Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Approximation of Coalescence Integrals in Population Balance Models with Local Mass Conservation

(revised version: April 2007)

by

Wolfgang Hackbusch

Preprint no.: 111 2006

Approximation of Coalescence Integrals in Population Balance Models with Local Mass Conservation

Wolfgang Hackbusch
Max-Planck-Institut *Mathematik in den Naturwissenschaften*
Inselstr. 22, D-04103 Leipzig

Abstract

The solution of population balance equations is a function $f(t, r, x)$ describing a population density of particles of the property x at time t and space r. For instance, the additional independent variable x may denote the mass of the particle. The describing equation contains additional sink and source terms involving integral operators. Since the coordinate x adds at least one further dimension to the spatial directions and time coordinate, an efficient numerical treatment of the integral terms is crucial. One of the more involved integral terms appearing in population balance models is the *coalescence integral*, which is of the form $\int_0^x \kappa(x-y, y) f(y) f(x-y) \mathrm{d}y$. The discretisation may use a locally refined grid. In this paper we describe an algorithm which (i) is efficient (the cost is $\mathcal{O}(N \log N)$, N: data size) and (ii) ensures local mass conservation.

AMS Subject Classifications: 44A35, 42A55, 45E99, 45K05, 92D25
Key words: convolution integral, non-uniform grids, discrete convolution, conservation of mass

1 Introduction

Populations of many (small) particles of different size are described by a function $f(t, r, x)$ which indicates the density of particles of size x at time t and space r. In general, x might denote also other properties than size and may also be vector valued describing several properties. As a standard reference to population balance models, we refer to Ramkrishna [7]. While the flow with respect to t, r, x is described by a pde[1] of the form

$$\frac{\partial}{\partial t} f + Df = Q(f), \qquad Df := \mathrm{div}_r A(f) + \mathrm{div}_x B(f) \tag{1.1}$$

(cf. [7, (2.7.9)]), additional sink and source terms appear due to the interaction of particles. Since the coordinate x adds at least one further dimension to the 1-3 spatial directions and time coordinate, an efficient numerical treatment is crucial in computational engineering.

[1] One may also consider the ordinary integro-differential equation $\frac{\partial}{\partial t} f = Q(f)$ with $Df := 0$. In general, $\mathrm{div}_r A(f)$ describes convection and diffusion with respect to the spatial variable r, while $\mathrm{div}_x B(f)$ is reponsible for the growth of the particles. The particular form of the differential operator D is not of interest in this paper.

One of the more involved integral terms appearing in population balance models is the *aggregation integral*, which we consider in this paper. It describes the effect that two particles say of mass y and $x-y$ combine to a new particle of mass x. As can be seen from [7, §3.3.2] or [2, p. 208], the aggregation integrals take the form[2]

$$\begin{aligned} Q &= Q_{\text{source}} - Q_{\text{sink}} \quad \text{with} \\ Q_{\text{source}}(f)(x) &:= \int_0^x \kappa(x-y,y) f(y) f(x-y) \mathrm{d}y, \\ Q_{\text{sink}}(f)(x) &:= 2f(x) \int_0^\infty \kappa(x,y) f(y) \mathrm{d}y. \end{aligned} \tag{1.2}$$

(The space/time variables r, t of f are not written, but note that such integrals appear for all grid points in space and time). The integral term is quadratic with respect to f and is of convolution type (at least, concerning the part $f(y)f(x-y)$). The kernel function $\kappa(\cdot,\cdot)$ describes the aggregation rate and depends on the particular model. In the case of crystallisation or emulsion processes, κ is also called agglomeration or coalescence rate, but the form (1.2) of the integral is the same.

The high computational cost of a naive treatment of the integral terms is one reason, why the full problem (time, up to three spatial coordinates, and the characteristic coordinate x) is not often treated numerically. A typical pessimistic statement can be found in [8]: "(The equations) generally resist solution", and instead the simpler method of moments is proposed. While there are several papers on the analysis of the nature of the solution f, we here concentrate on the efficient algorithmic approach. The computational cost involved by the integral term will be proportional to the number of degrees of freedom (up to a logarithmic factor).

This problem is already treated by the author in [3]. The first basic step is to approximate the kernel function $\kappa(x,y)$ involved in (1.2) by some separable approximation

$$\kappa(x,y) \approx \sum_{\nu=1}^{k} \alpha_\nu(x) \beta_\nu(y) \tag{1.3}$$

(the α_ν, β_ν may be defined piecewise, see [5]). As explained in the cited papers, the approximation error in (1.3) can be expected to decrease exponentially with k. Therefore, it is easy to make this error smaller than the discretisation error introduced by the space/time discretisation.

Replacing κ by the right-hand side, the integral in Q_{source} becomes

$$\sum_{\nu=1}^{k} \int_0^x [\beta_\nu(y) f(y)] [\alpha_\nu(x-y) f(x-y)] \, \mathrm{d}y. \tag{1.4}$$

The brackets underline that the integral is a usual convolution[3] $[\beta_\nu f] * [\alpha_\nu f]$. Hence, the computation of the integral is straightforward and efficient via FFT (fast Fourier transform), provided that the discretisation uses a uniform mesh.

[2]The kernel function κ used here corresponds to $\kappa/2$ in [7, §3.3.2].

[3]The standard convolution integral is $\int_{\mathbb{R}} F(y)G(x-y)\mathrm{d}y$. If both functions are zero for negative arguments, the integral becomes $\int_0^x F(y)G(x-y)\mathrm{d}y$.

The second topic of [3] is the evaluation of these kinds of integrals in the presence of refined grids. In fact, it is typical for population balance models that the grid should become finer when the mass x approaches zero.

The evaluation in [3] is approximate, since it produced (exact) point evaluations which can be used for approximations in the given grid. As mentioned in [3, Sect. 4], the described discretisation does not ensure mass conservation. Only in the case of a uniform grid, a mass conserving modification was given.

There are also other discretisation approaches like in [9]. As described in [9, §4.2], mass conservation cannot be guaranteed exactly, but numerical experiments show a good approximation. Finite volume methods with mass conservation are described in [1] and [6] using a reformulation of the original equation.

While the approach of [3] aims to compute the values of $[\beta_\nu f] * [\alpha_\nu f]$ at the grid points, the algorithm in the paper [4] computes L^2 orthogonal projections of convolutions onto the approximation space. Since these projections are determined exactly, they form a basis for methods with exact mass conservation. We will see in §4.1 that approximation by piecewise polynomials of at least first degree implies mass conservation. However, in the standard case of piecewise constant polynomials one has to modify the algorithm from [4]. This modification is the major part of this article.

The rest of the paper is organised as follows.

In Section 2 we define the mass integrals and recall the global mass conservation of coalescence integrals (before discretisation). In §2.3 we describe the properties which are needed to guarantee that the approximation of the integral term is locally mass conserving.

In Section 3 we introduce the locally refined mesh together with the corresponding spaces of ansatz spaces. These spaces consist of discontinuous piecewise polynomials whose degree may be zero (piecewise constant functions; see §3.3.1) or $p > 0$ (see §3.3.2).

Section 4 contains the result (Theorem 4.1) that for $p > 0$ the approximation defined in [4] already conserves mass locally, except in the last interval where the support of the convolution function is truncated. This result offers a first (non-optimal) possibility to compute a piecewise constant approximation with local mass conservation (see §4.2).

Section 5 is the core of the paper and concerns the case of piecewise constant approximations. It contains the locally mass conserving algorithm for the projected convolution (see §§5.8-5.10). The algorithm follows the lines of the (not mass conserving) algorithm from [4]. However, the basic quantities introduced in §5.1-5.7 are a bit different. As in [4], under certain conditions, the overall cost is $\mathcal{O}(N \log N)$, where N is the data size.

Section 6 discusses the treatment of the last interval which must compensate the loss of mass associated to the truncated part of the support of $f * g$.

Finally, the discretisation of the sink term Q_{sink} from (1.2) is discussed in Section 7.

2 Conservation of Mass

In the following we assume that the "property" coordinate x in equation (1.1) denotes the mass[4] of the particles. Following [9], we consider the mass as an important quantities which should be preserved by the integral term $Q(f)$ and by its discretisation. One may consider a

[4]Equivalently, we may associate the *volume* of the particle, provided that the specific mass is constant.

model where only agglomeration of particles and, maybe, dispersion (splitting of one particle into two of more smaller ones) occur. The simplest model of this kind would be

$$\frac{\partial}{\partial t} f = Q(f).$$

The mathematical formulation of the mass conservation by Q is $\int x\, Q(f)(x)\mathrm{d}x = 0$. In this case the number of particles $\int f(x)\mathrm{d}x$ is changing, but the total mass $\int x f(x)\mathrm{d}x$ of the exact solution is constant.

However, the picture is more involved since also the number of particles may be considered as a quantity to be preserved. If no agglomeration and dispersion take place, the population balance model may describe, e.g., the growth (or decrease) of the particles. In that case the number of particles is constant, while the mass is changing. In order to retain mass conservation, one has to take into account the mass of the surrounding nutrient component.

Here, and in the following, $\int \ldots \mathrm{d}x$ denotes the integration over $\mathbb{R}$. In fact, the integrands will vanish for $x < 0$, since negative masses do not appear. In practice, the support of the integrands should be bounded, since the mass of a particle cannot exceed the total mass $x_{\max}$ of the system and therefore the distribution $f(x)$ should vanish for $x > x_{\max}$. The mathematical model, however, does not necessarily imply this property[5]. Therefore, we make the assumption that the support of f lies in $[0, x_{\max})$ which is at least ensured by truncating the function by zero for $x > x_{\max}$.

2.1 Notation of Mass Functionals

Since f measures the *number density*, the integral $\int_\Omega \int_{x_1}^{x_2} f(x,r,t)\mathrm{d}x\mathrm{d}r$ describes the number of particles in the volume Ω with masses in the interval $[x_1, x_2)$. Instead, the (global) *mass* at a spatial point r and at time t is given by the integral

$$m(f) := \int x f(x,r,t)\mathrm{d}x.$$

To be precise, $m(f)$ depends on r and t and describes the *spatial mass density.* $\int_\Omega m(f)(r,t)\mathrm{d}r$ is the mass in the volume Ω at time t. In the sequel we will suppress the variables r, t, since the integration in (1.2) does not involve r, t.

To express the *local mass* in an interval $[x_1, x_2]$ we introduce the characteristic function

$$\chi_{[x_1,x_2]} := \begin{cases} 1 & \text{for } x \in [x_1, x_2], \\ 0 & \text{otherwise,} \end{cases}$$

and introduce the functional $m_{[x_1,x_2]}$:

$$m_{[x_1,x_2]}(f) := m(\chi_{[x_1,x_2]} f) = \int_{x_1}^{x_2} x f(x)\mathrm{d}x.$$

[5]The true distribution is such that $\int_\Omega \int_{x_1}^{x_2} f(x,r,t)\mathrm{d}x\mathrm{d}r$ is a non-negative integer, the number of particles. In the mathematical model this integral can be non-integer.

2.2 Global Mass Conservation of Coalescence Integrals

The partial differential equation $\frac{\partial}{\partial t}f + Df = g$ does not conserve mass if g is a source term ($m(g) > 0$) or sink term ($m(g) < 0$). No mass is added or subtracted if $m(g) = 0$.

The coalescence integral $Q(f)$ from (1.2) describes that particles of mass $x-y$ and y join with a rate $\kappa(x-y, y)$. The aggregation produces the source term $Q_{\text{source}}(f)$. At the same time this process is absorbing particles and leads to the sink term $-Q_{\text{sink}}(f)$. For completeness we prove that $m(Q_{\text{source}}) = m(Q_{\text{sink}})$, so that $m(Q) = 0$ implies mass conservation of the continuous pde. Below we use that by the nature of the problem, the kernel function is symmetric: $\kappa(x,y) = \kappa(y,x)$. The integral

$$
\begin{aligned}
m(Q_{\text{source}}) &= m\left(\int_0^x \kappa(x-y,y)f(y)f(x-y)\mathrm{d}y\right) \\
&= \int x \int_0^x \kappa(x-y,y)f(y)f(x-y)\mathrm{d}y\mathrm{d}x \\
&= \iint x\kappa(x-y,y)f(y)f(x-y)\mathrm{d}x\mathrm{d}y \underset{z:=x-y}{=} \iint (y+z)\,\kappa(z,y)f(y)f(z)\mathrm{d}y\mathrm{d}z \\
&= \underbrace{\iint y\kappa(z,y)f(y)f(z)\mathrm{d}y\mathrm{d}z}_{\text{interchange } y \text{ and } z, \text{ use symmetry}} + \iint z\kappa(z,y)f(y)f(z)\mathrm{d}y\mathrm{d}z \\
&= 2\iint z\kappa(z,y)f(y)f(z)\mathrm{d}y\mathrm{d}z
\end{aligned}
$$

equals $m(Q_{\text{sink}}) = \int x \cdot 2f(x) \int_0^\infty \kappa(x,y)f(y)\mathrm{d}y\mathrm{d}x \underset{z\leftrightarrow x}{=} 2\iint z\kappa(z,y)f(y)f(z)\mathrm{d}y\mathrm{d}z$. The proof shows that symmetry of κ is needed. This fact leads us to the following remark.

Remark 2.1 *a) The mass conservation of (1.1), (1.2) does not depend on the symmetric kernel function κ. Therefore, the replacement of κ by the separable approximation $\tilde{\kappa}(x,y) := \sum_{\nu=1}^k \alpha_\nu(x)\beta_\nu(y)$ in (1.3) does not violate mass conservation, provided that $\tilde{\kappa}(x,y)$ is symmetric.*

b) If $\tilde{\kappa}(x,y)$ is an unsymmetric approximation, $\tilde{\tilde{\kappa}}(x,y) := [\tilde{\kappa}(x,y) + \tilde{\kappa}(y,x)]/2$ is a possible symmetric separable approximation.

c) The later treatment of the factors $\beta_\nu f$ and $\alpha_\nu f$ in (1.4) becomes much simpler, if the choice of α_ν, β_ν is piecewise constant on the grid explained below in (2.1).

