be published second issue 2015)
45. K. Nanbu, Theory of cumulative small-angle collisions in plasmas. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.* **55**(4), 4642–4652 (1997)
46. B.I. Cohen, L. Divol, A.B. Langdon, E.A. Williams, Effects of ion-ion collisions and inhomogeneity in two-dimensional kinetic ion simulations of stimulated Brillouin backscattering. *Phys. Plasmas* **13**(2), 022705 (2006)
47. B.I. Cohen, A.M. Dimits, A. Friedman, R.E. Caflisch, Time-step considerations in particle simulation algorithms for coulomb collisions in plasmas. *IEEE Trans. Plasma Sci.* **38**(9), 2394–2406 (2010)
48. P. Vabishchevich, *Additive Operator-difference Schemes: Splitting Schemes* (De Gruyter, Berlin, 2014)
49. A. Taflove, S.C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd edn. (Artech House, Norwood, 2005)
50. Y. Yang, R.S. Chen, E.K.N. Yung, The unconditionally stable Crank Nicolson FDTD method for three-dimensional Maxwell's equations. *Microw. Opt. Technol. Lett.* **48**(8), 1619–1622 (2006)
51. J. Shibayama, M. Muraki, J. Yamauchi, H. Nakano, Efficient implicit FDTD algorithm based on locally one-dimensional scheme. *Electron. Lett.* **41**(19) (2005)
52. A. Taflove, *Computational Electrodynamics: The Finite Difference Time Domain Method* (Artech House Inc., Boston, 1995)
53. R.I. McLachlan, G.R.W. Quispel, Splitting methods. *Acta Numerica* 341–434 (2002)
54. R. Hockney, J. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1985)
55. J. Duras, K. Matyash, D. Tskhakaya, O. Kalentev, R. Schneider, Self-force in 1D electrostatic particle-in-cell codes for non-equidistant grids. *Contrib. Plasma Phys.* **54**(8), 697–711 (2014)
56. K. Lueskow, J. Duras, O. Kalentev, K. Matyash, J. Geiser, R. Schneider, D. Tskhakaya. Non-equidistant particle-in-cell for ion thruster plumes, in *Proceedings of the 33rd IEPC*, October, 2013, Washington, DC, USA, IEPC-2013-067 (2013)
57. D. Tskhakaya, K. Matyash, R. Schneider, F. Taccogna, The particle-in-cell method. *Contrib. Plasma Phys.* **47**(8–9), 563–594 (2007)
58. O. Kalentev, K. Matyash, J. Duras, K. Lueskow, R. Schneider, N. Koch, M. Schirra, Electrostatic ion thrusters—towards predictive modeling. *Contrib. Plasma Phys.* **54**(2), 235–248 (2014)
59. F.H. Harlow, The particle-in-cell method for numerical solution of problems in fluid dynamics. *Methods Comput. Phys.* 319–343 (1964)
60. P. Colella, M.R. Dorr, J.A.F. Hittinger, D.F. Martin, High-order, finite-volume methods in mapped coordinates. *J. Comput. Phys.* **230**(8), 2952–2976 (2011)
61. L. Patacchini, I.H. Hutchinson, Explicit time-reversible orbit integration in particle in cell codes with static homogeneous magnetic field. *J. Comput. Phys.* **228**(7), 2604–2615 (2009)
62. G. Lapenta, DEMOCRITUS: an adaptive particle in cell (PIC) code for object-plasma interactions. *J. Comput. Phys.* **230**(12), 4679–4695 (2011)
63. J. Geiser, M. Arab, Modelling, optimization and simulation for a chemical vapor deposition. *J. Porous Media*, Begell House Inc., Redding, USA **12**(9), 847–867 (2009)
64. B. Gustafsson, *High Order Difference Methods for Time dependent PDE*. Springer Series in Computational Mathematics, vol. 38 (Springer, Heidelberg, 2007)
65. G.H. Shortly, R. Weller, Numerical solutions of Laplace's equation. *J. Appl. Phys.* **9**, 334–348 (1938)

66. Chr. Grossmann, H.G. Roos, M. Stynes, *Numerical Treatment of Partial Differential Equations*. Universitext, 1st edn. (Springer, New York, 2007)
67. F. Taccogna, S. Longo, M. Capitelli et al., Particle-in-cell simulation of stationary plasma thruster. *Contrib. Plasma Phys.* **47**(8–9), 635–656 (2007)
68. Maxima version 5.26.0. Maxima: A Computer Algebra System. Online software resource: <http://maxima.sourceforge.net/> (2011)
69. I.P. Omelyana, I.M. Mrygloda, R. Folk, Optimized Forest-Ruth- and Suzuki-like algorithms for integration of motion in many-body systems. *Comput. Phys. Commun.* **146**(2), 188–202 (2002)
70. E. Forest, R.D. Ruth, Fourth-order symplectic integration. *Phys. D: Nonlinear Phenom.* **43**(1), 105–117 (1990)
71. S.A. Chin, J. Geiser, Multi-product operator splitting as a general method of solving autonomous and non-autonomous equations. *IMA J. Numer. Anal.* **31**(4), 1552–1577 (2011)
72. P. Colella, P.C. Norgaard, Controlling self-force errors at refinement boundaries for AMR-PIC. *J. Comput. Phys.* **229**, 947–957 (2010)
73. K. Matyash, O. Kalentev, R. Schneider, Kinetic simulation of the stationary HEMP thruster including the near-field plume region, in *Presented at the 31st International Electric Propulsion Conference (IEPC), IEPC-2009-110* (2009)
74. J. Geiser, F. Riedel, Comparison of integrators for electromagnetic particle in cell methods: algorithms and applications, in *Proceedings*, [arXiv:1411.0816](https://arxiv.org/abs/1411.0816), November 2014