



**HAL**  
open science

## A finite volume preserving scheme on nonuniform meshes and for multidimensional coalescence

Louis Forestier-Coste, Simona Mancini

► **To cite this version:**

Louis Forestier-Coste, Simona Mancini. A finite volume preserving scheme on nonuniform meshes and for multidimensional coalescence. *SIAM Journal on Scientific Computing*, 2012, pp.B840-B860. 10.1137/110847998 . hal-00621069v2

**HAL Id: hal-00621069**

**<https://hal.science/hal-00621069v2>**

Submitted on 25 Oct 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A FINITE VOLUME PRESERVING SCHEME ON NON-UNIFORM MESHES AND FOR MULTIDIMENSIONAL COALESCENCE\*

L. FORESTIER-COSTE<sup>†</sup> AND S. MANCINI<sup>†</sup>

**Abstract.** In this paper we present a deterministic numerical approximation of the coalescence or Smoluchowski equation. The proposed numerical scheme conserves the first order momentum and deals with non-uniform grids. The generalization to a multidimensional framework is also described. We validate the scheme considering some classical tests both in one and two dimensions and discuss its behavior when gelation occurs. Our numerical results and code are compared with those already existent in literature.

**Key words.** finite volume method, non-uniform mesh, coalescence equation

**AMS subject classifications.** 65R20, 82C05

**1. Introduction.** In the last years, the resolution of the coalescence or Smoluchowski equation (see [24]) has been largely studied both from the theoretical and numerical points of view. In fact the successive merging of particles occurs in various physical phenomena spreading from planetary science to aerosols, polymers (see [18] and references therein) and bio-mathematics (see for example [4], [5] and [6]). From the numerical point of view, several methods has been proposed for solving the coalescence equation, either starting from the discrete form (see for example [16] and [17]) or considering the continuous form (see [7], [10], [14], [21], [22] and [17] for deterministic methods, or [1], [3], [8] and [9] for stochastic ones). More recently, some works concerning the numerical resolution of the multidimensional coalescence problem have been proposed (see for example [15], [21] and [23], ). In particular, [21] and [23] concern a numerical approach based on the conservative form of the coalescence equation discussed in the one-dimensional framework in [10] and [21]. We underline here that the numerical discretization we propose is based on the original coalescence equation and not on the conservative form introduced in [10].

Before entering the details of our study, we first recall the one-dimensional continuous and homogeneous in space Smoluchowski equation, we shall refer to as the coalescence equation. Let  $f(x, t)$  denote the distribution function of a set of particles at time  $t$  and with size  $x \geq 0$ . Note that, the size  $x$  usually represents the mass or the volume of a particle. Then the coalescence equation reads

$$\partial_t f = \frac{1}{2} \int_0^x H(x', x - x') f(x') f(x - x') dx' - f(x) \int_0^\infty H(x, x') f(x') dx', \quad (1.1)$$

where the coalescence kernel  $H(x, x')$  represents the merging rate of particles of size  $x$  and  $x'$ ; it is assumed to be positive,  $H(x, x') > 0$ , and symmetric,  $H(x, x') = H(x', x)$ . The first term on the right-hand side of equation (1.1), represents the gain term and it accounts for those particles of size  $x'$  merging with particles of size  $x - x'$  and giving particles of size  $x$ . In other words, when two particles merge their size just sum up, so that coalescence process conserves the total size of the particles involved with. The second term on the right-hand side of equation (1.1) is the loss term and accounts for

---

\*This work was partially funded by the ERC-starting grant DEMONS (n. 202844) under the European FP7. Authors thanks the referees for they many constructing comments.

<sup>†</sup>Fédération Denis Poisson (FR 2964), MAPMO (UMR 6628), BP. 6759, Université d'Orléans, F-45067 Orléans, France. (louis.forestier-coste@math.cnrs.fr, simona.mancini@univ-orleans.fr)

those particles of size  $x$  which disappear by coalescing with particles of size  $x'$ . The more general formulation for the coalescence kernel reads, see for example [10],

$$H(x, x') = x^\mu (x')^\nu + x^\nu (x')^\mu, \quad 0 \leq \mu \leq \nu \leq 1. \quad (1.2)$$

In this paper we are concerned with the numerical validation of the proposed scheme, we restrict the study to the following academic coalescence kernels, for which the coalescence equation (1.1) admits explicit solutions, but which aren't physically relevant. Choosing in (1.2),  $\mu = \nu = 0$  or  $\mu = \nu = 1$ , we respectively obtain, up to a factor 2, the constant kernel  $H(x, x') = 1$  or the multiplication one  $H(x, x') = xx'$ . We shall also consider the kernel  $H(x, x') = x + x'$  obtained by choosing in (1.2)  $\mu = 1$  and  $\nu = 0$ , or vice versa. We consider here only time independent kernels, but the proposed numerical scheme can be easily generalized to time dependent ones.

Let us now denote by  $M_p(t)$ , the  $p^{\text{th}}$  order moments associated to the distribution function  $f(t, x)$ :

$$\mathcal{M}_p(t) = \int_0^\infty x^p f(x, t) dx. \quad (1.3)$$

We recall that  $\mathcal{M}_0$  and  $\mathcal{M}_1$ , the zero and first order moments, respectively represent the density number of particles and their total size (total mass or volume). It is well known that, when considering for example the constant coalescence kernel  $H(x, x') = 1$  or the additive one  $H(x, x') = x + x'$ , the first order moment is conserved in time. On the other hand, the zero and second order moments are respectively decreasing and increasing in time. When considering the multiplicative coalescence kernel,  $H(x, x') = xx'$ , the gelation phenomena occurs, as was proved in [19]. This means that, due to the fast increment of the coalescence kernel, a runaway growth leads to the formation of particles with infinite size in finite time. Briefly, some matter escape from the system of particles, so that the first order moment  $\mathcal{M}_1$  is not conserved, but decreases in time, see for example [10].

We finally recall that equation (1.1) can be written in the following conservative form, see for example [10], [21] and [25]:

$$x \partial_t f = -\partial_x \int_0^x \int_{x-u}^\infty u H(u, v) f(v) f(u) dv du. \quad (1.4)$$

Many numerical schemes dealing with the discretization of equation (1.1) or (1.4) have been proposed in recent years, and it would be not possible to give an exhaustive list. In this paper, we will compare and validate our numerical scheme with respect to [10] and [21] concerning the one-dimensional case, and with respect to [21] concerning the multidimensional case. In particular, in [10] the numerical scheme is based on a finite volume discretization of (1.4), which is robust for non-uniform meshes, the conservation of the first order moment being given by the conservative form (1.4) and by the choice of the way to truncate the infinite integral appearing in (1.4). The multidimensional numerical scheme described in [21] is also based on a finite volume method and on the conservative form (1.4). We note that it is very efficient in terms of CPU times, but solutions in the two dimensional framework show singularities.

Our work has been suggested by the study of gas bubbles growth in magmas. Recent results in volcanology (see [2] and [12]) highlight that bubbles coalescence growing by decompression and exsolution in a volcano conduit is a relevant phenomena and may heavily affect the kind of eruption (effusive or explosive). The growth and

merging process of bubbles can be modeled at a statistical level by means of an expansion-coalescence equation, see [11]. Gas bubbles are described in terms of their volume and mass, which are in this framework independent variables (i.e. the perfect gas law doesn't hold during the bubbles growth), so that the bubble distribution function  $f = f(v, m, t)$  is defined on a bidimensional phase space and on time. The study of the expansion term is rather classical, it is governed by the volumes and masses growth rates and is described by a conservative equation. Whereas, defining and solving the coalescence term on a bidimensional domain is a novelty and need to be studied more in details, see [11]. We underline here that that bubbles coalescence implies that bubbles volumes and masses may growth of several orders of magnitude. Using a uniform mesh implies to have a huge number of discretization points in order to accurately capture the evolution of smaller and bigger bubbles and so very long CPU times. Moreover, bigger bubbles do not need to be discretized on a thin mesh. Therefore, a numerical scheme defined for non-uniform meshes seems a good choice.

In this paper, we propose a numerical scheme based on a finite volume approach for equation (1.1), which can be used with non-uniform meshes and the overlaps of the created particles with the defined mesh, and which conserves the discrete first order moment, at least until the gelation time is reached. We note that, the use of non-uniform meshes allows to consider bigger domains than uniform ones and to get smaller CPU times, for instance, when considering a non-uniform mesh with half of the discretizations points with respect to the uniform one, the CPU time diminishes of one order of magnitude. Moreover, the first order moment conservation is an essential requirement for the volcanology problem because the mass conservation of two merging bubbles must be verified. The numerical scheme we propose is second order accurate and consistent with the continuous equation. The numerical discretization will be detailed in the one-dimensional framework, being its generalization to the multidimensional case straight forward. The numerical validation will be done both for the one-dimensional and the two-dimensional cases.