2.3 Local Mass Conservation of Approximations

Let q be any density distribution. We want to approximate q by $\tilde{q}$ where $\tilde{q}$ belongs to a (finitely dimensional) function space. The precise definition of the piecewise polynomial spaces will be given in §3.2 based on a set of grid points

$$0 = x_0 < x_1 < \ldots < x_{N-1} < x_N. \tag{2.1}$$

This gives rise to the associated intervals $[x_i, x_{i+1})$ for $0 \le i < N$.

If the support of q is larger than $[0, x_N]$ (e.g., since $\operatorname{supp}(q) = [0, \infty)$), the approximation $\tilde{q}$ in $[x_{N-1}, x_N)$ is responsible for the part $q|_{[x_{N-1},\infty)}$. In order to avoid extra notations, we extend $\tilde{q}$ in $[x_N, \infty)$ by zero and redefine $x_N := \infty$. Therefore, formerly, the last interval is $[x_{N-1}, \infty)$, although the support of $\tilde{q}$ is finite. The special treatment of the "last interval" will be discussed in Section 6.

$\tilde{q}$ satisfies the property of *local mass conservation* in $[x_i, x_{i+1})$, if and only if

$$m_{[x_i,x_{i+1})}(q) = m_{[x_i,x_{i+1})}(\tilde{q}). \tag{2.2}$$

The approximation is *locally mass conserving*, if (2.2) holds for all intervals $[x_i, x_{i+1})$, $0 \le i < N$. Obviously, local mass conservation implies global mass conservation:

$$\begin{aligned} m(q) &= m\left(\sum_{i=0}^{N-1} \chi_{[x_i,x_{i+1})} q\right) = \sum_{i=0}^{N-1} m\left(\chi_{[x_i,x_{i+1})} q\right) \underset{(2.2)}{=} \sum_{i=0}^{N-1} m\left(\chi_{[x_i,x_{i+1})} \tilde{q}\right) \\ &= m\left(\sum_{i=0}^{N-1} \chi_{[x_i,x_{i+1})} \tilde{q}\right) = m(\tilde{q}). \end{aligned}$$

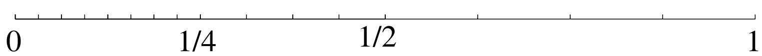
The following observation states the principal difficulty. Let f be a piecewise polynomial function and choose α_ν, β_ν according to Remark 2.1c. Then $f_1 := \alpha_\nu f$ and $f_2 := \beta_\nu f$ belong again to the same function space. However, the convolution $\omega_{\text{exact}} := f_1 * f_2$ (one of the terms in (1.4)) does not belong to this function space. Therefore, a new approximation ω of ω_{exact} is to be computed. In order to fulfil local mass conservation, ω has to satisfy

$$m_{[x_i,x_{i+1})}(\omega_{\text{exact}}) = m_{[x_i,x_{i+1})}(\omega) \qquad (0 \le i \le N). \tag{2.3}$$

3 Approximation Spaces

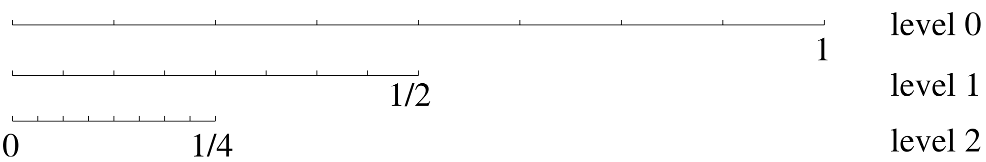
3.1 Locally Refined Meshes

A simple example of a locally refined mesh is depicted below:



(3.1a)

The mesh size 1/8 in $[1/2, 1]$, 1/16 in $[1/4, 1/2]$ and 1/32 in $[0, 1/4]$ is a typical refinement towards $x = 0$. The depicted mesh can be decomposed into different levels as indicated in



(3.1b)

The latter representation uses several levels. Each level ℓ is associated with an *equidistant* grid of size

$$h_\ell := 2^{-\ell} h \qquad (0 \le \ell \le L). \tag{3.2}$$

The largest level number appearing in the grid hierarchy is denoted by L.

The grids depicted in (3.1b) are embedded into infinite grids $\mathcal{M}_\ell$ which are defined below. With h_ℓ from (3.2) we denote the subintervals of level ℓ by[6]

$$I_\nu^\ell := [\nu h_\ell, (\nu+1)\, h_\ell) \qquad \text{for } \nu \in \mathbb{N}_0,\ \ell \in \mathbb{N}_0. \tag{3.3}$$

This defines the meshes

$$\mathcal{M}_\ell := \left\{ I_\nu^\ell : \nu \in \mathbb{N}_0 \right\} \qquad \text{for } \ell \in \mathbb{N}_0. \tag{3.4}$$

The (equidistant) grid points of $\mathcal{M}_\ell$ are $\{\nu h_\ell : \nu \in \mathbb{N}_0\}$.

A finite and *locally refined mesh* $\mathcal{M}$ is a set of finitely many disjoint intervals from various levels, i.e.,[7]

$$\mathcal{M} \subset \bigcup\nolimits_{\ell \in \mathbb{N}_0} \mathcal{M}_\ell, \qquad \text{all } I, I' \in \mathcal{M} \text{ with } I \neq I' \text{ are disjoint}, \qquad \#\mathcal{M} < \infty\,. \tag{3.5}$$

3.2 Functions Spaces

Let $p \in \mathbb{N}_0$ denote the degree of the following polynomials[8]. The piecewise polynomial space $\mathcal{S}$ corresponding to the mesh $\mathcal{M}$ and the degree p is defined by

$$\mathcal{S} = \mathcal{S}\,(\mathcal{M}) = \left\{ \begin{array}{c} \phi \in L^\infty(\mathbb{R}) : \phi|_{I_\nu^\ell} \text{ polynomial of degree } p \text{ if } I_\nu^\ell \in \mathcal{M}, \\ \phi(x) = 0 \text{ if } x \notin I \text{ for all } I \in \mathcal{M} \end{array} \right\}.$$

Note that functions from $\mathcal{S}$ are in general discontinuous at the grid points.

In the present application[9] it makes sense to use the same approximation space $\mathcal{S}$ for both factors of the convolution $[\beta_\nu f] * [\alpha_\nu f]$ in (1.4) and for the approximation of $[\beta_\nu f] * [\alpha_\nu f]$.

3.3 Bases

3.3.1 Piecewise Constant Case

Define

$$\Phi_i^\ell(x) := \left\{ \begin{array}{cl} 1/\sqrt{h_\ell} & \text{for } x \in I_i^\ell, \\ 0 & \text{otherwise}, \end{array} \right. \qquad (\ell \in \mathbb{N}_0,\ i \in \mathbb{N}_0)\,. \tag{3.6}$$

Note that $\operatorname{supp}(\Phi_i^\ell) = I_i^\ell$. The approximation space $\mathcal{S}\,(\mathcal{M})$ has the representation

$$\mathcal{S}\,(\mathcal{M}) = \operatorname{span}\left\{ \Phi_i^\ell : I_i^\ell \in \mathcal{M} \right\}.$$

Obviously, the basis $\left\{ \Phi_i^\ell : I_i^\ell \in \mathcal{M} \right\}$ is orthonormal.

For later use we mention the recursion formula

$$\Phi_i^\ell = \frac{1}{\sqrt{2}} \left(\Phi_{2i}^{\ell+1} + \Phi_{2i+1}^{\ell+1} \right). \tag{3.7}$$

[6] The use of $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$ (instead of $\mathbb{Z}$ as in [4]) in $\nu \in \mathbb{N}_0$ reflects the fact that here the functions of interest have their support in $[0, \infty)$.

[7] The sign # denotes the cardinality of a set.

[8] More generally, we may fix different polynomial degrees $p_\nu^\ell \in \mathbb{N}_0$ for each interval $I = I_\nu^\ell \in \mathcal{M}$.

[9] In [4], we considered the more general case of three different approximation spaces.

3.3.2 Case of degree $p > 0$

In this case we need $p+1$ basis functions per interval. The best choice of orthonormal basis functions $\{\Phi^{\ell}_{(i,\alpha)} : i \in \mathbb{N}_0,\ 0 \le \alpha \le p\}$ are the *Legendre polynomials*; more precisely, the standard Legendre polynomials of degree α defined in the reference interval $[-1,1]$ are to be mapped onto I^{ℓ}_i by an affine mapping and scaled such that $\int |\Phi^{\ell}_{(i,\alpha)}|^2 \mathrm{d}x = 1$.

Remark 3.1 *In the case of $p=1$, the function $\Phi^{\ell}_{(i,0)}$ equals Φ^{ℓ}_i from (3.6), while*

$$\Phi^{\ell}_{(i,1)}(x) = \sqrt{12 h_{\ell}^{-3}}\left(x - x^{\ell}_{i+1/2}\right),$$

where $x^{\ell}_{i+1/2} = (i+1/2)\, h_{\ell}$ is the midpoint of $I^{\ell}_i = [ih_{\ell}, (i+1)\, h_{\ell})$.

3.4 Basic Approach

3.4.1 L^2-Orthogonal Projection onto $\mathcal{S}$

The factors $f_1 := \alpha_{\nu} f$ and $f_2 := \beta_{\nu} f$ yield one of the terms $f_1 * f_2$ in (1.4). Let $f \in \mathcal{S}$. Under the condition of Remark 2.1c, also $f_1, f_2 \in \mathcal{S}$ follows. However, as stated at the end of §2.3, $\omega_{\text{exact}} := f_1 * f_2$ does not belong to $\mathcal{S}$ (see [4, Remark 1.2] for more details). The best approximation $\omega \in \mathcal{S}$ of ω_{exact} in the L^2 sense has the representation

$$\omega = \sum_{i,\ell} \sum_{\alpha=0}^{p} \omega^{\ell}_{(i,\alpha)} \Phi^{\ell}_{(i,\alpha)} \qquad \text{with } \omega^{\ell}_{(i,\alpha)} := \int \Phi^{\ell}_{(i,\alpha)}\, \omega_{\text{exact}}\, \mathrm{d}x \tag{3.8a}$$

(here we exploit the orthonormality of the basis; the sum $\sum_{i,\ell}$ is taken over all pairs (i,ℓ) with $I^{\ell}_i \in \mathcal{M}$).

In the piecewise constant case, (3.8a) becomes

$$\omega = \sum_{i,\ell} \omega^{\ell}_i \Phi^{\ell}_i \qquad \text{with } \omega^{\ell}_i := \int \Phi^{\ell}_i\, \omega_{\text{exact}}\, \mathrm{d}x\,. \tag{3.8b}$$

The exact and efficient computation of (3.8b) or (3.8a) is described in [4] and need not be repeated here.

3.4.2 Mass Conservation

The intervals $[x_i, x_{i+1})$ from §2.3 are the elements $I^{\ell}_i \in \mathcal{M}$. By definition, as already stated in (2.3), local mass conservation holds, if and only if

$$m_{I^{\ell}_i}(\omega_{\text{exact}}) = m_{I^{\ell}_i}(\omega) \qquad \text{for all } I^{\ell}_i \in \mathcal{M}. \tag{3.9}$$

In the following we discuss whether (3.9) holds for the solution ω of (3.8a,b) or – if (3.8b) does not hold – how the definition of the coefficients ω^{ℓ}_i has to be modified and how their fast computation can be organised.

4 Mass Conserving Approximation

This section contains two positive results.

4.1 Polynomial Approximation by At Least Piecewise Linear Functions

Theorem 4.1 *Let $p \geq 1$. The solution $\omega \in \mathcal{S}(\mathcal{M})$ of (3.8a) satisfies the local mass conservation property (3.9) in all intervals $I \in \mathcal{M}$, except the last one, where the correction from Section 6 is necessary.*

Proof. An equivalent formulation of (3.8a) is

$$\omega \in \mathcal{S}(\mathcal{M}) \quad \text{and} \quad \int \Phi^{\ell}_{(i,\alpha)}\, \omega_{\text{exact}}\, \mathrm{d}x = \int \Phi^{\ell}_{(i,\alpha)}\, \omega\, \mathrm{d}x \quad \text{for all } I^{\ell}_i \in \mathcal{M},\ 0 \leq \alpha \leq p. \tag{4.1}$$

The piecewise linear function $x\chi_{I^{\ell}_i}$ is a linear combination of $\Phi^{\ell}_{(i,0)}$ and $\Phi^{\ell}_{(i,1)}$. Therefore (4.1) implies

$$m_{I^{\ell}_i}(\omega_{\text{exact}}) = \int_{I^{\ell}_i} x\, \omega_{\text{exact}}\, \mathrm{d}x = \int_{I^{\ell}_i} x\, \omega\, \mathrm{d}x = m_{I^{\ell}_i}(\omega) \quad \text{for all } I^{\ell}_i \in \mathcal{M}$$

proving (3.9). ■

4.2 First Approach for Piecewise Constant Functions

The foregoing proof requires $p \geq 1$. In fact, it is easy to see that for $p = 0$ (case of (3.8b)) the local (and even the global) masses of $\omega_{\text{exact}} = f_1 * f_2$ and ω may be different.

In the following $\mathcal{S}^1$ denotes the function space $\mathcal{S}(\mathcal{M})$ with $p = 1$, while $\mathcal{S}^0$ is the notation for $\mathcal{S}(\mathcal{M})$ with $p = 0$.

Let us assume that we have an implementation for computing the mapping

$$F^1 : f_1, f_2 \in \mathcal{S}^1 \longmapsto \omega \in \mathcal{S}^1,$$

where $\omega_{\text{exact}} = f_1 * f_2$ and ω satisfy (3.8a) for $p = 1$ (F^1 is described in [4, §8]). We want to use F^1 in order to construct

$$F^0 : f_1, f_2 \in \mathcal{S}^0 \longmapsto \tilde{\omega} \in \mathcal{S}^0$$

with $\omega_{\text{exact}} = f_1 * f_2$ and $\tilde{\omega}$ satisfying the local mass conservation property (3.9).

In fact, since $\mathcal{S}^0 \subset \mathcal{S}^1$, the functions $f_1, f_2 \in \mathcal{S}^0$ may be used as input data of F^1. By Theorem 4.1, $\omega = F^1(f_1, f_2) \in \mathcal{S}^1$ has the same local mass as $\omega_{\text{exact}} = f_1 * f_2$. Now we use the condition (3.9) to construct $\tilde{\omega} \in \mathcal{S}^0$ such that

$$m_{I^{\ell}_i}(\omega) = m_{I^{\ell}_i}(\tilde{\omega}) \quad \text{for all } I^{\ell}_i \in \mathcal{M}.$$

Let $\omega = \sum_{i,\ell}(\omega^{\ell}_{(i,0)}\Phi^{\ell}_{(i,0)} + \omega^{\ell}_{(i,1)}\Phi^{\ell}_{(i,1)})$ be the representation of $\omega \in \mathcal{S}^1$. We split ω into $\omega = \omega^I + \omega^{II}$ with $\omega^I = \sum_{i,\ell}\omega^{\ell}_{(i,0)}\Phi^{\ell}_{(i,0)}$ and $\omega^{II} = \sum_{i,\ell}\omega^{\ell}_{(i,1)}\Phi^{\ell}_{(i,1)}$. Similarly, we make the ansatz $\tilde{\omega} := \tilde{\omega}^I + \tilde{\omega}^{II}$ with $\tilde{\omega}^I := \omega^I \in \mathcal{S}^0$ (note that $\Phi^\ell_{(i,0)}$ and Φ^ℓ_i are only different notations for the same function). Obviously, $m_{I^\ell_i}(\omega^I) = m_{I^\ell_i}(\tilde{\omega}^I)$ holds.

Using the exact representation of $\Phi^\ell_{(i,1)}$ from Remark 3.1, we obtain

$$m_{I^\ell_i}(\Phi^\ell_{(i,1)}) = \int_{ih_\ell}^{(i+1)h_\ell} x\Phi^\ell_{(i,1)}(x)\mathrm{d}x = \sqrt{h_\ell^3/12}, \tag{4.2a}$$

while

$$m_{I^\ell_i}(\Phi^\ell_i) = \int_{ih_\ell}^{(i+1)h_\ell} x\frac{1}{\sqrt{h_\ell}}\mathrm{d}x = \sqrt{h_\ell}\, x^\ell_{i+1/2} \tag{4.2b}$$

with $x^\ell_{i+1/2} = (i+1/2)\,h_\ell$ as in Remark 3.1. Hence,

$$\tilde{\omega}^{II} := \sum_{i,\ell} \omega^\ell_{(i,1)} \frac{m_{I^\ell_i}(\Phi^\ell_{(i,1)})}{m_{I^\ell_i}(\Phi^\ell_i)}\,\Phi^\ell_i = \frac{h_\ell}{\sqrt{12}} \sum_{i,\ell} \frac{\omega^\ell_{(i,1)}}{x^\ell_{i+1/2}}\,\Phi^\ell_i \in \mathcal{S}^0$$

satisfies $m_{I^\ell_i}(\omega^{II}) = m_{I^\ell_i}(\tilde{\omega}^{II})$.

Altogether, we have constructed a uniquely defined $\tilde{\omega} \in \mathcal{S}^0$ with the local mass conservation property (3.9). The coefficients of $\tilde{\omega} = \sum_{i,\ell} \tilde{\omega}^\ell_i \Phi^\ell_i$ are $\tilde{\omega}^\ell_i := \omega^\ell_{(i,0)} + \frac{h_\ell}{\sqrt{12}} \frac{\omega^\ell_{(i,1)}}{x^\ell_{i+1/2}}$.

The necessary computational work is the performance of F^1 and $\mathcal{O}(\dim \mathcal{S}^0)$ operation for the computation of $\tilde{\omega}^\ell_i$.

5 Mass Conservation with Piecewise Constant Functions

Although the previous computation is of linear logarithmic complexity (see [4] for F^1), the algorithm is not optimal, since the input and output data of F^1 have the doubled dimension of those of F^0. This fact should be used to simplify the algorithm. In this section we follow the lines of the algorithm from [4] and insert the necessary modifications in order to obtain local mass conservation.

5.1 Notations

In the following, $\mathcal{S} = \mathcal{S}(\mathcal{M})$ denotes the locally refined space of piecewise constant functions.

So far we used the notations $f_1, f_2 \in \mathcal{S}$ for the factors of the convolution $[\beta_\nu f] * [\alpha_\nu f]$ which is one of the k terms in (1.4). In order to avoid the subindices, we now rename f_1, f_2 by f, g (this is also the notation from [4]). The precise formulation of the problem to be solved is:

$$\text{Given } f, g \in \mathcal{S}, \text{ compute } \omega \in \mathcal{S} \text{ such that } m_I(f * g) = m_I(\omega) \quad \text{for all } I \in \mathcal{M}. \tag{5.1}$$

Besides $\mathcal{M}$ we have the infinite and uniform mesh $\mathcal{M}_\ell$ from (3.4) with step size h_ℓ. Since $\mathcal{M}$ is a union of intervals of various levels, there are non-empty intersections $\mathcal{M} \cap \mathcal{M}_\ell$ for $0 \le \ell \le L$. Define the index sets

$$\mathcal{I}_\ell := \left\{ i \in \mathbb{N}_0 : I^\ell_i \in \mathcal{M} \cap \mathcal{M}_\ell \right\} \qquad \text{for } 0 \le \ell \le L. \tag{5.2}$$

As in (3.8b) the function $f \in \mathcal{S}$ has a representation $f = \sum_{i,\ell} f_i^\ell \Phi_i^\ell$ (summation over all (i,ℓ) with $I_i^\ell \in \mathcal{M}$). We split f into the different levels:

$$f = \sum_{\ell=0}^{L} f_\ell \qquad \text{with } f_\ell := \sum_{i \in \mathcal{I}_\ell} f_i^\ell \Phi_i^\ell \in \mathcal{S}_\ell \tag{5.3}$$

(note that f_ℓ with a subindex ℓ is a function, whereas f_i^ℓ with a superindex ℓ is a coefficient). Here $\mathcal{S}_\ell \subset L^2(\mathbb{R})$ denotes the span of Φ_i^ℓ, $i \in \mathbb{N}_0$. Similarly, we do for $g \in \mathcal{S}$:

$$g = \sum_{\ell=0}^{L} g_\ell \qquad \text{with } g_\ell := \sum_{i \in \mathcal{I}_\ell} g_i^\ell \Phi_i^\ell \in \mathcal{S}_\ell. \tag{5.4}$$

Formally, we set $f_i^\ell = 0$ for all $i \notin \mathcal{I}_\ell$ and consider $f_\ell := \sum_{i \in \mathbb{N}_0} f_i^\ell \Phi_i^\ell$ as a piecewise constant function of the infinite mesh $\mathcal{M}_\ell$. Analogously, we do for g_ℓ.

Next we define the mass conserving projection $\Pi_\ell : V \to \mathcal{S}_\ell$ (here, V is the space $\{\varphi \in L^2(0,\infty) : x\varphi(x) \in L^1(0,\infty)\}$):

$$\Pi_\ell : \varphi \in V \mapsto \varphi_\ell \in \mathcal{S}_\ell \text{ with } \varphi_\ell = \sum_{i \in \mathbb{N}_0} \varphi_i^\ell \Phi_i^\ell,\ \varphi_i^\ell := \frac{m_{I_i^\ell}(\varphi)}{m_{I_i^\ell}(\Phi_i^\ell)} \tag{5.5}$$

(the value of $m_{I_i^\ell}(\Phi_i^\ell)$ is given in (4.2b)).

The convolution $f * g$ can be written as

$$f * g = \sum_{\ell'=0}^{L} \sum_{\ell=0}^{L} f_{\ell'} * g_\ell \,.$$

Since the convolution is symmetric, we can rewrite the sum as

$$f * g = \sum_{\ell' \le \ell} f_{\ell'} * g_\ell + \sum_{\ell < \ell'} g_\ell * f_{\ell'} \,, \tag{5.6}$$

where ℓ', ℓ are restricted to the level intervals $0 \le \ell' \le \ell \le L$. Hence, the basic task is as follows.

Problem 5.1 *Let $\ell' \le \ell$, $f_{\ell'} \in \mathcal{S}_{\ell'}$, $g_\ell \in \mathcal{S}_\ell$, and $\ell'' \in \mathbb{N}_0$ a further level. Then, the projection $\Pi_{\ell''}(f_{\ell'} * g_\ell)$ is to be computed. More precisely, only the restriction of $\Pi_{\ell''}(f_{\ell'} * g_\ell)$ to $\bigcup_{i \in \mathcal{I}_{\ell''}} I_i^{\ell''}$ is needed, since only this part appears in $\mathcal{S} = \mathcal{S}(\mathcal{M})$.*

Because of the splitting (5.6), we may assume $\ell' \le \ell$ without loss of generality. In the case of the second sum one has to interchange the roles of the symbols f and g and to avoid the cases $\ell = \ell'$.

Before we present the solution algorithm in §§5.8-5.10, we introduce some further notations.

5.2 Coarsening $\Pi_\ell : \mathcal{S}_{\ell+1} \to \mathcal{S}_\ell$ and $\hat{\omega}$-Coefficients

Let $\omega_{\ell+1} = \sum_i \omega_i^{\ell+1}\Phi_i^{\ell+1} \in \mathcal{S}_{\ell+1}$ be a piecewise constant function at level $\ell+1$. We want to replace it by $\Pi_\ell\omega_{\ell+1} =: \omega_\ell = \sum_i \omega_i^\ell \Phi_i^\ell$. The interval I_i^ℓ is the union $I_{2i}^{\ell+1} \cup I_{2i+1}^{\ell+1}$ so that $m_{I_i^\ell}(\omega_{\ell+1}) = m_{I_{2i}^{\ell+1}}(\omega_{\ell+1}) + m_{I_{2i+1}^{\ell+1}}(\omega_{\ell+1})$. Using the values (4.2b), we rewrite $m_{I_i^\ell}(\omega_{\ell+1}) = m_{I_i^\ell}(\omega_\ell)$ as

$$\omega_i^\ell \sqrt{h_\ell}\, x_{i+1/2}^\ell = \omega_{2i}^{\ell+1}\sqrt{h_{\ell+1}}\, x_{2i+1/2}^{\ell+1} + \omega_{2i+1}^{\ell+1}\sqrt{h_{\ell+1}}\, x_{2i+3/2}^{\ell+1},$$

leading to the solution

$$\omega_i^\ell = \frac{\omega_{2i}^{\ell+1}\sqrt{h_{\ell+1}}\, x_{2i+1/2}^{\ell+1} + \omega_{2i+1}^{\ell+1}\sqrt{h_{\ell+1}}\, x_{2i+3/2}^{\ell+1}}{\sqrt{h_\ell}\, x_{i+1/2}^\ell} = \frac{1}{\sqrt{2}} \frac{\omega_{2i}^{\ell+1} x_{2i+1/2}^{\ell+1} + \omega_{2i+1}^{\ell+1} x_{2i+3/2}^{\ell+1}}{x_{i+1/2}^\ell}.$$

This formula is obviously not translation invariant. It is advantageous for the computations to introduce the auxiliary coefficients

$$\hat{\omega}_i^\ell := \omega_i^\ell x_{i+1/2}^\ell \qquad (i \in \mathbb{N}_0,\ x_{i+1/2}^\ell = (i+1/2)\, h_\ell). \tag{5.7}$$

Then the relation between $\hat{\omega}_i^\ell$ and $\hat{\omega}_i^{\ell+1}$ is much simpler:

$$\hat{\omega}_i^\ell = \frac{1}{\sqrt{2}} \left(\hat{\omega}_{2i}^{\ell+1} + \hat{\omega}_{2i+1}^{\ell+1} \right). \tag{5.8}$$

5.3 General Coarsening

The previous result gives rise to the following more general question. Let $f \in V$ be some function and consider the two projections

$$\Pi_{\ell+1}(f) = \sum_{i=0}^\infty \varphi_i^{\ell+1}\Phi_i^{\ell+1}, \qquad \Pi_\ell(f) = \sum_{i=0}^\infty \varphi_i^\ell \Phi_i^\ell \tag{5.9}$$

into the fine grid of step size $h_{\ell+1}$ and the coarse one of step size h_ℓ. Is there a simple method to compute the coefficients φ_i^ℓ from $\varphi_i^{\ell+1}$?