This paper is organized as follows. In section 2 we detail the one-dimensional numerical approximation. Section 3 is devoted to its multidimensional generalization. Numerical results and the validation of the code are presented and discussed both for the one and two dimensional problems in section 4. Finally, section 5 resumes our results and presents research axes for future works.

**2. One-dimensional scheme.** Let us first introduce the one-dimensional numerical scheme we will apply in our simulations. How to obtain it as well as how to handle the overlap and the first order moment conservation will be described in sections 2.1 and 2.2.

We choose to discretize the coalescence equation (1.1) applying a finite volume approximation. Let us split  $\mathbb{R}^+$  in the cells  $i = [x_{i-1/2}, x_{i+1/2}]$ , for  $i \in \mathbb{N}$ , with the usual assumption  $x_{-1/2} = 0$ , see for example [10], and let us define the grid points and the mesh step by:

$$x_i = \frac{x_{i-1/2} + x_{i+1/2}}{2}, \quad \Delta x_i = x_{i+1/2} - x_{i-1/2}.$$

We shall denote in the sequel the merging of two cells  $j$  and  $k$  giving a cell  $i$  by  $i = (j + k)$ , so that the bounds of the created cell are:  $x_{i-1/2} = x_{j-1/2} + x_{k-1/2}$  and  $x_{i+1/2} = x_{j+1/2} + x_{k+1/2}$ .

For  $n \in \mathbb{N}$ , let us consider the discrete time  $t^{n+1} = t^n + \Delta t^n$ . For  $i$  and  $j \in \mathbb{N}$ , we denote by  $f_i^n$  the average value of  $f$  at time  $t^n$  on the cell  $i$ , which is an approximation of  $f(x_i, t^n)$  and by  $H_{i,j}$  the discrete coalescence kernel,  $H_{i,j} = H(x_i, x_j)$ .

Let us define the sets  $S^i$  and  $R_{j,k}$  by

$$S^i = \{(j, k) \in \mathbb{N} \times \mathbb{N} : (j + k) \cap i \neq \emptyset\}, \quad (2.1)$$

$$R_{j,k} = \{i \in \mathbb{N} : i \cap (j + k) \neq \emptyset\}, \quad (2.2)$$

They respectively represent the set of mesh couples  $(j, k)$  such that their sum intersect the mesh  $i$ , and the set of meshes  $i$  intersecting the sum of the meshes  $j$  and  $k$ .

With this notations, our finite volume numerical approximation reads

$$f_i^{n+1} = f_i^n + \Delta t^n \left( \frac{1}{2} \sum_{(j,k) \in S^i} \tilde{H}_{j,k} f_j^n f_k^n \lambda_{j,k}^i \frac{\Delta x_j \Delta x_k}{\Delta x_i} - \sum_{j=0}^{\infty} H_{i,j} f_i^n f_j^n \Delta x_j \right), \quad (2.3)$$

where  $\tilde{H}_{j,k}$  is a modified kernel ensuring the first order moment conservation, see section 2.2, and  $\lambda_{j,k}^i$  is a proportionality coefficient arising when considering overlaps, see section 2.1.

**2.1. Cells overlap.** In this section we explain how we handle the cells overlap, that is how we compute the term  $\sum_{(j,k) \in S^i}$ .

Let us write the finite volume discretization for (1.1). Integrating over a cell  $i$ , we have:

$$\begin{aligned} \partial_t \int_{x_{i-1/2}}^{x_{i+1/2}} f dx &= \frac{1}{2} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_0^x H(x-x', x') f(x-x') f(x') dx' dx - \\ &\int_{x_{i-1/2}}^{x_{i+1/2}} \int_0^\infty H(x, x') f(x) f(x') dx' dx. \end{aligned} \quad (2.4)$$

Applying a midpoint quadrature formula, the term on the left-hand side of (2.4) reads

$$\partial_t f_i(t) \Delta x_i, \quad (2.5)$$

and the last term on the right-hand side of (2.4), that is the loss term, becomes

$$\sum_{j=0}^{\infty} H_{i,j} f_i f_j \Delta x_i \Delta x_j. \quad (2.6)$$

Before dealing with the discretization of the gain term in (2.4), let us introduce and define some elements we will need. When considering non-uniform grids, the sum of two cells, say  $j$  and  $k$ , may give a cell, say  $i$ , overlapping various cells and which extrema do not coincide with those of the cell  $(j+k)$ . For instance in the volcanology problem, given a constant  $h$ , we define the non-uniform mesh as:  $x_i = i^3 h$ . Fermat's theorem, see [27], ensures that given a cell  $j$  and a cell  $k$ , it doesn't exist a cell  $i$  such that we exactly have  $i = (j+k)$ . Hence the cell  $i$  intersects one or more cells. Figure 2.1 shows the three basic overlaps configurations, but we may have any their linear combination. In order to take into account the overlap, let us define the indexes  $I_i(j)$  and  $J_i(j)$  such that they are the largest integers verifying:

$$x_{I_i(j)+1/2} + x_{j+1/2} < x_{i+1/2}, \quad x_{J_i(j)-1/2} + x_{j-1/2} < x_{i+1/2}, \quad (2.7)$$

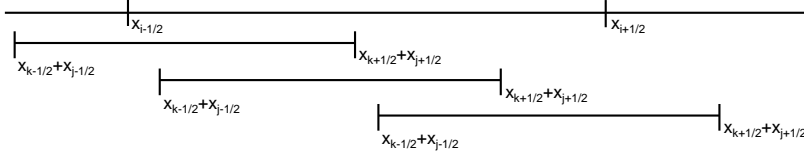


FIG. 2.1. Three basic configurations for overlaps.

where the first condition imposes that the integer  $I_i(j)$  is such that the upper bound of the created cell  $I_i(j) + j$  must be smaller than the one of the cell  $i$ ; and the second condition imposes that the integer  $J_i(j)$  is such that the lower bound of the created cell  $J_i(j) + j$  must be smaller than the upper bound of the cell  $i$ . We note that for all  $j \geq i$  the indexes  $I_i(j)$  and  $J_i(j)$  don't exist. The overlap appears in between  $I_i(j)$  and  $J_i(j)$  so that we have to consider only a portion of it.

We now detail the three situations which will be considered in the sequel, see (2.17). The portion of cells over which each sum is computed is represented in figures 2.2 to 2.4 by the dashed lines.

The first sum is the integral over the cells  $j$  and  $k$  such that  $(j, k) \in S^i$  and  $x_{j+1/2} + x_{k+1/2} < x_{i+1/2}$ . It corresponds to  $\sum_{k=I_i(j)+1}^{I_i(j)}$ .

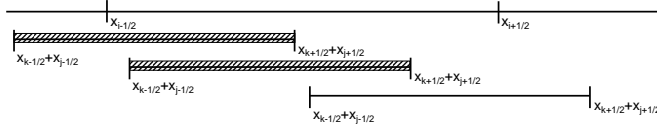


FIG. 2.2. First sum representation.

The second sum is the integral on the cells  $j$  and  $k$  such that  $(j, k) \in S^i$  and  $x_{j+1/2} + x_{k+1/2} > x_{i+1/2}$ . It corresponds to  $\sum_{k=I_i(j)+1}^{J_i(j)}$ .

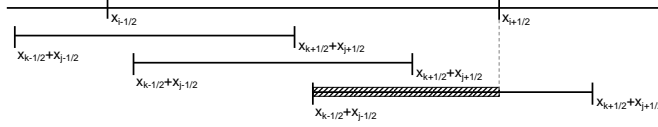


FIG. 2.3. Second sum representation.

The third sum is the integral on the cells  $j$  and  $k$  such that  $(j, k) \in S^{i-1}$  and  $x_{j+1/2} + x_{k+1/2} > x_{(i-1)+1/2} = x_{i-1/2}$ . It corresponds to  $\sum_{k=I_{i-1}(j)+1}^{J_{i-1}(j)}$ .

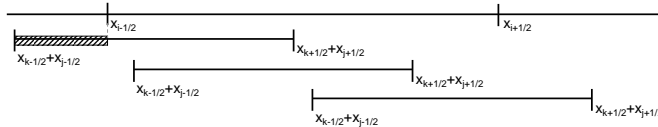


FIG. 2.4. Third sum representation.