For a formalisation we form the following sequences in $\mathbb{R}^{\mathbb{N}_0}$:

$$\varphi_\ell := \left(\varphi_i^\ell\right)_{i=0}^\infty, \qquad \varphi_{\ell+1} := \left(\varphi_i^{\ell+1}\right)_{i=0}^\infty. \tag{5.10}$$

Next, we define two operations acting on sequences. The first one is the diagonal operator Ξ_ℓ with

$$\Xi_\ell : \left(\varphi_i^\ell\right)_{i=0}^\infty \mapsto \left(x_{i+1/2}^\ell\, \varphi_i^\ell\right)_{i=0}^\infty,$$

where $x_{i+1/2}^\ell = (i+1/2)\, h_\ell$. The second operator R is independent of ℓ:

$$R : (a_i)_{i=0}^\infty \mapsto \left(\tfrac{1}{\sqrt{2}}\left(a_{2i} + a_{2i+1}\right)\right)_{i=0}^\infty \qquad \text{for any } a \in \mathbb{R}^{\mathbb{N}_0}.$$

The answer to the question from above is given by the following theorem.

Theorem 5.2 *Let the sequences (5.10) be defined by (5.9). Then*

$$\varphi_\ell = \Xi_\ell^{-1} R\, \Xi_{\ell+1}\, \varphi_{\ell+1} \tag{5.11}$$

holds.

Proof. The coefficient φ_i^ℓ is defined in (5.5) by

$$\varphi_i^\ell = \frac{m_{I_i^\ell}(f)}{m_{I_i^\ell}(\Phi_i^\ell)} = \frac{m_{I_i^\ell}(f)}{\sqrt{h_\ell}\, x_{i+1/2}^\ell} = \frac{1}{\sqrt{h_\ell}\, x_{i+1/2}^\ell} \int_{ih_\ell}^{(i+1)h_\ell} x f(x) dx.$$

Setting $F_i^\ell := \int_{ih_\ell}^{(i+1)h_\ell} x f(x) dx$ and $F_\ell := \left(F_i^\ell\right)_{i=0}^\infty$, we can reformulate the last equation as

$$\varphi_\ell = \frac{1}{\sqrt{h_\ell}} \Xi_\ell^{-1} F_\ell \tag{5.12a}$$

Analogously, we have

$$\varphi_{\ell+1} = \frac{1}{\sqrt{h_{\ell+1}}} \Xi_{\ell+1}^{-1} F_{\ell+1}. \tag{5.12b}$$

Since

$$\begin{aligned} F_i^\ell &= \int_{ih_\ell}^{(i+1)h_\ell} x f(x) dx = \int_{2ih_{\ell+1}}^{2(i+1)h_{\ell+1}} x f(x) dx \\ &= \int_{2ih_{\ell+1}}^{(2i+1)h_{\ell+1}} x f(x) dx + \int_{(2i+1)h_{\ell+1}}^{(2i+2)h_{\ell+1}} x f(x) dx = F_{2i}^{\ell+1} + F_{2i+1}^{\ell+1}, \end{aligned}$$

the sequences F_ℓ and $F_{\ell+1}$ are related by

$$F_\ell = \sqrt{2} R F_{\ell+1}. \tag{5.12c}$$

Combining (5.12a-c), we get $\varphi_\ell = \frac{1}{\sqrt{h_\ell}} \Xi_\ell^{-1} \sqrt{2} R \sqrt{h_{\ell+1}} \Xi_{\ell+1} \varphi_{\ell+1}$. Since $\sqrt{2}\sqrt{h_{\ell+1}}/\sqrt{h_\ell} = 1$, the assertion is proved. ■

5.4 μ-Coefficients

The mapping $f_{\ell'} \in \mathcal{S}_{\ell'}$, $g_\ell \in \mathcal{S}_\ell \mapsto \omega_{\ell''} := \Pi_{\ell''}(f_{\ell'} * g_\ell)$ is linear in both arguments $f_{\ell'}$ and g_ℓ. Hence, the function $\Pi_{\ell''}\left(\Phi_j^{\ell'} * \Phi_k^\ell\right)$ is of interest. Its ith component is $\int \Phi_i^{\ell''}(x)\; \Pi_{\ell''}\left(\Phi_j^{\ell'} * \Phi_k^\ell\right)(x)\mathrm{d}x$ and denoted by

$$\mu_{i,j,k}^{\ell'',\ell',\ell} := \int \Phi_i^{\ell''}(x)\; \Pi_{\ell''}\left(\Phi_j^{\ell'} * \Phi_k^\ell\right)(x)\mathrm{d}x. \tag{5.13}$$

Formally the projected convolution $\Pi_{\ell''}(f_{\ell'} * g_\ell)$ of $f_{\ell'} = \sum_{j\in\mathbb{N}_0} f_j^{\ell'} \Phi_j^{\ell'} \in \mathcal{S}_{\ell'}$ and $g_{\ell'} = \sum_{k\in\mathbb{N}_0} g_k^\ell \Phi_k^\ell \in \mathcal{S}_\ell$ can be represented by

$$\Pi_{\ell''}(f_{\ell'} * g_\ell) = \sum_{i\in\mathbb{N}_0} \omega_i^{\ell''} \Phi_i^{\ell''} \qquad \text{with} \quad \omega_i^{\ell''} = \sum_{j,k\in\mathbb{N}_0} f_j^{\ell'} g_k^\ell \mu_{i,j,k}^{\ell'',\ell',\ell}. \tag{5.14}$$

As we will see, it suffices to know the concrete values of $\mu_{i,j,k}^{\ell'',\ell',\ell}$ for $\ell''=\ell'=\ell$. The product $\Phi_j^\ell * \Phi_k^\ell$ is the piecewise linear hat function with support in $[(j+k)\,h_\ell,(j+k+2)\,h_\ell]$ and value 1 at $(j+k+1)\,h_\ell$. Hence, $\mu_{i,j,k}^{\ell,\ell,\ell}$ is non-zero only for $i=j+k$ and $i=j+k+1$:

$$\mu_{i,j,k}^{\ell,\ell,\ell} := \begin{cases} \frac{1}{2}\sqrt{h_\ell}+\frac{1}{12}h_\ell^{3/2}/x_{i+1/2}^\ell & \text{for } i=j+k, \\ \frac{1}{2}\sqrt{h_\ell}-\frac{1}{12}h_\ell^{3/2}/x_{i+1/2}^\ell & \text{for } i=j+k+1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

Proof. We consider only the case $i=j+k$. The mass $m_{I_i^\ell}(\Phi_j^\ell * \Phi_k^\ell)$ equals $\int_{ih_\ell}^{(i+1)h_\ell}\frac{x-ih_\ell}{h_\ell}x\mathrm{d}x = \frac{1}{2}(i+1/2)\,h_\ell^2+\frac{1}{12}h_\ell^2 = \frac{h_\ell}{2}x_{i+1/2}^\ell+\frac{1}{12}h_\ell^2$. Division by $m_{I_i^\ell}(\Phi_i^\ell)=\sqrt{h_\ell}\,x_{i+1/2}^\ell$ (cf. (4.2b)) yields the ith coefficient of $\Pi_\ell(\Phi_j^\ell * \Phi_k^\ell)=\mu_{i,j,k}^{\ell,\ell,\ell}$ (cf. (5.5)). ■

For later use we mention two recursion properties of the μ-coefficients.

Lemma 5.3 *The following recursions hold:*

$$\begin{aligned} \mu_{i,j,k}^{\ell'',\ell',\ell} &= \frac{1}{\sqrt{2}}\left(\mu_{i,2j,k}^{\ell'',\ell'+1,\ell}+\mu_{i,2j+1,k}^{\ell'',\ell'+1,\ell}\right), \\ \mu_{i,j,k}^{\ell'',\ell',\ell} &= \frac{1}{\sqrt{2}}\left(\mu_{i,j,2k}^{\ell'',\ell',\ell+1}+\mu_{i,j,2k+1}^{\ell'',\ell',\ell+1}\right). \end{aligned} \tag{5.16}$$

Proof. Since $\Pi_{\ell''}$ is linear, (3.7) implies

$$\begin{aligned} \mu_{i,j,k}^{\ell'',\ell'-1,\ell} &= \int \Phi_i^{\ell''}(x)\;\Pi_{\ell''}\left(\Phi_j^{\ell'-1} * \Phi_k^\ell\right)(x)\mathrm{d}x \\ &= \int \Phi_i^{\ell''}(x)\;\Pi_{\ell''}\left(\frac{1}{\sqrt{2}}\left[\Phi_{2j}^{\ell'}+\Phi_{2j+1}^{\ell'}\right] * \Phi_k^\ell\right)(x)\mathrm{d}x = \frac{1}{\sqrt{2}}\left(\mu_{i,2j,k}^{\ell'',\ell',\ell}+\mu_{i,2j+1,k}^{\ell'',\ell',\ell}\right). \end{aligned}$$

The proof of the second identity is analogous. ■

5.5 Type A and Type B Coefficients

As in [4], the plan is to perform the sum $\sum_{k\in\mathbb{N}_0} g_k^\ell \mu_{i,j,k}^{\ell'',\ell',\ell}$ by some cheap recursions and to perform the j-summation with the help of FFT. Here, FFT is used to compute a discrete convolution $\sum_j a_j b_{i-j}$ of two sequences $a,b\in\mathbb{R}^{\mathbb{N}_0}$. To reach the form of the discrete convolution we need a certain shift invariance property of the coefficients $\mu_{i,j,k}^{\ell'',\ell',\ell}$. In [4], the corresponding coefficient named $\gamma_{i,j,k}^{\ell'',\ell',\ell}$ has the property $\gamma_{i,j,k}^{\ell'',\ell',\ell}=\gamma_{0,0,k-i2^{\ell-\ell''}+j2^{\ell-\ell'}}^{\ell'',\ell',\ell}$, provided that $\ell\geq\max\{\ell',\ell''\}$. However, different from [4], the quantities $\mu_{i,j,k}^{\ell'',\ell',\ell}$ appearing here are not translation invariant because of $m(\Phi_i^\ell)\neq m(\Phi_{i'}^\ell)$ for $i\neq i'$. This makes the construction of the algorithm more involved. Coefficients like $\mu_{i,j,k}^{\ell'',\ell',\ell}$ are to be split into a "Type A" and a "Type B" part. These types are defined next.

5.5.1 Type A Coefficients

This is the case of perfect translation invariance.

If a family of coefficients $a_{i,j,k}^{\ell'',\ell',\ell}$ is dependent on three pairs (i,ℓ''), (j,ℓ'), (k,ℓ), we say that $a_{i,j,k}^{\ell'',\ell',\ell}$ is of Type A if

$$a_{i,j,k}^{\ell'',\ell',\ell} = a_{0,0,k-i2^{\ell-\ell''}+j2^{\ell-\ell'}}^{\ell'',\ell',\ell} \qquad \text{for } \ell \geq \max\{\ell',\ell''\} \text{ and } i,j,k \in \mathbb{N}_0. \tag{5.17}$$

After summation over k, we get coefficients $a_{i,j}^{\ell'',\ell'}$ depending only on (i,ℓ''), (j,ℓ'). Then Type A is defined by

$$\begin{aligned} a_{i,j}^{\ell'',\ell'} &= a_{0,j-i2^{\ell'-\ell''}}^{\ell'',\ell'} && \text{for } \ell'' \leq \ell' \text{ and } i,j,k \in \mathbb{N}_0, \\ a_{i,j}^{\ell'',\ell'} &= a_{i-j2^{\ell''-\ell'},0}^{\ell'',\ell'} && \text{for } \ell'' \geq \ell' \text{ and } i,j,k \in \mathbb{N}_0. \end{aligned} \tag{5.18}$$

5.5.2 Type B Coefficients

The three indices i,j,k of $a_{i,j,k}^{\ell'',\ell',\ell}$ can be interpreted as follows: $i \in \mathbb{N}_0$ is the principal index, while j and k are fixed parameters. Then $a_{*,j,k}^{\ell'',\ell',\ell} := \left(a_{i,j,k}^{\ell'',\ell',\ell}\right)_{i=0}^{\infty} \in \mathbb{R}^{\mathbb{N}_0}$ is a sequence with additional dependence on j,k. Applying the operator $\Xi_{\ell''}$ from above, we form the sequence $\hat{a}_{*,j,k}^{\ell'',\ell',\ell} := \Xi_{\ell''} a_{*,j,k}^{\ell'',\ell',\ell}$, i.e.,

$$\hat{a}_{i,j,k}^{\ell'',\ell',\ell} = x_{i+1/2}^{\ell''} a_{i,j,k}^{\ell'',\ell',\ell}.$$

Then $a_{i,j,k}^{\ell'',\ell',\ell}$ is said to be of Type B, if $\hat{a}_{i,j,k}^{\ell'',\ell',\ell}$ is of Type A.

In the case of $a_{i,j}^{\ell'',\ell'}$ we define analogously

$$\hat{a}_{i,j}^{\ell'',\ell'} = x_{i+1/2}^{\ell''} a_{i,j}^{\ell'',\ell'}.$$

Again $a_{i,j}^{\ell'',\ell'}$ is said to be of Type B, if $\hat{a}_{i,j}^{\ell'',\ell'}$ is of Type A.

An example of a splitting into the types A,B will be given in §5.5.4. The translation invariance with respect to the indices j,k is discussed next.

5.5.3 Translation Invariance in j,k

It is trivial to see that $\Phi_j^{\ell'} * \Phi_k^{\ell}$ and $\Phi_{j'}^{\ell'} * \Phi_{k'}^{\ell}$ with $jh_{\ell'} + kh_\ell = j'h_{\ell'} + k'h_\ell$ are equal. Hence, also $\mu_{i,j,k}^{\ell'',\ell',\ell} = \mu_{i,j',k'}^{\ell'',\ell',\ell}$ holds.

Let $\ell \geq \ell'$. Then $jh_{\ell'} + kh_\ell = j'h_{\ell'} + k'h_\ell$ becomes $j2^{\ell-\ell'} + k = j'2^{\ell-\ell'} + k'$. The choice $j' = 0$ leads to

$$\mu_{i,j,k}^{\ell'',\ell',\ell} = \mu_{i,0,k+j2^{\ell-\ell'}}^{\ell'',\ell',\ell}.$$

5.5.4 Inspection of $\mu^{\ell,\ell,\ell}_{i,j,k}$

We split $\mu^{\ell,\ell,\ell}_{i,j,k}$ from (5.15) into

$$\begin{aligned}
\mu^{\ell,\ell,\ell}_{i,j,k} &= {}^{A}\mu^{\ell,\ell,\ell}_{i,j,k} + {}^{B}\mu^{\ell,\ell,\ell}_{i,j,k} \qquad \text{with} \qquad (5.19)\\
{}^{A}\mu^{\ell,\ell,\ell}_{i,j,k} &:= \begin{cases} \frac{1}{2}\sqrt{h_\ell} & \text{for } j+k \le i \le j+k+1, \\ 0 & \text{otherwise,} \end{cases}\\
{}^{B}\hat{\mu}^{\ell,\ell,\ell}_{i,j,k} &:= \left\{ \begin{array}{cl} +\frac{1}{12}h_\ell^{3/2} & \text{for } i=j+k, \\ -\frac{1}{12}h_\ell^{3/2} & \text{for } i=j+k+1, \\ 0 & \text{otherwise.,} \end{array} \right\}, \qquad {}^{B}\mu^{\ell,\ell,\ell}_{i,j,k} := {}^{B}\hat{\mu}^{\ell,\ell,\ell}_{i,j,k} \,/ x^{\ell}_{i+1/2}
\end{aligned}$$

Obviously, the coefficients ${}^{A}\mu^{\ell,\ell,\ell}_{i,j,k}$ are of Type A, while ${}^{B}\mu^{\ell,\ell,\ell}_{i,j,k}$ is of Type B.

5.6 Condensation of Type A and Type B Coefficients

Let $a^{\ell'',\ell',\ell}_{i,j,k}$ be of Type A. Condensation from level ℓ'' to $\ell''-1$ is described in (5.11) by

$$a^{\ell''-1,\ell',\ell}_{*,j,k} = \Xi^{-1}_{\ell''-1} R\, \Xi_{\ell''}\, a^{\ell'',\ell',\ell}_{*,j,k},$$

where $a^{\ell'',\ell',\ell}_{*,j,k} := \left(a^{\ell'',\ell',\ell}_{i,j,k}\right)_{i=0}^{\infty}$ represents the sequence in the first index and where the operators $\Xi^{-1}_{\ell''-1}$, R, $\Xi_{\ell''}$ act on this index. The componentwise formulation of this relation is

$$a^{\ell''-1,\ell',\ell}_{i,j,k} = \frac{1}{\sqrt{2}} \frac{a^{\ell'',\ell',\ell}_{2i,j,k} x^{\ell''}_{2i+1/2} + a^{\ell'',\ell',\ell}_{2i+1,j,k} x^{\ell''}_{2i+3/2}}{x^{\ell''-1}_{i+1/2}}. \tag{5.20}$$

Using the identities $\frac{x^{\ell''}_{2i+1/2}}{x^{\ell''-1}_{i+1/2}} = 1 - \frac{h_{\ell''}/4}{x^{\ell''-1}_{i+1/2}}$ and $\frac{x^{\ell''}_{2i+1/2}}{x^{\ell''-1}_{i+1/2}} = 1 + \frac{h_{\ell''}/4}{x^{\ell''-1}_{i+1/2}}$, we get

$$a^{\ell''-1,\ell',\ell}_{i,j,k} = \frac{1}{\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{2i,j,k} + a^{\ell'',\ell',\ell}_{2i+1,j,k}\right) + \frac{h_{\ell''}}{4\sqrt{2}} \frac{a^{\ell'',\ell',\ell}_{2i+1,j,k} - a^{\ell'',\ell',\ell}_{2i,j,k}}{x^{\ell''-1}_{i+1/2}}. \tag{5.21}$$

This suggests the splitting

$$\begin{aligned}
a^{\ell''-1,\ell',\ell}_{i,j,k} &= {}^{A}a^{\ell''-1,\ell',\ell}_{i,j,k} + {}^{B}a^{\ell''-1,\ell',\ell}_{i,j,k} \qquad \text{with} \qquad (5.22)\\
{}^{A}a^{\ell''-1,\ell',\ell}_{i,j,k} &:= \frac{1}{\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{2i,j,k} + a^{\ell'',\ell',\ell}_{2i+1,j,k}\right),\\
{}^{B}\hat{a}^{\ell''-1,\ell',\ell}_{i,j,k} &:= \frac{h_{\ell''}}{4\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{2i+1,j,k} - a^{\ell'',\ell',\ell}_{2i,j,k}\right), \qquad {}^{B}a^{\ell''-1,\ell',\ell}_{i,j,k} := {}^{B}\hat{a}^{\ell''-1,\ell',\ell}_{i,j,k} \,/ x^{\ell''-1}_{i+1/2}.
\end{aligned}$$

Lemma 5.4 *Let $a^{\ell'',\ell',\ell}_{i,j,k}$ be of Type A. Then ${}^{A}a^{\ell''-1,\ell',\ell}_{i,j,k}$ is of Type A, while ${}^{B}a^{\ell''-1,\ell',\ell}_{i,j,k}$ is of Type B.*

Proof. Let $\ell \geq \max\{\ell', \ell''\}$. In $a^{\ell'',\ell',\ell}_{2i,j,k} + a^{\ell'',\ell',\ell}_{2i+1,j,k}$ we use the Type A property of $a^{\ell'',\ell',\ell}_{i,j,k}$:

$$ {}^{A}a^{\ell''-1,\ell',\ell}_{i,j,k} = \frac{1}{\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{2i,j,k} + a^{\ell'',\ell',\ell}_{2i+1,j,k} \right) = \frac{1}{\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{0,0,k-(2i)2^{\ell-\ell''}+j2^{\ell-\ell'}} + a^{\ell'',\ell',\ell}_{0,0,k-(2i+1)2^{\ell-\ell''}+j2^{\ell-\ell'}} \right) $$

(cf. (5.17)). Similarly,

$$ \begin{aligned} {}^{A}a^{\ell''-1,\ell',\ell}_{0,0,k-i2^{\ell-(\ell''-1)}+j2^{\ell-\ell'}} &= \frac{1}{\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{0,0,k-i2^{\ell-(\ell''-1)}+j2^{\ell-\ell'}} + a^{\ell'',\ell',\ell}_{1,0,k-i2^{\ell-(\ell''-1)}+j2^{\ell-\ell'}} \right) \\ &= \frac{1}{\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{0,0,k-(2i)2^{\ell-\ell''}+j2^{\ell-\ell'}} + a^{\ell'',\ell',\ell}_{1,0,k-(2i+1)2^{\ell-\ell''}+j2^{\ell-\ell'}} \right) \end{aligned} $$

follows and proves ${}^{A}a^{\ell''-1,\ell',\ell}_{i,j,k} = {}^{A}a^{\ell''-1,\ell',\ell}_{0,0,k-i2^{\ell-(\ell''-1)}+j2^{\ell-\ell'}}$, which is (5.17) for ${}^{A}a^{\ell''-1,\ell',\ell}_{i,j,k}$.

Next, we have to show that ${}^{B}\hat{a}^{\ell''-1,\ell',\ell}_{i,j,k}$ is of Type A. In fact,

$$ {}^{B}\hat{a}^{\ell''-1,\ell',\ell}_{i,j,k} = \frac{h_{\ell''}}{4\sqrt{2}} \left(a^{\ell'',\ell',\ell}_{0,0,k-(2i+1)2^{\ell-\ell''}+j2^{\ell-\ell'}} - a^{\ell'',\ell',\ell}_{0,0,k-(2i)2^{\ell-\ell''}+j2^{\ell-\ell'}} \right) = {}^{B}\hat{a}^{\ell''-1,\ell',\ell}_{0,0,k-i2^{\ell-(\ell''-1)}+j2^{\ell-\ell'}} $$

follows as above. ■

In the second part we assume that $a^{\ell'',\ell',\ell}_{i,j,k}$ be of Type B. Note that the "hat coefficients" $\hat{a}^{\ell'',\ell',\ell}_{i,j,k} := a^{\ell'',\ell',\ell}_{i,j,k} x^{\ell''}_{i+1/2}$ simplify the representation (5.20):

$$ \hat{a}^{\ell''-1,\ell',\ell}_{i,j,k} = \frac{1}{\sqrt{2}} \left(\hat{a}^{\ell'',\ell',\ell}_{2i,j,k} + \hat{a}^{\ell'',\ell',\ell}_{2i+1,j,k} \right). \tag{5.23} $$

Lemma 5.5 *Let $a^{\ell'',\ell',\ell}_{i,j,k}$ be of Type B. Then $a^{\ell''-1,\ell',\ell}_{i,j,k}$ is also of Type B.*

Proof. Again the representation (5.20) is valid, which should be written as (5.23). Since $a^{\ell'',\ell',\ell}_{i,j,k}$ is of Type B, $\hat{a}^{\ell'',\ell',\ell}_{i,j,k}$ is of Type A. Hence,

$$ \begin{aligned} \hat{a}^{\ell''-1,\ell',\ell}_{i,j,k} &= \frac{1}{\sqrt{2}} \left(\hat{a}^{\ell'',\ell',\ell}_{2i,j,k} + \hat{a}^{\ell'',\ell',\ell}_{2i+1,j,k} \right) = \frac{1}{\sqrt{2}} \left(\hat{a}^{\ell'',\ell',\ell}_{0,0,k-(2i)2^{\ell-\ell''}+j2^{\ell-\ell'}} + \hat{a}^{\ell'',\ell',\ell}_{0,0,k-(2i+1)2^{\ell-\ell''}+j2^{\ell-\ell'}} \right) \\ &= \hat{a}^{\ell''-1,\ell',\ell}_{0,0,k-i2^{\ell-(\ell''-1)}+j2^{\ell-\ell'}} \end{aligned} $$

follows as the previous parts, proving the Type A property of $\hat{a}^{\ell''-1,\ell',\ell}_{i,j,k}$ and therefore the Type B property of $a^{\ell''-1,\ell',\ell}_{i,j,k}$. ■

The case of coefficients $a^{\ell'',\ell'}_{i,j}$ need not be discussed since Lemmata 5.4 and 5.5 can easily be reformulated for this case.

5.7 G-Coefficients

Equation (5.14) contains the summation

$$ G^{\ell'',\ell'}_{i,j} := \sum_{k\in\mathbb{N}_0} g^{\ell}_{k} \mu^{\ell'',\ell',\ell}_{i,j,k}, \tag{5.24} $$

which defines the G-coefficients. As mentioned in Problem 5.1, $\ell' \leq \ell$ holds without loss of generality. Furthermore, we assume $\ell'' \leq \ell$.

Lemma 5.6 *For $\ell'' = \ell' = \ell$, the G-coefficients are given by*

$$G_{i,j}^{\ell,\ell} = g_{i-j}^{\ell}\mu_{i,j,i-j}^{\ell,\ell,\ell} + g_{i-j-1}^{\ell}\mu_{i,j,i-j-1}^{\ell,\ell,\ell} = {}^{A}G_{i,j}^{\ell,\ell} + {}^{B}G_{i,j}^{\ell,\ell} \qquad \text{with} \tag{5.25a}$$

$${}^{A}G_{i,j}^{\ell,\ell} := \frac{\sqrt{h_\ell}}{2}\left(g_{i-j}^{\ell} + g_{i-j-1}^{\ell}\right), \tag{5.25b}$$

$${}^{B}\hat{G}_{i,j}^{\ell,\ell} := \frac{h_\ell^{3/2}}{12}\left(g_{i-j}^{\ell} - g_{i-j-1}^{\ell}\right) \quad \text{and} \quad {}^{B}G_{i,j}^{\ell,\ell} := {}^{B}\hat{G}_{i,j}^{\ell,\ell}\,/x_{i+1/2}^{\ell}. \tag{5.25c}$$

The part ${}^{A}G_{i,j}^{\ell,\ell}$ is of Type A, while ${}^{B}G_{i,j}^{\ell,\ell}$ is of Type B.

Proof. In the sum (5.24) only two μ-coefficients are non-zero (see (5.15)). The type properties are obvious. ■

Next, we want to obtain a recursion formula which computes $G_{i,j}^{\ell-1,\ell-1}$ from $G_{i,j}^{\ell,\ell}$. First, we prove

$$G_{i,j}^{\ell,\ell'-1} = \frac{1}{\sqrt{2}}\left(G_{i,2j}^{\ell,\ell'} + G_{i,2j+1}^{\ell,\ell'}\right). \tag{5.26}$$

Proof. $G_{i,j}^{\ell,\ell'-1} = \sum_{k\in\mathbb{N}_0} g_k^{\ell}\mu_{i,j,k}^{\ell,\ell'-1,\ell} \underset{(5.16)}{=} \sum_{k\in\mathbb{N}_0} g_k^{\ell}\frac{1}{\sqrt{2}}\left(\mu_{i,2j,k}^{\ell,\ell',\ell} + \mu_{i,2j+1,k}^{\ell,\ell',\ell}\right) = \frac{1}{\sqrt{2}}\left(G_{i,2j}^{\ell,\ell'} + G_{i,2j+1}^{\ell,\ell'}\right)$. ■

Using the splitting $G_{i,j}^{\ell,\ell} = {}^{A}G_{i,j}^{\ell,\ell} + {}^{B}G_{i,j}^{\ell,\ell}$, we perform equation (5.26) separately for both parts:

$${}^{A}G_{i,j}^{\ell,\ell-1} = \frac{1}{\sqrt{2}}\left({}^{A}G_{i,2j}^{\ell,\ell} + {}^{A}G_{i,2j+1}^{\ell,\ell}\right), \qquad {}^{B}G_{i,j}^{\ell,\ell-1} = \frac{1}{\sqrt{2}}\left({}^{B}G_{i,2j}^{\ell,\ell} + {}^{B}G_{i,2j+1}^{\ell,\ell}\right).$$

Obviously, ${}^{A}G_{i,j}^{\ell,\ell-1}$ is of Type A and ${}^{B}G_{i,j}^{\ell,\ell-1}$ of Type B. Note that the second equation can also be written as

$${}^{B}\hat{G}_{i,j}^{\ell,\ell-1} = \frac{1}{\sqrt{2}}\left({}^{B}\hat{G}_{i,2j}^{\ell,\ell} + {}^{B}\hat{G}_{i,2j+1}^{\ell,\ell}\right)$$

using ${}^{B}\hat{G}_{i,j}^{\ell,\ell'} := {}^{B}G_{i,j}^{\ell,\ell'}\,x_{i+1/2}^{\ell}$.