Let us introduce  $\overline{m}_{j,k}^i$  and  $\underline{m}_{j,k}^i$  respectively denoting the maximum and minimum

bounds of the intersection of the cell  $(j+k)$  with a given cell  $i$ :

$$\overline{m_{j,k}^i} = \min(x_{i+1/2}, x_{j+1/2} + x_{k+1/2}), \quad (2.8)$$

$$\underline{m_{j,k}^i} = \max(x_{i-1/2}, x_{j-1/2} + x_{k-1/2}), \quad (2.9)$$

and let us define the proportionality factor  $\lambda_{j,k}^i$ , for a given cell  $i$ , as

$$\lambda_{j,k}^i = \left( \frac{\overline{m_{j,k}^i} - \underline{m_{j,k}^i}}{\Delta x_j + \Delta x_k} \right)_+. \quad (2.10)$$

Note that the proportionality coefficient  $\lambda_{j,k}^i$  is such that  $0 \leq \lambda_{j,k}^i \leq 1$ , the equality sign holding either when the intersection is empty ( $\lambda_{j,k}^i = 0$ ) or when the sum of the cells  $(j+k)$  is entirely contained in the cell  $i$  ( $\lambda_{j,k}^i = 1$ ). We also note that it is symmetric,  $\lambda_{j,k}^i = \lambda_{k,j}^i$ .

Omitting for simplicity the time dependence of the distribution function  $f$ , we now consider the discretization of the gain term in (2.4) and prove that

PROPOSITION 2.1. *Let  $S^i$  and  $\lambda_{j,k}^i$  be respectively defined by (2.1) and (2.10). Then the gain term on the right-hand side of equation (2.4) is approximated by*

$$\sum_{(j+k) \in S^i} H_{j,k} f_j f_k \lambda_{j,k}^i \Delta x_j \Delta x_k. \quad (2.11)$$

*Proof.* Inverting the order of integration, applying the change of variable  $z = x - x'$ , inverting again the order of integration and replacing  $z$  by  $x$ , the gain term in (2.4) splits in the sum of two integrals respectively defined on the intervals  $[0, x_{i+1/2}] \times [0, x_{i+1/2} - x']$  and  $[0, x_{i-1/2}] \times [0, x_{i-1/2} - x']$ :

$$\begin{aligned} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_0^x H(x-x', x') f(x-x') f(x') dx' dx = & \quad (2.12) \\ \int_0^{x_{i+1/2}} \int_0^{x_{i+1/2}-x'} H(x', x) f(x') f(x) dx dx' - & \\ \int_0^{x_{i-1/2}} \int_0^{x_{i-1/2}-x'} H(x', x) f(x') f(x) dx dx'. & \end{aligned}$$

The two integrals on the right-hand side of (2.12) are treated by means of the same technique, we detail here only the first one. It can be split in the sum of the integrals defined on each cell  $j$ , with  $j \leq i$ . That is:

$$\sum_{j=0}^i \int_{x_{j-1/2}}^{x_{j+1/2}} \int_0^{x_{i+1/2}-x'} H(x', x) f(x') f(x) dx dx'. \quad (2.13)$$

The overlap then appears in between  $I_i(j)$  and  $J_i(j)$ . In other words, in (2.13) the second integral can be split in the sum of all the integrals defined on the cells  $k$  such that, when summed with the cell  $j$ , they have the upper bound smaller than  $I_i(j)$  and those such that the summed lower bound is bigger than  $I_i(j)$  and smaller than

$J_i(j)$ . This gives:

$$\sum_{j=0}^i \int_{x_{j-1/2}}^{x_{j+1/2}} \left( \sum_{k=0}^{I_i(j)} \int_{x_{k-1/2}}^{x_{k+1/2}} H(x', x) f(x') f(x) dx + \sum_{k=I_i(j)+1}^{J_i(j)} \int_{x_{k-1/2}}^{x_{k+1/2}} H(x', x) f(x') f(x) \Lambda_{j,k}^i dx \right) dx'. \quad (2.14)$$

Note that we have introduced the weight  $\Lambda_{j,k}^i$  which represents the portion of the cell  $i$  overlapping with  $(j+k)$  and that we need only for theoretical aspects, but not for the implementation of the numerical scheme which is based on (2.11). This coefficient appears only in the second sum, since in the first one it is equal to 1, being the summed cells entirely contained in the cell  $i$ .

Analogously, the second integral on the right-hand side of (2.12) writes

$$\sum_{j=0}^{i-1} \int_{x_{j-1/2}}^{x_{j+1/2}} \left( \sum_{k=0}^{I_{i-1}(j)} \int_{x_{k-1/2}}^{x_{k+1/2}} H(x', x) f(x') f(x) dx + \sum_{k=I_{i-1}(j)+1}^{J_{i-1}(j)} \int_{x_{k-1/2}}^{x_{k+1/2}} H(x', x) f(x') f(x) \Lambda_{j,k}^{i-1} dx \right) dx'. \quad (2.15)$$

We can now apply a quadrature formula to (2.14) and (2.15) and arrange their sum as follows:

$$\begin{aligned} & \sum_{j=0}^i \left( \sum_{k=I_{i-1}(j)+1}^{I_i(j)} H_{j,k} f_j f_k \Delta x_j \Delta x_k + \sum_{k=I_i(j)+1}^{J_i(j)} H_{j,k} f_j f_k \Lambda_{j,k}^i \Delta x_j \Delta x_k - \right. \\ & \quad \left. \sum_{k=I_{i-1}(j)+1}^{J_{i-1}(j)} H_{j,k} f_j f_k \Lambda_{j,k}^{i-1} \Delta x_j \Delta x_k \right) + \\ & \quad \sum_{k=0}^{I_{i-1}(i)} H_{i,k} f_i f_k \Delta x_i \Delta x_k + \sum_{k=I_{i-1}(i)+1}^{J_{i-1}(i)} H_{i,k} f_i f_k \Lambda_{i,k}^{i-1} \Delta x_i \Delta x_k. \end{aligned} \quad (2.16)$$

Since  $I_{i-1}(i)$  and  $J_{i-1}(i)$  don't exist, the last two sums in (2.16) are null. Therefore, the numerical approximation of the gain term in (2.4) reads

$$\begin{aligned} & \sum_{j=0}^i \left( \sum_{k=I_{i-1}(j)+1}^{I_i(j)} H_{j,k} f_j f_k \Delta x_j \Delta x_k + \sum_{k=I_i(j)+1}^{J_i(j)} H_{j,k} f_j f_k \Lambda_{j,k}^i \Delta x_j \Delta x_k - \right. \\ & \quad \left. - \sum_{k=I_{i-1}(j)+1}^{J_{i-1}(j)} H_{j,k} f_j f_k \Lambda_{j,k}^{i-1} \Delta x_j \Delta x_k \right), \end{aligned} \quad (2.17)$$

where each sum corresponds to one of the sums represented in figures 2.2 to 2.4. Considering definition (2.1), and defining  $\lambda_{j,k}^i$  as a combination of 1,  $\Lambda_{j,k}^i$  and  $-\Lambda_{j,k}^{i-1}$  we can collect the three sums in (2.17), obtaining (2.11).  $\square$



Considering (2.5), (2.6), (2.11) and discretizing the time derivative, the finite volume discretization of equation (1.1) reads

$$f_i^{n+1} = f_i^n + \Delta t^n \left( \frac{1}{2} \sum_{(j,k) \in S^i} H_{j,k} f_j^n f_k^n \lambda_{j,k}^i \frac{\Delta x_j \Delta x_k}{\Delta x_i} - \sum_{j=0}^{\infty} H_{i,j} f_i^n f_j^n \Delta x_j \right). \quad (2.18)$$

This discretization is constructed in order to take care of the overlaps in the non-uniform mesh framework, but the conservation of the first order moment at a numerical level is not ensured. We shall deal with it in the next section.

We end this section with a remark concerning uniform meshes, showing that the proposed discretization works in this framework too.

**PROPOSITION 2.2.** *Let the mesh be defined by  $x_{i-1/2} = i\Delta x$ , for  $i \in \mathbb{N}$ , then the proportionality coefficients  $\lambda_{j,k}^i$  are given by:*

$$\lambda_{j,k}^i = \begin{cases} 1/2 & \text{if } i = \{j+k, j+k+1\}, \\ 0 & \text{otherwise.} \end{cases}$$

*Proof.* Consider the sum  $(j+k) = i$ . Then the cell  $i$  has bounds:

$$x_{i-1/2} = (j+k)\Delta x, \quad x_{i+1/2} = (j+k+2)\Delta x,$$

its size is  $2\Delta x$ , and its centre  $x_i$  is given by  $x_i = (j+k+1)\Delta x$ . Choosing  $s = j+k+1$  then by definition  $x_i = x_{s-1/2}$ , that is: the centre of the new cell  $i$  fall on the lower border of the mesh indexed by  $s = j+k+1$ . Since the lower bound of the cell  $s$  coincides with the upper bound of the cell  $s-1$ , the created cell  $i$  with centre in  $s$  and dimension  $2\Delta x$  covers both cells  $s$  and  $s-1$ .