$G_{i,j}^{\ell,\ell-1}$ can be interpreted as the coefficients of $\Pi_\ell(\Phi_j^{\ell-1} * g_\ell) = \sum_i G_{i,j}^{\ell,\ell-1}\Phi_i^{\ell}$, since

$$\Pi_\ell(\Phi_j^{\ell-1} * g_\ell) = \Pi_\ell(\Phi_j^{\ell-1} * \sum_{k\in\mathbb{N}_0} g_k^{\ell}\Phi_k^{\ell}) = \sum_{k\in\mathbb{N}_0} g_k^{\ell}\Pi_\ell(\Phi_j^{\ell-1} * \Phi_k^{\ell}) = \sum_{k\in\mathbb{N}_0} g_k^{\ell}\mu_{i,j,k}^{\ell,\ell-1,\ell} = G_{i,j}^{\ell,\ell-1}.$$

Condensation from the fine mesh of level ℓ to the coarser at level $\ell-1$ is described by (5.20) and defines $G_{i,j}^{\ell-1,\ell-1}$. We apply this formula separately to ${}^{A}G_{i,j}^{\ell,\ell-1}$ and ${}^{B}G_{i,j}^{\ell,\ell-1}$. The term ${}^{A}G_{i,j}^{\ell,\ell-1}$ yields ${}^{AA}G_{i,j}^{\ell-1,\ell-1} + {}^{BA}G_{i,j}^{\ell-1,\ell-1}$ with

$${}^{AA}G_{i,j}^{\ell-1,\ell-1} = \frac{1}{\sqrt{2}}\left({}^{A}G_{2i,j}^{\ell,\ell-1} + {}^{A}G_{2i+1,j}^{\ell,\ell-1}\right),$$

$${}^{BA}\hat{G}_{i,j}^{\ell-1,\ell-1} = \frac{h_\ell}{4\sqrt{2}}\left({}^{A}G_{2i+1,j,k}^{\ell,\ell-1} - {}^{A}G_{2i,j,k}^{\ell,\ell-1}\right), \quad {}^{BA}G_{i,j}^{\ell-1,\ell-1} = {}^{BA}\hat{G}_{i,j}^{\ell-1,\ell-1}\,/x_{i+1/2}^{\ell-1}$$

(cf. (5.22)). The Type B term ${}^{B}G_{i,j}^{\ell,\ell-1}$ yields ${}^{BB}G_{i,j}^{\ell-1,\ell-1}$ which should be expressed by ${}^{BB}\hat{G}_{i,j}^{\ell-1,\ell-1} = x_{i+1/2}^{\ell-1}\,{}^{BB}G_{i,j}^{\ell-1,\ell-1}$:

$${}^{BB}\hat{G}_{i,j}^{\ell-1,\ell-1} = \frac{1}{\sqrt{2}}\left({}^{B}\hat{G}_{2i,j}^{\ell,\ell-1} + {}^{B}\hat{G}_{2i+1,j}^{\ell,\ell-1}\right)$$

(cf. (5.23)).

Gathering the different parts of $G_{i,j}^{\ell-1,\ell-1}$, we get

$$G_{i,j}^{\ell-1,\ell-1} = {}^{A}G_{i,j}^{\ell-1,\ell-1} + {}^{B}G_{i,j}^{\ell-1,\ell-1} \qquad \text{with} \tag{5.27a}$$

$${}^{A}G_{i,j}^{\ell-1,\ell-1} = \frac{1}{\sqrt{2}}\left({}^{A}G_{2i,j}^{\ell,\ell-1} + {}^{A}G_{2i+1,j}^{\ell,\ell-1}\right), \tag{5.27b}$$

$${}^{B}\hat{G}_{i,j}^{\ell-1,\ell-1} = \frac{h_\ell}{4\sqrt{2}}\left({}^{A}G_{2i+1,j}^{\ell,\ell-1} - {}^{A}G_{2i,j}^{\ell,\ell-1}\right) + \frac{1}{\sqrt{2}}\left({}^{B}\hat{G}_{2i,j}^{\ell,\ell-1} + {}^{B}\hat{G}_{2i+1,j}^{\ell,\ell-1}\right). \tag{5.27c}$$

As can be seen from this formulae, the implementation should use the quantities ${}^{A}G_{i,j}^{\ell-1,\ell-1}$ and ${}^{B}\hat{G}_{i,j}^{\ell-1,\ell-1}$ (instead of ${}^{B}G_{i,j}^{\ell-1,\ell-1} = {}^{B}\hat{G}_{i,j}^{\ell-1,\ell-1}/x_{i+1/2}^{\ell-1}$).

Combining the steps $G_{i,j}^{\ell,\ell} \mapsto G_{i,j}^{\ell,\ell-1}$ from (5.26) and $G_{i,j}^{\ell,\ell-1} \mapsto G_{i,j}^{\ell-1,\ell-1}$ from (5.27a-c), we obtain the following final result.

Lemma 5.7 *Let $\ell > 0$. The coefficients $G_{i,j}^{\ell-1,\ell-1} = {}^{A}G_{i,j}^{\ell-1,\ell-1} + {}^{B}G_{i,j}^{\ell-1,\ell-1}$ can be computed from $G_{i,j}^{\ell,\ell} = {}^{A}G_{i,j}^{\ell,\ell} + {}^{B}G_{i,j}^{\ell,\ell}$ by*

$${}^{A}G_{i,j}^{\ell-1,\ell-1} = {}^{A}G_{2i,2j}^{\ell,\ell} + \frac{1}{2}\left({}^{A}G_{2i,2j+1}^{\ell,\ell} + {}^{A}G_{2i+1,2j}^{\ell,\ell}\right), \tag{5.28a}$$

$${}^{B}\hat{G}_{i,j}^{\ell-1,\ell-1} = \frac{h_\ell}{8}\left({}^{A}G_{2i,2j+1}^{\ell,\ell} - {}^{A}G_{2i+1,2j}^{\ell,\ell}\right) + {}^{B}\hat{G}_{2i,2j}^{\ell,\ell} + \frac{1}{2}\left({}^{B}\hat{G}_{2i,2j+1}^{\ell,\ell} + {}^{B}\hat{G}_{2i+1,2j}^{\ell,\ell}\right). \tag{5.28b}$$

${}^{A}G_{i,j}^{\ell-1,\ell-1}$ is of Type A, while ${}^{B}G_{i,j}^{\ell-1,\ell-1}$ is of Type B.

Proof. Substitution of (5.26) into (5.27b) yields

$${}^{A}G_{i,j}^{\ell-1,\ell-1} = \frac{1}{2}\left({}^{A}G_{2i,2j}^{\ell,\ell} + {}^{A}G_{2i,2j+1}^{\ell,\ell} + {}^{A}G_{2i+1,2j}^{\ell,\ell} + {}^{A}G_{2i+1,2j+1}^{\ell,\ell}\right).$$

Since ${}^{A}G_{i,j}^{\ell,\ell}$ is of Type A, ${}^{A}G_{2i+1,2j+1}^{\ell,\ell} = {}^{A}G_{0,(2j+1)-(2i+1)}^{\ell,\ell} = {}^{A}G_{0,2j-2i}^{\ell,\ell} = {}^{A}G_{2i,2j}^{\ell,\ell}$ holds and allows the simplification in (5.28a).

Similarly, we first get

$$\begin{aligned}{}^{B}\hat{G}_{i,j}^{\ell-1,\ell-1} &= \frac{h_\ell}{8}\left({}^{A}G_{2i,2j}^{\ell,\ell} + {}^{A}G_{2i,2j+1}^{\ell,\ell} - {}^{A}G_{2i+1,2j}^{\ell,\ell} - {}^{A}G_{2i+1,2j+1}^{\ell,\ell}\right)\\ &\quad + \frac{1}{2}\left({}^{B}\hat{G}_{2i,2j}^{\ell,\ell} + {}^{B}\hat{G}_{2i,2j+1}^{\ell,\ell} + {}^{B}\hat{G}_{2i+1,2j}^{\ell,\ell} + {}^{B}\hat{G}_{2i+1,2j+1}^{\ell,\ell}\right)\end{aligned}$$

and use ${}^{A}G_{2i,2j}^{\ell,\ell} = {}^{A}G_{2i+1,2j+1}^{\ell,\ell}$ and ${}^{B}\hat{G}_{2i,2j}^{\ell,\ell} = {}^{B}\hat{G}_{2i+1,2j+1}^{\ell,\ell}$. ∎

5.8 Algorithm for $\Pi_{\ell''}(f_{\ell'} * g_\ell)$ in the Case $\ell'' \le \ell' \le \ell$

As stated in Problem 5.1, $f * g$ is split into convolutions $f_{\ell'} * g_\ell$ of $f_{\ell'} \in \mathcal{S}_{\ell'}$ and $g_\ell \in \mathcal{S}_\ell$, where $\ell' \le \ell$ can be assumed. The exact result $\omega_{\text{exact}} := f_{\ell'} * g_\ell$ is to be mapped into $\omega_{\ell''} := \Pi_{\ell''}\omega_{\text{exact}}$. More precisely, the components of $\omega_{\ell''} = \sum_i \omega_i^{\ell''}\Phi_i^{\ell''}$ for $i \in \mathcal{I}_{\ell''}$ are needed. By definition of $\Pi_{\ell''}$ we have local mass conservation in all intervals $I_i^{\ell''}$ $(i \in \mathcal{I}_{\ell''})$.

The case $\ell'' \le \ell' \le \ell$ considered here, was called Case A in [4]. The algorithm has a starting phase (Step 1), a first recursion from level ℓ down to $\ell' + 1$ (Step 2) and another recursion from level ℓ' down to $\ell'' + 1$ (Step 3).

5.8.1 Step 1

Using the coefficients of $g_\ell = \sum_i g_i^\ell \Phi_i^\ell$, define

$$ {}^A G_{i,0}^{\ell,\ell} := \frac{\sqrt{h_\ell}}{2}\left(g_i^\ell + g_{i-1}^\ell\right), \qquad {}^B \hat{G}_{i,0}^{\ell,\ell} := \frac{h_\ell^{3/2}}{12}\left(g_i^\ell - g_{i-1}^\ell\right). \tag{5.29} $$

Formally, the index i takes all values in $\mathbb{N}_0$. In practice, i must be restricted to a convex integer interval containing the support of $\left(g_i^\ell\right)_{i=0}^\infty$ (see details in [4]). Note that (5.29) is a reformulation of (5.25b,c) with $i-j$ replaced by i and makes use of ${}^A G_{i,j}^{\ell,\ell} = {}^A G_{i-j,0}^{\ell,\ell}$ etc.

5.8.2 Step 2

For $\lambda := \ell, \ell-1, \ldots, \ell'+1$ perform

$$ {}^A G_{i,0}^{\lambda-1,\lambda-1} = {}^A G_{2i,0}^{\lambda,\lambda} + \frac{1}{2}\left({}^A G_{2i-1,0}^{\lambda,\lambda} + {}^A G_{2i+1,0}^{\lambda,\lambda}\right), \tag{5.30} $$

$$ {}^B \hat{G}_{i,0}^{\lambda-1,\lambda-1} = \frac{h_\lambda}{8}\left({}^A G_{2i-1,0}^{\lambda,\lambda} - {}^A G_{2i+1,0}^{\lambda,\lambda}\right) + {}^B \hat{G}_{2i,0}^{\lambda,\lambda} + \frac{1}{2}\left({}^B \hat{G}_{2i-1,0}^{\lambda,\lambda} + {}^B \hat{G}_{2i+1,0}^{\lambda,\lambda}\right). $$

Finally, ${}^A G_{i,0}^{\ell',\ell'}$ and ${}^B \hat{G}_{i,0}^{\ell',\ell'}$ are computed[10].