Finally, computing  $\lambda_{j,k}^s$  and  $\lambda_{j,k}^{s-1}$  by means of (2.10) gives  $\lambda_{j,k}^s = \lambda_{j,k}^{s-1} = 1/2$ .  $\square$

**2.2. First order moment conservation.** We want to give a numerical approximation of the coalescence equation (1.1) conserving the discrete first order moment, since we know that the first order moment is conserved at the continuous level, at least until the gelation time is reached. As it is shown later on, see figure 4.5, for the multiplication kernel  $H(x, x') = xx'$ , the conservation of the first order moment is assured until the gelation time is reached, the proposed discretization is consistent with the conservation property of the continuous equation. In order ensure this conservation, we choose to modify the discrete coalescence kernel  $H_{j,k}$  in the gain term by defining a kernel  $\tilde{H}_{j,k}$  in such a way that the previous property holds true for every discrete time  $t^n$ .

Consider definition (1.3) and define the discrete momentum of order  $p$  at time  $t^n$

$$M_p^n = \sum_{i=0}^{\infty} x_i^p f_i^n \Delta x_i. \quad (2.19)$$

Then, for  $p = 1$  and  $n \in \mathbb{N}$ , the discrete first order moment conservation reads

$$\sum_{i=0}^{\infty} f_i^{n+1} x_i \Delta x_i = \sum_{i=0}^{\infty} f_i^n x_i \Delta x_i.$$

Therefore, replacing in (2.18)  $H_{j,k}$  by  $\tilde{H}_{j,k}$ , multiplying it by  $x_i$  and summing over all  $i$  we must have:

$$\frac{1}{2} \sum_{i=0}^{\infty} \sum_{(j,k) \in S^i} \tilde{H}_{j,k} f_j^n f_k^n \lambda_{j,k}^i x_i \Delta x_j \Delta x_k = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} H_{i,j} f_i^n f_j^n x_i \Delta x_j \Delta x_i. \quad (2.20)$$

PROPOSITION 2.3. *The discrete first order moment (2.19) is conserved if and only if the modified kernel  $\tilde{H}_{j,k}$  is defined by:*

$$\tilde{H}_{j,k} = H_{j,k} \frac{2(x_j + x_k)}{\sum_{i \in R_{j,k}} x_i (\lambda_{j,k}^i + \lambda_{k,j}^i)}. \quad (2.21)$$

*Proof.* Let us first prove that (2.21) is necessary by considering a particular discrete distribution function  $f_i^n$  of discrete Dirac type. For any index  $j$  and  $k$ , assume that  $f_i^n = 1$  if  $i = j, k$  and that  $f_i^n = 0$  otherwise. This is equivalent to consider only those terms in the sums that act on  $H_{j,j}$ ,  $H_{j,k}$  and  $H_{k,k}$ . Then the right-hand side of (2.20), gives

$$(x_j + x_k) H_{j,k} f_j^n f_k^n \Delta x_j \Delta x_k.$$

Concerning the left-hand side of (2.20), since the first sum is done over all  $i$ , it must exist at least one index  $s$  such that  $(j, k) \in S^s$ . Then the left-hand side of (2.20) becomes

$$\sum_{s \in R_{j,k}} \tilde{H}_{j,k} f_j^n f_k^n \lambda_{j,k}^s x_s \Delta x_j \Delta x_i.$$

Recalling the first moment conservation given by (2.20), it must be:

$$\sum_{s \in R_{j,k}} \tilde{H}_{j,k} f_j^n f_k^n \lambda_{j,k}^s x_s \Delta x_j \Delta x_k = (x_j + x_k) H_{j,k} f_j^n f_k^n \Delta x_j \Delta x_k.$$

Hence, (2.20) implies (2.21) at least for discrete Dirac type distributions.

We now prove that (2.21) implies (2.20) for any function  $f_i^n, f_j^n$ . Let us consider the canonical basis  $\mathbb{E}_{jk}$  of the set of matrix with real coefficients, which is  $\mathbb{E}_{jk}$  is the matrix with all coefficient equal to zero except at the  $j^{\text{th}}$  line and  $k^{\text{th}}$  column. Then, equation (2.20) can be written using tensors algebra as:

$$\tilde{\mathbb{H}} : \mathbb{F} = \mathbb{H} : \mathbb{F}, \quad (2.22)$$

where  $\mathbb{F}$ ,  $\mathbb{H}$  and  $\tilde{\mathbb{H}}$  respectively are the following tensors:

$$\mathbb{F} = \sum_{j,k} f_j^n f_k^n \mathbb{E}_{jk}, \quad \mathbb{H} = 2 \sum_{j,k} H_{j,k} (x_j + x_k) \Delta x_j \Delta x_k \mathbb{E}_{jk},$$

and

$$\tilde{\mathbb{H}} = \sum_{jk} \tilde{H}_{j,k} \left( \sum_{i=0}^{\infty} x_i \lambda_{j,k}^i \mathbb{1}_{i \in R_{j,k}} \right) \Delta x_j \Delta x_k \mathbb{E}_{jk}.$$

Since (2.22) is linear in  $\mathbb{F}$ , and it holds for the basis elements  $\mathbb{E}_{jk}$  then it holds for any function  $f_i^n$ .  $\square$

PROPOSITION 2.4. *In the uniform grid case  $\tilde{H}_{j,k}$  is equivalent to  $H_{j,k}$ .*

*Proof.* As proved in Proposition 2.2, the only cells  $i$  for which the  $\lambda_{j,k}^i$  in (2.21) are not zero are  $i = j + k + 1$  and  $i = j + k$ . Hence from relation (2.21), recalling the definition of  $x_i$  and of  $\lambda_{j,k}^i$ , and simplifying, we get:

$$\tilde{H}_{j,k} = H_{j,k} \frac{x_j + x_k}{(x_{j+k+1} + x_{j+k})/2} = H_{j,k} \frac{2(j+k+1)}{2j+2k+2} = H_{j,k},$$

concluding the proof.  $\square$

Consider a positive initial data  $f_i^0 > 0$  for all  $i$ . Since we are dealing with an explicit in time scheme, the solution  $f_i^n$  may become negative. In order to ensure its positivity we must define a stability (CFL) condition for the time step. Denoting respectively by  $G_i^n$  and  $L_i^n$  the gain and loss integrals in (2.3), the condition reads

$$\Delta t^n < \min_i \left( \left| \frac{f_i^n}{G_i^n - L_i^n} \right| \right). \quad (2.23)$$

We note that our scheme is consistent. In fact, considering a constant distribution function  $f(x, t)$ , then the loss term (2.6) corresponds to the approximation of  $\int H(x, x') dx'$ . Concerning the gain term, consistency may be proved repeating the same procedure applied to deduce (2.17), but with  $f(x, t)$  constant. Moreover, from the numerical results (see Figures 4.2 and 4.3), we observe that our scheme is second order accurate and thus that it is consistent.

**3. Multidimensional discretization.** Before dealing with the validation of the numerical scheme (2.3), we give in this section the generalization to the multidimensional case. In fact, as explained in the introduction, our study is suggested by an application in volcanology concerning two dimensional distribution functions  $f(t, x, y)$ . In the sequel we will detail the multidimensional generalization of our scheme, although the numerical simulations are performed in the two-dimensional framework.

For  $d > 1$ , define  $\mathbb{R}_+^d$  the space of vectors  $\vec{x} = \{x_r\}$ ,  $r = 1, \dots, d$ , such that  $x_r \geq 0$  for all  $r$ , let  $\vec{x} \in \mathbb{R}_+^d$  and  $t > 0$ , let also  $f(\vec{x}, t)$  denote the particle density function representing the distribution of particles which at time  $t$  have sizes  $\vec{x}$ . Then the coalescence equation reads

$$\partial_t f = \frac{1}{2} \int_0^{\vec{x}} H(\vec{x}', \vec{x} - \vec{x}') f(\vec{x}') f(\vec{x} - \vec{x}') d\vec{x}' - \int_0^{\vec{x}} H(\vec{x}, \vec{x}') f(\vec{x}) f(\vec{x}') d\vec{x}', \quad (3.1)$$

where  $H(\vec{x}, \vec{x}')$  is the coalescence kernel and

$$d\vec{x}' = \prod_{r=1}^d dx'_r.$$

Like in the one-dimensional case the kernel will be here independent on time and we will consider, in the numerical simulations, only the constant case:  $H(\vec{x}, \vec{x}') = 1$ .