Next, we want to compute the coefficients $\omega_i^{\ell'}$ of $\Pi_{\ell'}(f_{\ell'} * g_\ell) = \sum_i \omega_i^{\ell'} \Phi_i^{\ell'}$. They are represented by

$$ \omega_i^{\ell'} = \sum_{j,k\in\mathbb{N}_0} f_j^{\ell'} g_k^\ell \mu_{i,j,k}^{\ell',\ell',\ell} = \sum_{j\in\mathbb{N}_0} f_j^{\ell'} G_{i,j}^{\ell',\ell'}. $$

Splitting $\omega_i^{\ell'}$ into ${}^A\omega_i^{\ell'} + {}^B\omega_i^{\ell'}$ and using ${}^A G_{i,j}^{\ell',\ell'} = {}^A G_{i-j,0}^{\ell',\ell'}$ and ${}^B \hat{G}_{i,j}^{\ell',\ell'} = {}^B \hat{G}_{i-j,0}^{\ell',\ell'}$, we get the relations

$$ {}^A\omega_i^{\ell'} = \sum_{j\in\mathbb{N}_0} f_j^{\ell'}\, {}^A G_{i-j,0}^{\ell',\ell'}, \qquad {}^B\hat{\omega}_i^{\ell'} = \sum_{j\in\mathbb{N}_0} f_j^{\ell'}\, {}^B \hat{G}_{i-j,0}^{\ell',\ell'}. \tag{5.31a} $$

The sums in (5.31a) are discrete convolutions of the sequence $\left(f_j^{\ell'}\right)_{j=0}^\infty$ with $\left({}^A G_{i,0}^{\ell',\ell'}\right)_{i=0}^\infty$ or $\left({}^B \hat{G}_{i,0}^{\ell',\ell'}\right)_{i=0}^\infty$, respectively. Here, the fast Fourier transform can be used. For important remarks concerning the sizes of the supports of the sequences we refer to [4, Sect. 6].

Next, we convert the Type A coefficients into ${}^A\hat{\omega}_i^{\ell'} = {}^A\omega_i^{\ell'} x_{i+1/2}^{\ell'}$ and compute

$$ \hat{\omega}_i^{\ell'} = {}^A\omega_i^{\ell'}\, x_{i+1/2}^{\ell'} + {}^B\hat{\omega}_i^{\ell'}. \tag{5.31b} $$

5.8.3 Step 3

For $\lambda := \ell', \ell'-1, \ldots, \ell''+1$ we condense $\Pi_\lambda(f_{\ell'} * g_\ell)$ into $\Pi_{\lambda-1}(f_{\ell'} * g_\ell)$:

$$ \hat{\omega}_i^{\lambda-1} = \frac{1}{\sqrt{2}}\left(\hat{\omega}_{2i}^\lambda + \hat{\omega}_{2i+1}^\lambda\right). \tag{5.32a} $$

This is the application of (5.8). Finally, $\hat{\omega}_i^{\ell''}$ is computed. It can be converted into

$$ \omega_i^{\ell''} := \hat{\omega}_i^{\ell''} / x_{i+1/2}^{\ell''}. \tag{5.32b} $$

This yields the desired coefficients of $\Pi_{\ell''}(f_{\ell'} * g_\ell)$.

[10] These equations follow similarly from (5.28a,b). Note that because of the type A property, i.e., ${}^A G_{i-j,0}^{\lambda-1,\lambda-1} = {}^A G_{i,j}^{\lambda-1,\lambda-1}$, only a sequence ${}^A G_{i,0}^{\lambda-1,\lambda-1}$ with *one* index needs to be computed and stored.

5.8.4 Intertwining the Computations for all $\ell'' \le \ell' \le \ell$

In §5.7 the level ℓ containing the data g_ℓ is fixed. The coefficient $G^{\ell,\ell}_{i,j}$ defined in (5.25a) originates from g_ℓ, while $G^{\ell-1,\ell-1}_{i,j}$ from (5.27a) is obtained from $G^{\ell,\ell}_{i,j}$. In the final algorithm we include a loop over ℓ (renamed λ). When considering $G^{\ell-1,\ell-1}_{i,j}$ from (5.27a), we can add a new contribution from $g_{\ell-1}$ (that is (5.25a) with ℓ replaced by $\ell-1$). Hence, the computation of the ${}^AG^{\ell,\ell}_{i,j} = {}^AG^{\ell,\ell}_{i-j,0}$ and ${}^B\hat{G}^{\ell,\ell}_{i,j} = {}^B\hat{G}^{\ell,\ell}_{i-j,0}$ contributions is performed by the loop

$$
\boxed{\begin{array}{l}
\textbf{for } \lambda := L \textbf{ downto } 0 \textbf{ do} \\
\textbf{begin if } \lambda = L \textbf{ then } {}^AG^{L,L}_{i,0} := {}^B\hat{G}^{L,L}_{i,0} := 0 \textbf{ else} \\
\qquad \textbf{begin } {}^AG^{\lambda,\lambda}_{i,0} := {}^AG^{\lambda+1,\lambda+1}_{2i,0} + \frac{1}{2}\left({}^AG^{\lambda+1,\lambda+1}_{2i-1,0} + {}^AG^{\lambda+1,\lambda+1}_{2i+1,0}\right); \\
\qquad\qquad {}^B\hat{G}^{\lambda,\lambda}_{i,0} := \frac{h_{\lambda+1}}{8}\left({}^AG^{\lambda+1,\lambda+1}_{2i-1,0} - {}^AG^{\lambda+1,\lambda+1}_{2i+1,0}\right) \\
\qquad\qquad\qquad + {}^B\hat{G}^{\lambda+1,\lambda+1}_{2i,0} + \frac{1}{2}\left({}^B\hat{G}^{\lambda+1,\lambda+1}_{2i-1,0} + {}^B\hat{G}^{\lambda+1,\lambda+1}_{2i+1,0}\right) \\
\qquad \textbf{end}; \\
\qquad {}^AG^{\lambda,\lambda}_{i,0} := {}^AG^{\lambda,\lambda}_{i,0} + \frac{\sqrt{h_\lambda}}{2}\left(g^\lambda_i + g^\lambda_{i-1}\right); \; {}^B\hat{G}^{\lambda,\lambda}_{i,0} := {}^B\hat{G}^{\lambda,\lambda}_{i,0} + \frac{h^{3/2}_\lambda}{12}\left(g^\lambda_i - g^\lambda_{i-1}\right) \\
\textbf{end};
\end{array}} \tag{5.33}
$$

Line 7 adds the contributions (5.25b,c) involving g^λ_i from level λ.

The coefficients ${}^AG^{\lambda,\lambda}_{i,0}$ and ${}^B\hat{G}^{\lambda,\lambda}_{i,0}$ produced by (5.33) are needed in line 3 of the following algorithm, where the convolution of these coefficients with $f_{\ell'}$ yields ${}^A\omega^{\ell'}_i$, ${}^B\hat{\omega}^{\ell'}_i$. Note that ${}^A\omega^{\ell'}_i$, ${}^B\hat{\omega}^{\ell'}_i$ contain all contributions from $f_{\ell'} * g_\ell$ for the actual level ℓ' and all $\ell' \le \ell \le L$. Hence, $\hat{\omega}^{\ell'}_i$ from line 4 would yield $\omega^{\ell'}_i = \hat{\omega}^{\ell'}_i / x^{\ell'}_{i+1/2} = \Pi_{\ell'}(f_{\ell'} * g_\ell)$ (cf. (5.31b)). The addition of $(\hat{\omega}^{\ell'+1}_{2i} + \hat{\omega}^{\ell'+1}_{2i+1})/\sqrt{2}$ in line 5 (cf. (5.32a)) produces updates $\hat{\omega}^{\ell'}_i$ with the property that $\omega^{\ell'}_i$ from line 6 equals $\sum_{\lambda,\ell \text{ with } \ell' \le \lambda \le \ell} \Pi_{\ell'}(f_\lambda * g_\ell)$ (cf. (5.32b)).

$$
\boxed{\begin{array}{l}
\textbf{for } \ell' := L \textbf{ downto } 0 \textbf{ do} \\
\textbf{begin} \\
\qquad \text{compute } {}^A\omega^{\ell'}_i,\ {}^B\hat{\omega}^{\ell'}_i \text{ by the discrete convolutions (5.31a);} \\
\qquad \hat{\omega}^{\ell'}_i := {}^A\omega^{\ell'}_i\, x^{\ell'}_{i+1/2} + {}^B\hat{\omega}^{\ell'}_i\,; \\
\qquad \textbf{if } \ell' < L \textbf{ then } \hat{\omega}^{\ell'}_i := \hat{\omega}^{\ell'}_i + \frac{1}{\sqrt{2}}\left(\hat{\omega}^{\ell'+1}_{2i} + \hat{\omega}^{\ell'+1}_{2i+1}\right); \\
\qquad \omega^{\ell'}_i := \hat{\omega}^{\ell'}_i / x^{\ell'}_{i+1/2} \\
\textbf{end};
\end{array}} \tag{5.34}
$$

Note that a similiar procedure must be repeated with f and g interchanged, because of the second sum in (5.6). A slight modification is necessary to avoid $\ell = \ell'$.

5.9 Algorithm for $\Pi_{\ell''}(f_{\ell'} * g_\ell)$ in the Case $\ell' < \ell'' \le \ell$

While $\ell' \le \ell$ can be assumed without loss of generality, ℓ'' can take all values between 0 and L. After the first case $\ell'' \ge \ell'$ in §5.8, we now consider $\ell' < \ell'' \le \ell$. It corresponds to Case B in [4].

5.9.1 Explanations for $\ell'' = \ell' + 1$

We will use a loop of ℓ'' from $\ell' + 1$ to ℓ. Here we discuss the first value $\ell'' = \ell' + 1$ and assume $\ell' + 1 \leq \ell$.

The function $f_{\ell'} = \sum_j f_j^{\ell'} \Phi_j^{\ell'}$ can be transformed into a function of level $\ell' + 1$ by using (3.7):

$$f_{\ell'} = \sum_j \hat{f}_j^{\ell'+1} \Phi_j^{\ell'+1} \qquad \text{with } \hat{f}_{2j}^{\ell'+1} := \hat{f}_{2j+1}^{\ell'+1} := \frac{1}{\sqrt{2}} f_j^{\ell'}. \tag{5.35}$$

Let $\hat{f}_{\ell'+1} := \left(\hat{f}_j^{\ell'+1}\right)_{j\in\mathbb{Z}}$ be the sequence of the newly defined coefficients. Since $\ell'' = \ell'+1 \leq \ell$, the three level numbers $\ell'', \ell'+1, \ell$ satisfy the inequalities of Case A. As in Step 2 of Case A (see §5.8.2) the desired coefficients of the projection at level $\ell'' = \ell' + 1$ are obtainable by (5.31a).

5.9.2 Complete Recursion

Step 1 in Case A has already produced the coefficients ${}^{A}G_{i,0}^{\ell,\ell}$, ${}^{B}\hat{G}_{i,0}^{\ell,\ell}$. For $\ell'' = \ell'+1, \ell'+2, \ldots, \ell$ we represent the function $f_{\ell'}$ at these levels ℓ'' by computing the coefficients $\hat{f}_j^{\ell''}$ as in (5.35):

$$\hat{f}_j^{\ell'} := f_j^{\ell'} \qquad \text{(starting value)}, \tag{5.36a}$$

$$\hat{f}_{2j}^{\ell''} := \hat{f}_{2j+1}^{\ell''} := \frac{1}{\sqrt{2}} \hat{f}_j^{\ell''-1} \qquad (\ell' + 1 \leq \ell'' \leq \ell)\,. \tag{5.36b}$$

Note, however, that only those coefficients are to be determined which are really needed in the next step, which are the discrete convolutions

$${}^{A}\omega_i^{\ell''} = \sum_{j\in\mathbb{N}_0} f_j^{\ell''}\, {}^{A}G_{i-j,0}^{\ell'',\ell''}\,, \quad {}^{B}\hat{\omega}_i^{\ell''} = \sum_{j\in\mathbb{N}_0} f_j^{\ell''}\, {}^{B}\hat{G}_{i-j,0}^{\ell'',\ell''} \qquad (\ell' + 1 \leq \ell'' \leq \ell) \tag{5.36c}$$

of the sequence $\hat{f}_{\ell''} := (\hat{f}_j^{\ell''})_{j\in\mathbb{Z}}$ with $\left({}^{A}G_{i,0}^{\ell',\ell'}\right)_{i=0}^{\infty}$ and $\left({}^{B}\hat{G}_{i,0}^{\ell',\ell'}\right)_{i=0}^{\infty}$.

5.9.3 Combined Computations for all $\ell' < \ell'' \leq \ell$

The algorithm is

$$\begin{array}{|l|l}
\hline
\hat{f}_j^0 := 0; & \text{explanations:} \\
\textbf{for } \ell'' := 1 \textbf{ to } L \textbf{ do} & \\
\textbf{begin } \hat{f}_j^{\ell''-1} := \hat{f}_j^{\ell''-1} + f_j^{\ell''-1}; & \text{start value (5.36a),} \\
\qquad \hat{f}_{2j}^{\ell''} := \hat{f}_{2j+1}^{\ell''} := \hat{f}_j^{\ell''-1}/\sqrt{2}; & \text{see (5.36b),} \\
\qquad \text{compute } {}^{A}\omega_i^{\ell''},\ {}^{B}\hat{\omega}_i^{\ell''} \text{ by the discrete convolutions (5.36c);} & \text{see (5.36c),} \\
\qquad \omega_i^{\ell''} := {}^{A}\omega_i^{\ell''} + {}^{B}\hat{\omega}_i^{\ell''}\ /x_{i+1/2}^{\ell''} & \text{see (5.31b), (5.32b).} \\
\textbf{end}; & \\
\hline
\end{array} \tag{5.37}$$

The sum $\hat{f}_j^{\ell''-1} + f_j^{\ell''-1}$ in the third line defines $\hat{f}_j^{\ell''-1}$ as coefficients of $\sum_{\ell'=0}^{\ell''-1} f_{\ell'} = \sum_j \hat{f}_j^{\ell''-1} \Phi_j^{\ell''-1}$. Therefore the next two lines consider all combinations of $\ell' < \ell''$. Since ${}^{A}G_{i,0}^{\ell'',\ell''}$ and ${}^{B}\hat{G}_{i,0}^{\ell'',\ell''}$ contain all contributions from $\ell \geq \ell''$, $\omega_{\ell''}$ from line 6 is the projection $\Pi_{\ell''}\left(\sum_{\ell',\ell \text{ with } \ell'<\ell''\leq\ell} f_{\ell'} * g_\ell\right)$.

5.10 Algorithm for $\Pi_{\ell''}(f_{\ell'} * g_\ell)$ in the Case $\ell' \leq \ell < \ell''$

Now the step size $h_{\ell''}$ used by the projection $\Pi_{\ell''}$ is smaller than both $h_{\ell'}$ and h_ℓ.

The exact convolution $\omega_{\text{exact}} := f_{\ell'} * g_\ell$ is globally continuous and piecewise linear in the intervals $I_j^\ell = [jh_\ell, (j+1)\,h_\ell)$. Note that the intervals $I_i^{\ell''}$ lie completely in one of the I_i^ℓ intervals.

The cheapest approach is to compute first the evaluations δ_j^ℓ of ω_{exact} at the grid points jh_ℓ, i.e.,

$$\delta_j^\ell := (f_{\ell'} * g_\ell)\,(jh_\ell). \tag{5.38}$$

Then the coefficients of $\Pi_{\ell''}\omega_{\text{exact}} = \sum_{i\in\mathbb{N}_0} \omega_i^{\ell''}\Phi_i^{\ell''}$ follow from the explicit formula

$$\omega_i^{\ell''} = \frac{(h_{\ell''})^{3/2}}{6x_{i+1/2}^{\ell''}} \left\{ \left(3\,(2i+1)\left[(j+1)\,\delta_j^\ell - j\delta_{j+1}^\ell\right]\right) + \frac{2h_{\ell''}}{h_\ell}\left(1+3i+3i^2\right)\left(\delta_{j+1}^\ell - \delta_j^\ell\right) \right\}.$$

In particular for $\ell'' = \ell + 1$ (i.e., $h_{\ell''} = h_\ell/2$), we have

$$\omega_i^{\ell+1} = \begin{cases} \frac{h_{\ell+1}^{3/2}}{6x_{i+1/2}^{\ell+1}}\left((2+9j)\,\delta_j^\ell + (1+3j)\,\delta_{j+1}^\ell\right) & \text{for } i = 2j, \\ \frac{h_{\ell+1}^{3/2}}{6x_{i+1/2}^{\ell+1}}\left((2+3j)\,\delta_j^\ell + (7+9j)\,\delta_{j+1}^\ell\right) & \text{for } i = 2j+1. \end{cases} \tag{5.39}$$

It remains to compute the point values δ_j^ℓ from (5.38). This problem is already discussed in [4, §5.3.2]. We repeat the algorithm [4, (5.16)]:

for $\ell := 0$ **to** $L-1$ **do**
begin if $\ell = 0$ **then begin** $\hat{f}_i^0 := 0$; $\hat{\delta}_i^0 := 0$ **end else**
  begin compute $\hat{f}_i^\ell$ from $\hat{f}_i^{\ell-1}$ by (5.36b);
    $\delta_{2i}^\ell := \delta_i^{\ell-1}$; $\delta_{2i+1}^\ell := \frac{1}{2}\left(\delta_i^{\ell-1} + \delta_{i+1}^{\ell-1}\right)$
  end;
  $\hat{f}_i^\ell := \hat{f}_i^\ell + f_i^\ell$;
  compute δ_i^ℓ by the convolution $\sum_{j\in\mathbb{Z}} \hat{f}_j^\ell\, g_{i-j-1}^\ell$;
  $\hat{\delta}_i^\ell := \hat{\delta}_i^\ell + \delta_i^\ell$;
  compute $\omega_i^{\ell+1}$ from $\hat{\delta}_i^\ell$ by (5.39)
end; (5.40)

Line 4 is the linear interpolation of the data from level $\ell-1$ at level ℓ. Note that lines 3-5 are performed only for $\ell > 0$.

5.11 Cost

Because of the splitting in Type A and Type B coefficients, the algorithm is a little more involved than the algorithm for the L^2 projection of $f * g$ discussed in [4], but the cost of the present algorithm is only increased by a constant factor. Therefore, under the same assumptions as in [4] the overall cost is $\mathcal{O}(kN\log N)$, where N is the data size of factors $f, g \in \mathcal{S}$. The factor k is the number of terms in (1.3).

6 Treatment of the Last Interval

6.1 General Explanations

As discussed in the beginning of Section 2, the support of f lies in $[0, x_{\max}]$, which implies that all intervals $I \in \mathcal{M}$ are contained in $[0, x_{\max})$ with $x_{\max} := \sup\{x \in I : I \in \mathcal{M}\}$. Moreover there is a "last interval" $I_{\text{last}} = [x_{\max} - h_\ell, x_{\max}) \in \mathcal{M}$. In the following, we assume that[11] $\ell = 0$, i.e., I_{last} belongs to level 0.

Since functions $f, g \in \mathcal{S} = \mathcal{S}(\mathcal{M})$ with support in $[0, x_{\max}]$ yield a convolution $\omega_{\text{exact}} = f * g$ with support in $[0, 2x_{\max}]$, the projection $\omega_{\text{exact}} \mapsto \omega \in \mathcal{S}$ must include the truncation $\omega|_{[x_{\max}, 2x_{\max}]} := 0$. In the case of an L^2 orthogonal projection, the local replacement of $\omega|_{[x_{\max}, 2x_{\max}]}$ by zero describes the projection into the zero space. This, however, does not conserve mass. The necessary corrections are discussed next.

6.1.1 Case of $p = 0$

We extend the mesh $\mathcal{M}$ by

$$\bar{\mathcal{M}} := \mathcal{M} \cup \{[jh_0, (j+1)\, h_0) : j_{\max} \leq j \leq 2j_{\max} - 1\},$$

where $j_{\max} := x_{\max}/h_0 \in \mathbb{N}$. Note that $\bar{\mathcal{M}}$ covers the interval $[0, 2x_{\max})$. The function space corresponding to $\bar{\mathcal{M}}$ is

$$\bar{\mathcal{S}} := \mathcal{S}\left(\bar{\mathcal{M}}\right).$$

As discussed before, $f * g$ for $f, g \in \mathcal{S}$ have a support which is completely covered by $\bar{\mathcal{M}}$. Therefore, $\omega = \Pi\,(f * g)$ can be computed in $\bar{\mathcal{S}}$ by the algorithm from §§5.8-5.10.

The mass connected with the intervals in $\bar{\mathcal{M}} \backslash \mathcal{M}$ is

$$m_{[x_{\max}, 2x_{\max})}(\omega) = \sum_{j=j_{\max}}^{2j_{\max}-1} m_{I_j^0}(\omega) = \sum_{j=j_{\max}}^{2j_{\max}-1} \sqrt{h_0}\, x_{j+1/2}^0\, \omega_j^0,$$

where ω_j^0 are the coefficients in the representation $\omega|_{[x_{\max}, 2x_{\max}]} = \sum_{j=j_{\max}}^{2j_{\max}-1} \omega_j^0 \Phi_j^0$. Hence, mass conservation requires a correction

$$\omega_{j_{\max}-1}^0 := \omega_{j_{\max}-1}^0 + \frac{1}{x_{j_{\max}-1/2}^0} \sum_{j=j_{\max}}^{2j_{\max}-1} x_{j+1/2}^0\, \omega_j^0,$$
$$\omega_j^0 := 0 \qquad \text{for } j \geq j_{\max}.$$

The coefficient $\omega_{j_{\max}-1}^0$ belongs to the last interval $I_{\text{last}} = I_{j_{\max}-1}^0$, while $\omega_j^0 := 0$ for $j \geq j_{\max}$ is another writing for $\omega|_{[x_{\max}, 2x_{\max}]} := 0$.

[11] It makes no sense to use smaller intervals in a region where the function is expected to be so small that it can be approximated by zero. However, $I_{\text{last}} \in M_\ell$ for $\ell > 0$ can be treated similarly.

6.1.2 Case of $p > 0$

A correction is also required for $p > 0$. Again, $\omega = \Pi(f * g) \in \bar{\mathcal{S}}$ is to be computed in $\bar{\mathcal{S}} = \mathcal{S}(\bar{\mathcal{M}})$ using the extended mesh. Let $\omega^0_{(j,0)}$ and $\omega^0_{(j,1)}$ be the coefficients of the representation

$$\omega|_{[x_{\max},2x_{\max}]} = \sum_{j=j_{\max}}^{2j_{\max}-1} \sum_{\alpha=0}^{p} \omega^0_{(j,\alpha)} \Phi^0_{(j,\alpha)}$$

(in fact, one should avoid the computation of $\omega^0_{(j,\alpha)}$ $(j \geq j_{\max}, \alpha \geq 2)$, since these coefficients are not needed).

From (4.2a,b) we conclude that[12]

$$m_{I^0_j}(\omega) = \sqrt{h_0}\, x^0_{j+1/2}\, \omega^0_{(j,0)} + \frac{h_0^{3/2}}{\sqrt{12}}\, \omega^0_{(j,1)}.$$

Therefore the correction in the last interval is

$$\omega^0_{(j_{\max}-1,0)} := \omega^0_{(j_{\max}-1,0)} + \frac{1}{x^0_{j_{\max}-1/2}} \sum_{j=j_{\max}}^{2j_{\max}-1} \left(x^0_{j+1/2}\, \omega^0_{(j,0)} + \frac{h_0}{\sqrt{12}}\, \omega^0_{(j,1)} \right).$$

Note that only the coefficient $\omega^0_{(j_{\max}-1,0)}$ is changed, but not $\omega^0_{(j_{\max}-1,1)}$, since this choice corrects the mass with the smallest change in the L^2 norm of $\omega|_{[0,x_{\max}]}$.

7 Discretisation of the Sink Integral

We recall the integral $Q_{\mathrm{sink}}(f)(x) = 2f(x)\int_0^\infty \kappa(x,y)f(y)\mathrm{d}y$ from (1.1) and (1.2). We replace the kernel κ by the same approximation $\sum_{\nu=1}^{k} \alpha_\nu(x)\beta_\nu(y)$ (cf. (1.3)) as we did before. Then the sink term becomes

$$2f(x)\sum_{\nu=1}^{k} \alpha_\nu(x) \int_0^\infty \beta_\nu(y)f(y)\mathrm{d}y. \tag{7.1}$$

According to Remark 2.1c, α_ν and β_ν are assumed to be piecewise constant. Let $f \in \mathcal{S}$ be an ansatz function which is piecewise polynomial of degree p. Then the same statement holds for $\alpha_\nu f$ and $\beta_\nu f$. The integrations

$$\int_{I_i^\ell} \beta_\nu(y)f(y)\mathrm{d}y = \int_{ih_\ell}^{(i+1)h_\ell} \beta_\nu(y)f(y)\mathrm{d}y \qquad \text{for } I_i^\ell \in \mathcal{M}$$

can be performed exactly, since the integrand is a polynomial of degree p. Therefore, the sink integral (7.1) is evaluated exactly and has again values in $\mathcal{S}$, so that no projections are required:

$$2f(x)\sum_{\nu=1}^{k} \alpha_\nu(x) \sum_{I\in\mathcal{M}} \int_{I_i^\ell} \beta_\nu(y)f(y)\mathrm{d}y \in \mathcal{S}.$$

The required computational work is proportional to $k\,(p+1)\,N$.

[12] Here we use that $\Phi^\ell_{(j,\alpha)}$ $(\alpha \geq 2)$ is orthogonal to x, i.e., $m(\Phi^\ell_{(j,\alpha)}) = 0$.

If, different from Remark 2.1c, α_ν and β_ν are piecewise polynomials of degree $q > 0$, the integrand of $\int_{ih_\ell}^{(i+1)h_\ell} \beta_\nu(y) f(y)\mathrm{d}y$ is now a polynomial of degree $p + q$ and can again be evaluated exactly, whereas $\alpha_\nu f$ does not belong to $\mathcal{S}$ and needs a projection back into $\mathcal{S}$. If $p > 0$, the L^2 orthogonal projection conserves mass, while for $p = 0$ one must use the mass conserving projection $\alpha_\nu f \longmapsto F \in \mathcal{S}$ with the property $m_I(\alpha_\nu f) = m_I(F)$.

References

[1] F. Filbet and Ph. Laurençot: *Numerical simulation of the Smoluchowski coagulation equation.* SIAM J. Sci. Comput. **25** (2004) 2004-2028.

[2] S.K. Friedlander: *Smoke, dust, and haze. Fundamentals of aerosol dynamics. 2nd edition.* Oxford University Press, New York, 2000.

[3] W. Hackbusch: *On the efficient evaluation of coalescence integrals in population balance models.* Computing **78** (2006) 145-172.

[4] W. Hackbusch: *Fast and exact projected convolution for non-equidistant grids.* Computing 2007 (to appear).

[5] J. Koch, W. Hackbusch, and K. Sundmacher: *$\mathcal{H}$-matrix methods for linear and quasi-linear integral operators appearing in population balances.* Computers Chem. Engng. **31** (2007) 745-759.

[6] S. Qamar and G. Warnecke.: *Solving population balance equation for two-component aggregation by a finite volume scheme.* Chem. Eng. Sci. **62** (2006) 679-693.

[7] D. Ramkrishna: *Population balances. Theory and applications to particulate systems in engineering.* Academic Press, San Diego, 2000.

[8] D.J. Smit, M.J. Hounslow, and W.R. Paterson: *Aggregation and gelation - I. analytical solutions for cst and batch operation.* Chem. Eng. Sci. **49** (1994) 1025-1035.

[9] H. Struchtrup, M. Luskin, and M.R. Zachariah: *A model for kinetically controlled internal phase segregation during aerosol coagulation.* J. Aerosol Science **32** (2001) 1479-1504.