Given  $p = \sum_{r=1}^d \alpha_r$ , with  $\alpha_r \geq 0$ , let us define the multidimensional moment of order  $p$ ,  $\mathcal{M}_p(t)$ , by:

$$\mathcal{M}_p(t) = \int_0^{\vec{\infty}} \prod_{r=1}^d x_r^{\alpha_r} f(\vec{x}, t) d\vec{x}. \quad (3.2)$$

Choosing  $\alpha_r = 0$  for each  $r$ , we obtain the zero order moment

$$\mathcal{M}_0(t) = \int_0^{\vec{\infty}} f(\vec{x}, t) d\vec{x},$$

which represents the density number of particles and is decreasing in time. Concerning the first order moment, we choose one particular value  $\hat{r}$  for which  $\alpha_{\hat{r}} = 1$ , and fix all the other values  $r$  to zero:

$$\alpha_r = \begin{cases} 1 & r = \hat{r} \\ 0 & \text{otherwise.} \end{cases}$$

Then the first order moment with respect to the size  $x_{\hat{r}}$ ,

$$\int_0^{\infty} x_{\hat{r}} f(\vec{x}, t) d\vec{x},$$

represents the total size in the  $x_{\hat{r}}$  component and, for the constant kernel  $H(\vec{x}, \vec{x}') = 1$ , is constant in time. Moreover, from the conservation of each first order moment for  $r = 1, \dots, d$ , we can easily deduce the conservation of

$$\mathcal{M}_1(t) = \int_0^{\infty} \Phi(\vec{x}) f(\vec{x}, t) d\vec{x}, \quad (3.3)$$

for all linear combination  $\Phi(\vec{x})$  of  $x_r$ , and in particular for the sum over all sizes:  $\Phi(\vec{x}) = \sum_{r=1}^d x_r$ . We will see that while at a continuous level the first order moments are conserved for each component  $x_{\hat{r}}$ , this will not be the case at the discrete level. We finally recall that in the multidimensional case too, the moments of order  $p > 1$  are increasing in time.

We now detail the numerical scheme. For each  $r = 1, \dots, d$ , let us split  $\mathbb{R}_+^d$  in the cells  $I_r = [x_{I_r-1/2}, x_{I_r+1/2}]$ , with as usual  $x_{-1/2} = 0$ , and define the discretization points and the mesh step for each  $r$  as:

$$x_{I_r} = \frac{x_{I_r-1/2} + x_{I_r+1/2}}{2}, \quad \Delta x_{I_r} = x_{I_r+1/2} - x_{I_r-1/2},$$

so that the vectorial mesh size is given by:

$$\Delta \vec{x}_I = \prod_{r=1}^d \Delta x_{I_r}.$$

Defining the discrete distribution function on the cell  $I$  at the iteration  $n$ ,  $f_I^n$ , by means of a cell average of  $f$  at time  $t^n$  on the cell  $I$ , the numerical scheme reads

$$f_I^{n+1} = f_I^n + \Delta t^n \left( \frac{1}{2} \sum_{(K,J) \in S^I} \tilde{H}_{J,K} f_K f_J \lambda_{J,K}^I \frac{\Delta \vec{x}_J \Delta \vec{x}_K}{\Delta \vec{x}_I} - \sum_{J=0}^{\infty} H_{I,J} f_I f_J \Delta \vec{x}_J \right), \quad (3.4)$$

where  $\sum_{J=0}^{\infty}$  and  $\sum_{(J,K) \in S^I}$  are now the product of all the sums for  $r = 1, \dots, d$ :

$$\sum_{J=0}^{\infty} = \prod_{r=1}^d \sum_{J_r=0}^{\infty}, \quad \sum_{(K,J) \in S^I} = \prod_{r=1}^d \sum_{(K_r, J_r) \in S^{I_r}},$$

the sets  $S^{I_r}$ , analogously as in the one-dimensional case, are defined by:

$$S^{I_r} = \{(K_r, J_r) \in \mathbb{N} \times \mathbb{N} : (K_r + J_r) \cap I_r \neq \emptyset\},$$

and the first sum in (3.4) represents the sum over all cells  $K_r$  such that the intersection between the cell  $(J_r + K_r)$  and  $I_r$  is not empty.

In order to define  $\lambda_{J,K}^I$ , let us introduce, for  $r = 1, \dots, d$  the following minimum and maximum, generalizing the one-dimensional ones (2.8) and (2.9) and respectively denoting the maximum and minimum bounds of the intersection of the cell  $(J_r + K_r)$  with a given cell  $I_r$ :

$$\overline{m_{J_r, K_r}^{I_r}} = \min(\vec{x}_{I_r+1/2}, \vec{x}_{J_r+1/2} + \vec{x}_{K_r+1/2}),$$

$$\underline{m_{J_r, K_r}^{I_r}} = \max(\vec{x}_{I_r-1/2}, \vec{x}_{J_r-1/2} + \vec{x}_{K_r-1/2}).$$

The proportionality factor  $\lambda_{J,K}^I$  in (3.4), is then defined by

$$\lambda_{J,K}^I = \prod_{r=1}^d \lambda_{J_r, K_r}^{I_r}, \quad \lambda_{J_r, K_r}^{I_r} = \left( \frac{\overline{m_{J_r, K_r}^{I_r}} - \underline{m_{J_r, K_r}^{I_r}}}{\Delta \vec{x}_{J_r} + \Delta \vec{x}_{K_r}} \right)_+, \quad (3.5)$$

which is a direct generalization of (2.10). We remark that, as in the one-dimensional case, for each  $r = 1, \dots, d$ ,  $\lambda_{J_r, K_r}^{I_r}$  is symmetric ( $\lambda_{J_r, K_r}^{I_r} = \lambda_{K_r, J_r}^{I_r}$ ) and that  $0 \leq \lambda_{J_r, K_r}^{I_r} \leq 1$ , the equality sign holding either when the intersection is empty ( $\lambda_{J_r, K_r}^{I_r} = 0$ ) or when  $(J_r + K_r)$  is completely contained in the cell  $I_r$  ( $\lambda_{J_r, K_r}^{I_r} = 1$ ).

Generalizing Proposition 2.3, and defining the set  $R_{J,K}$  as the product of the sets:

$$R_{J_r, K_r} = \{I_r \in \mathbb{N} : I_r \cap J_r + K_r \neq \emptyset\},$$

the modified kernel  $\tilde{H}_{J,K}$  satisfies:

**PROPOSITION 3.1.** *Let  $\Phi_J = \Phi(\vec{x}_J)$ ,  $\Phi_K = \Phi(\vec{x}_K)$  and  $\Phi_I = \Phi(\vec{x}_I)$  respectively be linear combinations of the vectors  $\vec{x}_J$ ,  $\vec{x}_K$  and  $\vec{x}_I$ . Then the discrete first order moment defined by (3.3) is conserved if and only if the modified kernel  $\tilde{H}_{J,K}$  verifies*

$$\tilde{H}_{J,K} = H_{J,K} \frac{(\Phi_J + \Phi_K)}{\sum_{I \in R_{J,K}} \Phi_I \lambda_{J,K}^I}. \quad (3.6)$$

*Proof.* The proof is similar to the one of Proposition 2.3.  $\square$

Note that, in the one-dimensional case there exists only one first order moment, whereas in the multidimensional framework we can define various first order moments. At a continuous level and for  $H(\vec{x}, \vec{x}') = 1$ , each first order moment as well as any their linear combination is conserved. On the other hand, concerning discrete first order moments, since the modified kernel  $\tilde{H}_{J,K}$  depends on the linear combination  $\Phi$  we choose, only one of them is conserved, see also [21] in which for the two dimensional case the authors choose the linear combination  $\Phi(x, y) = x + y$ .

**4. Numerical results.** In order to validate our scheme, we perform some of the numerical tests done in [10] and [16] for the one-dimensional problem and the gelation phenomena, and those in [21], for the two-dimensional case.

Before detailing the considered numerical tests, we recall that the sum in the loss term is defined up to infinity, see (2.6), and when performing numerical simulations we have to truncate it. In [10] there is the choice between a truncated formula for the right-hand side of (1.4) which conserves the first order moment and one which doesn't

conserve it. Here we have to choose the value  $x_{max}$  corresponding to the maximum value for the  $x$  domain. This implies that particles will exit the computational domain. However, we are able to quantify when particles exit the domain and the corresponding mass. Note also that, constructing the numerical scheme for non-uniform meshes permits to consider larger  $x_{max}$  values.

We validate the numerical scheme considering the following uniform (1, 2, 3 and 4) and non-uniform (5, 6, 7 and 8) meshes:

1.  $\Delta x = 0.1, x_{max} = 50,$
2.  $\Delta x = 0.5, x_{max} = 50,$
3.  $\Delta x = 0.01, x_{max} = 50,$
4.  $\Delta x = 0.5, x_{max} = 2500,$
5.  $x_i = x_{max} 2^{(i-N)/3}, x_{max} = 5000, N = 150$  points,
6.  $x_i = x_{max} 2^{(i-N)/3}, x_{max} = 500000, N = 165$  points,
7.  $x_i = x_{max} 2^{(i-N)/6}, x_{max} = 5000, N = 300$  points,
8.  $x_i = x_{max} 2^{(i-N)/1.5}, x_{max} = 5000, N = 75.$

Since we want to save about a thousand values in time for our numerical solutions, at each time iteration we choose the time step as the minimum between the one defined by the CFL condition (2.23) and  $10^{-3}$ . Computations are stopped either when time is equal to  $t = 1.5$  or when 2% of the mass exits the computational domain, for the one-dimensional tests, and at time  $t = 100$  for the two-dimensional ones.

We note that non-uniform meshes are more suitable than uniform ones concerning CPU times aspects. For instance, fixing the maximum size of the domain at  $x_{max} = 100$ , we have performed numerical simulations for both the coalescence kernels  $H = 1$  and  $H = xx'$  and the uniform mesh with  $N = 500$ , that is  $\Delta x = 0.2$ , and the non-uniform mesh with  $N = 250$ , that gives meshes varying from the order  $10^{-5}$  to the order 1. In both cases, we gain one order of magnitude for the CPU time when using the non-uniform mesh with respect to the uniform one. We also note that, in the two dimensional case, our computations are much more costly in terms of CPU times with respect to those given in [21]: the CPU times for tests 5, 6 and 7 are of the order of 100 *min*, whereas in [21] CPU times for the same tests are of the order of 5 *min*. Nevertheless, see section 4.4, our numerical solutions in the two dimensional case are smooth and show a smaller diffusion away from the diagonal  $x = y$  than those traced in [21] (figures 6 and 7).

**4.1. One-dimensional case:**  $H(x, x') = 1$ . We consider the one-dimensional case with the constant kernel  $H(x, x') = 1$  and compare our results with those in [10]. We define the following initial condition

$$f_0(x) = e^{-x}.$$

We recall that in this case, see [10] and references therein, the coalescence equation admits the solution

$$f(x, t) = \left( \frac{2}{2+t} \right)^2 \exp \left( \frac{-2x}{2+t} \right), \quad (4.1)$$

with zero and first moments

$$\mathcal{M}_0(t) = \frac{2}{2+t}, \quad \mathcal{M}_1(t) = 1.$$

In figure 4.1 we trace the zero (left) and first (right) order moments against the analytical solution (continuous line). While for the zero order moment we can barely

see the difference between the various mesh choices, concerning the first order moment, we note that each mesh choice is constant in time, but the starting value is not the good one when applying non-uniform meshes. This error is due to the quadrature formula we apply to compute the initial first order moment. Note that the numerical result of test 4 coincides with the result of test 2. Moreover, the better fit seems to be given by test number 3, that is the uniform mesh with the smallest space step. Nevertheless, we note that non-uniform meshes allow us to compute solutions on bigger domains, i.e. for larger values of  $x_{max}$ , than uniform ones, with still an acceptable error.

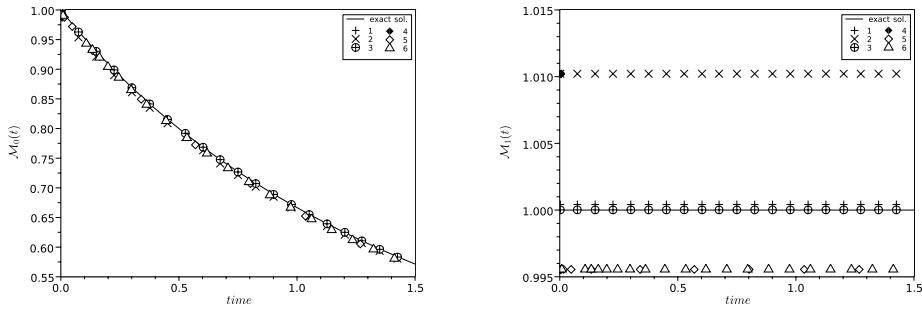


FIG. 4.1.  $H(x, x') = 1$ : zero (left) and first (right) order moments.

We also compute, as done in [10], the discrete  $L_1$  error defined by:

$$\epsilon = \sum_{i=0}^N \Delta x_i |f(x_i, t_n) - f_i^n|, \tag{4.2}$$

for the uniform meshes with  $N = 125, 250, 500$  and truncation  $x_{max} = 50$ , as well as for the non-uniform tests 5, 7 and 8. The results in normal (left) and logarithmic (right) scale are traced in figures 4.2 and 4.3 and they numerically show that the scheme is second order accurate, as it was expected since all integrals in our discretization are approximated by midpoint quadrature formulas.

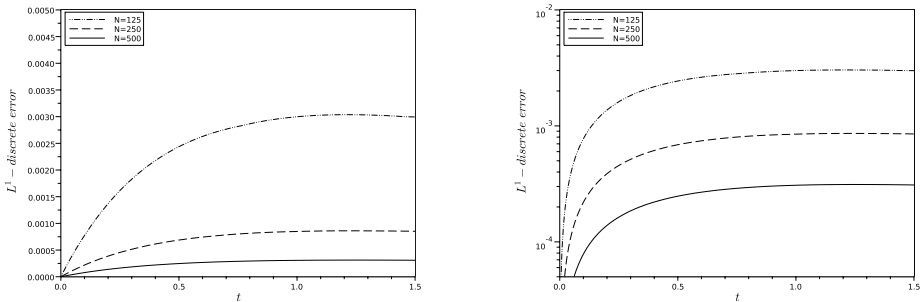


FIG. 4.2.  $H(x, x') = 1$ : uniform mesh,  $\epsilon$  in normal (left) and logarithmic (right) scale.

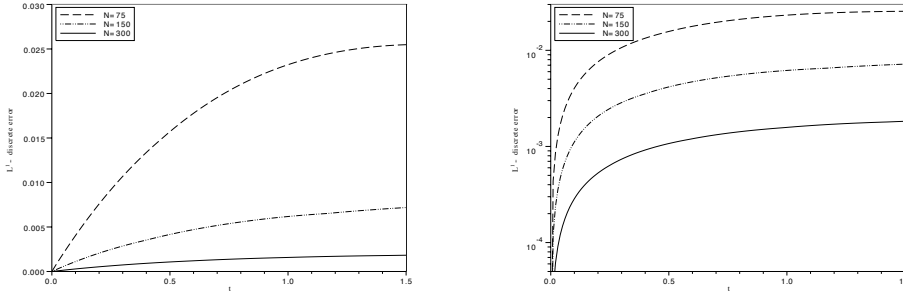


FIG. 4.3.  $H(x, x') = 1$ : non-uniform mesh,  $\epsilon$  in normal (left) and logarithmic (right) scale.

**4.2. One-dimensional case:**  $H(x, x') = x + x'$ . We now consider one of the one-dimensional tests performed in [21]. We define the initial condition

$$f_0(x) = e^{-x},$$

defining  $\alpha = 1 - e^{-t}$ , the analytical solution reads

$$f(x, t) = \frac{1 - \alpha}{x\sqrt{\alpha}} I_1(2x\sqrt{\alpha}) e^{-x(1+\alpha)}, \text{ with,}$$

with  $I_1$  the modified Bessel function at order one

$$I_1(z) = \frac{1}{\pi} \int_0^\pi \exp(z \cos(\theta)) \cos(\theta) d\theta. \quad (4.3)$$

The moments of order zero, one, two and three are respectively given by

$$\mathcal{M}_0(t) = e^{-t}, \quad \mathcal{M}_1(t) = 1, \quad \mathcal{M}_2(t) = 2e^{2t}, \quad \mathcal{M}_3(t) = 6e^{3t} (2e^t - 1).$$

We compute the numerical solution applying the non-uniform mesh of test 5, but with  $N = 40$  and  $x_{max} = 2000$ , as done in [21]. We stop the computation once the first order moment begin to decrease, because part of the density has reached the boundary of the truncated domain. The results are shown in figure 4.4 where we have traced, in logarithmic scale, the analytical and discrete moments up to the third order. Figure 4.4 shows that the third order moment is always increasing, as it is expected by theoretical results. The overestimation of the discrete moments of order bigger than one may be explained by the quadrature formula applied in order to compute the integrals. Compared with the numerical results published in [21], the third order moment we compute are totally different. In particular, the one traced in [21] is non-monotone, and we don't have any way of comparison since the authors don't give any formula for the third order moment.

**4.3. One-dimensional case:**  $H(x, x') = xx'$ . Consider the one-dimensional case with multiplicative kernel  $H(x, x') = xx'$  which is well known to yield to gelation. Following [10], we choose the initial condition

$$f_0(x) = \frac{e^{-x}}{x},$$



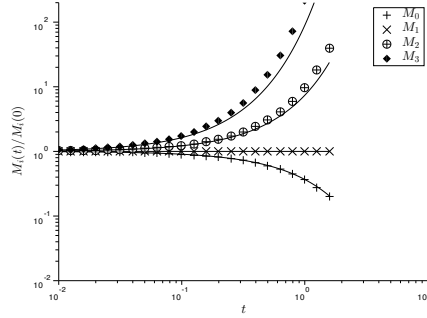


FIG. 4.4.  $H(x, x') = x + x'$ : zero, first, second and third order moments.

for which the coalescence problem admits the solution

$$f(x, t) = e^{-Tx} \frac{I_1(2x\sqrt{t})}{x^2\sqrt{t}}, \quad (4.4)$$

with the time  $T$  given by:

$$T = \begin{cases} 1 + t & \text{if } t \leq 1, \\ 2\sqrt{t} & \text{otherwise,} \end{cases} \quad (4.5)$$

and  $I_1$  the modified Bessel function at order one defined by (4.3).

The first order moment can be computed:

$$\mathcal{M}_1(t) = \begin{cases} 1 & \text{if } t < 1, \\ t^{-1/2} & \text{otherwise.} \end{cases}$$

Note that for  $t > 1$  the first order moment is decreasing, the gelation time being  $t = 1$ .

In figure 4.5, we compare the normalized first order moment obtained by our numerical simulation applying the uniform meshes 1, 2 and 3 (left) or the non-uniform meshes 5 and 7 (right) to the analytical solution truncated respectively at  $x_{max} = 50$  (dotted line, left graph) or  $x_{max} = 5000$  (dotted line, right graph). We also plot the exact analytical solution (continuous line, both graphs). The numerical results given by 1, 2 and 3 well fit the truncated solution (dotted line), but we observe a little overestimation once the gelation phenomena takes place. This may be due to the very small truncation value we choose. Concerning the non-uniform meshes, we note that, as expected, the numerical result for the finest mesh (test 7) has a better fit with the truncated solution (dotted line). Note that the truncation value  $x_{max}$  for test 7 is larger than for the uniform tests: non-uniform meshes allow to compute solutions with a larger truncation value  $x_{max}$  and give more accurate results.

Moreover, as done for the constant kernel, we compute the discrete  $L_1$  error defined by (4.2) for the uniform mesh with  $N = 125, 250, 500$  and truncation  $x_{max} = 50$ . The result is shown in figure 4.6 both in normal (left) and logarithmic (right) scale. As expected the scheme is second order accurate. Note that, since the truncation value is small and that the kernel is multiplicatif, particles rapidly exit the computational domain, yielding to small oscillations in the computation of the  $L_1$  error, which are more visible for the test  $N = 500$ . The  $L_1$  error for non-uniform meshes can't be

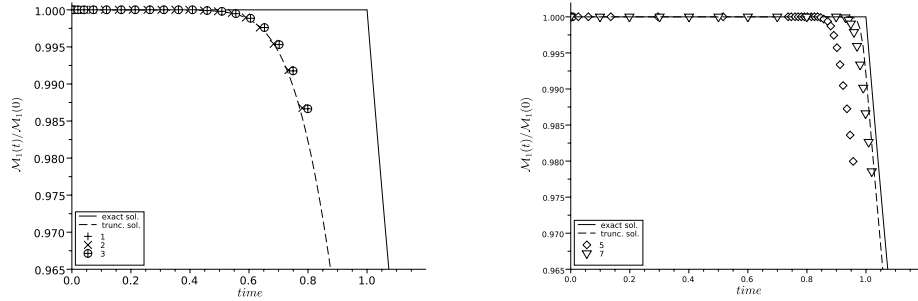


FIG. 4.5.  $H(x, x') = xx'$ : uniform meshes 1, 2, 3 (left) and non-uniform meshes 5 and 7 (right).

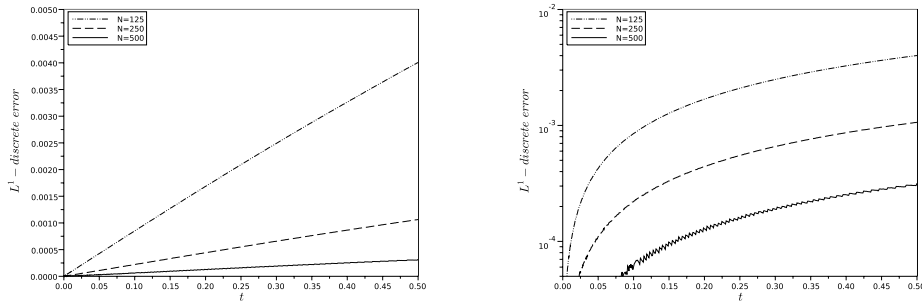


FIG. 4.6.  $H(x, x') = xx'$ :  $\epsilon$  in normal (left) and logarithmic (right) scale.

computed since the values of the analytical solution are too close to zero and its computation on the smaller cells is impossible.

We finally remark that when computing moments of order bigger than 1, we obtain the same behavior as in [10], that is, as time approach the gelation time, the moments of order  $p \geq 3/2$  blow up.

**4.4. Two dimensional validation.** In this section, we compare our numerical results with those obtained in [21] for the two-dimensional case. Before entering the details, let us write the general formula for a moment of order  $p = r + s$ , in two dimensions

$$\mathcal{M}_{r,s}(t) = \int_0^\infty \int_0^\infty x^r y^s f(t, x, y) dx dy.$$

Following [21], we assume a constant coalescence kernel  $H(\vec{x}, \vec{x}') = 1$  and we compute the solution up to a final time  $t = 100$ . In both problems we shall consider, we choose the test 5 where we have fixed the number of discretization points to  $N = N_x = N_y = 40$ , and choose a truncation value  $x_{max} = y_{max} = 1000$ .

First, we consider the initial condition:

$$f_0(x, y) = e^{-x-y}.$$

The analytical solution reads, see also [20],

$$f(t, x, y) = \frac{4 e^{-x-y}}{(t+2)^2} I_0(\theta), \quad \text{with} \quad \theta = \sqrt{\frac{4t}{t+2}}$$

and  $I_0$  the modified Bessel function of first kind of order zero

$$I_0(z) = \frac{1}{\pi} \int_0^\pi \exp(z \cos(\theta)) d\theta. \quad (4.6)$$

The zero, first and second order moments associated with this solution are, see [21],

$$\mathcal{M}_{0,0}(t) = \frac{2}{2+t}, \quad \mathcal{M}_{1,0}(t) = \mathcal{M}_{0,1}(t) = 1, \quad \mathcal{M}_{1,1}(t) = 1+t.$$

In figure 4.7 we trace in logarithmic scale the analytical solution.

Figure 4.8 shows the numerical results. Note that the zero, first and second order

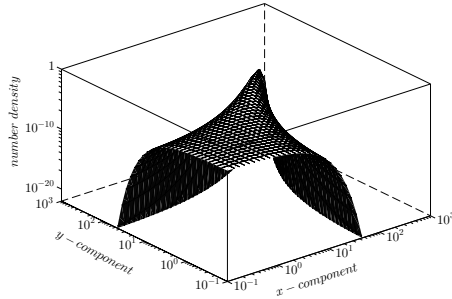


FIG. 4.7.  $f_0(x, y) = e^{-x-y}$ : analytical solution at  $t = 100$

numerical moments well fit the analytical ones (continuous lines, right graph). Moreover (left graph), the computed solution is smooth, but has a diffusion with respect to the analytical one in particular around the diagonal  $x = y$ . This diffusion diminishes when considering the extrema of the computational domain. We note that the numerical results (figure 6) in [21], have a good fit for the moments too, and that the computed solution shows singularities and has a larger diffusion away from the diagonal  $x = y$ .

We now consider a final test also performed in [21]. The initial condition reads

$$f_0(x, y) = 16xy e^{-2(x+y)}.$$

In this case the analytical solution is given by, see [13],

$$f(t, x, y) = \frac{8}{\sqrt{t(t+2)^3}} e^{-2(x+y)} [I_0(\theta) - J_0(\theta)],$$

where

$$\theta = 4\sqrt{xy} \left( \frac{t}{t+2} \right)^{1/4}.$$

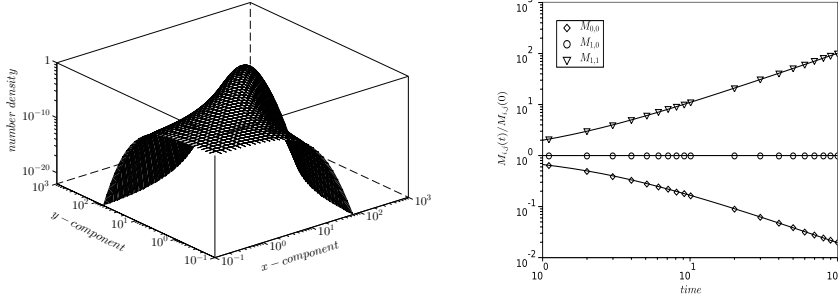


FIG. 4.8.  $f_0(x, y) = e^{-x-y}$ : Computed solution (left). Zero, first and second order moments (right).

and  $J_0$  and  $I_0$  respectively are, the Bessel function

$$J_0(z) = \frac{1}{\pi} \int_0^\pi \cos(z \sin(\theta)) d\theta$$

and the modified Bessel function of first kind of order zero, defined by (4.6). We consider the same moments as in [21] (see also [26]):

$$\mathcal{M}_{0,0}(t) = \frac{2}{2+t},$$

$$\mathcal{M}_{1,0}(t) = \mathcal{M}_{0,1}(t) = 1,$$

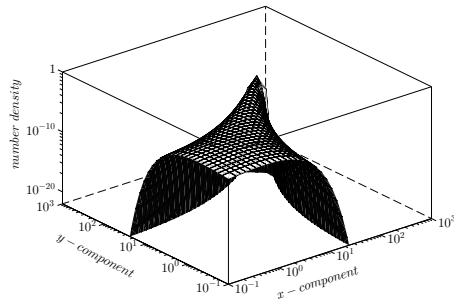
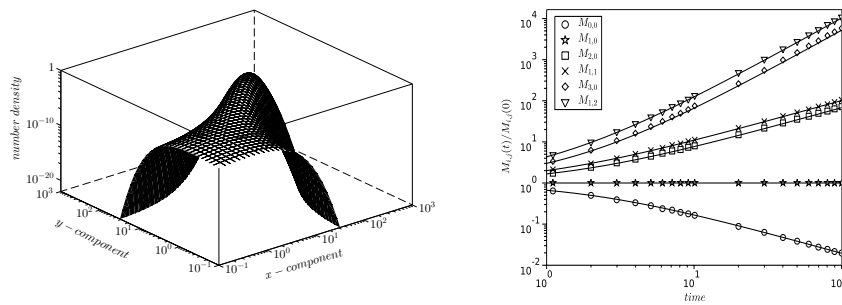
$$\mathcal{M}_{2,0}(t) = \frac{3+2t}{2}, \quad \mathcal{M}_{1,1}(t) = 1+t,$$

$$\mathcal{M}_{3,0}(t) = \frac{3(1+t)(2+t)}{2}, \quad \mathcal{M}_{2,1}(t) = \frac{(3+7t+3t^2)}{2}.$$

The analytical solution at  $t = 100$  is traced in logarithmic scale in figure 4.9.

In figure 4.10 we trace the computed moments and the analytical ones (right graph). Concerning the moments, we note a better agreement than the one obtained in [21] (figure 7). Moreover (left), diffusion of the numerical solution away from the diagonal  $x = y$  is smaller than the one showed in [21], and we still have more diffused along the diagonal. Again, the two-dimensional solution in [21] shows singularities.

**5. Conclusion.** In this paper we have described a finite volume discretization of the coalescence equation (1.1) which conserves the first order moment, at least until the gelation time is reached. The proposed numerical scheme is suitable both for uniform and non-uniform meshes, and deal with the overlaps of created cells proportionally distributing it over the defined meshes. The scheme is second order accurate and consistent with the continuous formulation of the coalescence problem. We emphasize the use of non-uniform meshes for CPU time aspects as well as for a better description of distribution functions with dimensions varying over several order of magnitude, as it will be the case in our volcanology problem. The generalization

FIG. 4.9.  $f_0(x, y) = 16xy e^{-2(x+y)}$ : analytical solution at  $t = 100$ .FIG. 4.10.  $f_0(x, y) = 16xy e^{-2(x+y)}$ : Computed solution (left). Computed moments (right).

to a multidimensional framework has been detailed, too.

The code has been validate considering the following classical kernels:  $H(x, x') = 1, x + x', xx'$ , for which analytical solutions to (1.1) as well as their moments have been recalled. The numerical results are compared with those already published, in particular with those of [10] and [21]. Our numerical scheme shows a good agreement with all the considered tests, and seems to behave better than the one in [21], at least concerning the smoothness of the numerical solution and the computation of higher order moments. Still some work may be done in order to reduce the observed numerical diffusion, for example, modifying the way the created cells are distributed over the existing mesh.

#### REFERENCES

- [1] H. Babovsky, On a Monte Carlo scheme for Smoluchowski's coagulation equation, Monte Carlo Methods Appl., 5(1999), pp.1-18.
- [2] A. Burgisser and J.E. Gardner, Experimental constraints on degassing and permeability in volcanic conduit flow, Bull. Volcanol., 67(2005), pp. 42-56.
- [3] M. Deaconu, N. Fournier, E. Tanré, Study of a stochastic particle system associated with the Smoluchowski coagulation equation, Methodol. Comput. Appl. Probab., 5(2003), pp. 131-158.
- [4] M. Duomic, A. Marciniak-Czochra, B. Perthame, J.P. Zubelli, A structured population model of cell differentiation, SIAM J. Appl. Math., 71 (6) (2011), p. 1918
- [5] M. Doumic, B. Perthame, J. Zubelli, Numerical Solution of an Inverse Problem in Size-

- Structured Population Dynamics, Inverse Problems, 25(2009).
- [6] M. Duomic, L. Tané, Estimating the Division Rate for the Growth-Fragmentation Equation, *J. Math. Biol.* (2012), to appear.
  - [7] L. D. Erasmus, D. Eyre, R. C. Everson, Numerical treatment of the population balance equation using a Spline-Galerkin method, *Comp. Chem. Eng.*, 8(1994), pp. 775-783.
  - [8] A. Eibeck, W. Wagner, An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena, *SIAM J. Sci. Comput.*, 22(2000), pp. 802-821.
  - [9] A. Eibeck, W. Wagner, Stochastic particle approximations for Smoluchowski's coagulation equation, *Ann. Appl. Probab.*, 11(2001), pp. 1137-1165.
  - [10] F. Filbet, P. Laurençot, Numerical simulation of the Smoluchowski equation, *SIAM J. Sci. Comput.*, 25(2004), pp. 2004-2028.
  - [11] L. Forestier-Coste, Croissance et coalescence de bulles dans les magmas : analyse mathématique et simulation numérique, PhD. Thesis, Université d'Orléans, (2012).
  - [12] L. Forestier-Coste, S. Mancini, A. Burgisser, F. James, Numerical resolution of a mono-disperse model of bubble growth in magmas, *Appl. Math. Mod.*, 36(2012), pp. 5936-5951.
  - [13] F. Gelbard, J.H. Seinfeld, Coagulation and growth of a multicomponent aerosol, *J. Coll. and Inter. Sci.*, 63(1978), pp.357-375.
  - [14] D. S. Krivitsky, Numerical solution of the Smoluchowski kinetic equation and asymptotics of the distribution function, *J. Phys. A*, 28(1995), pp. 2025-2039.
  - [15] R. Kumar, J. Kumar, G. Warnecke, Numerical methods for solving two-dimensional aggregation population balance equations, *Chem. Eng. Sci.*, 35 (2011), pp. 999-1009.
  - [16] J. Kumar, M. Peglow, G. Warnecke, S. Heinrich, An efficient numerical technique for solving population balance equation involving aggregation, breakage, growth and nucleation, *Powder Technology*, 182(2008), pp. 81-104.
  - [17] M. H. Lee, On the validity of the coagulation equation and the nature of runaway growth, *Icarus*, 143(2000), pp. 74-86.
  - [18] M. H. Lee, A survey of numerical solutions to the coagulation equation, *J. Phys. A*, 34(2001), pp. 10219-10241.
  - [19] F. Leyvraz, H. R. Tschudi, Singularities in the kinetics of coagulation processes, *J. Phys. A*, 14(1981), pp. 3389-3405.
  - [20] A.A. Lushnikov, Evolution of coagulating systems III. Coagulating mixtures, *J. Coll. and Inter. Sci.*, 54(1976), pp. 94-101.
  - [21] S. Qamar, G. Warnecke, Solving population balance equations for two-component aggregation by a finite-volume scheme, *Chem. Eng. Sci.*, 62(2007), pp. 679-693.
  - [22] S. Qamar and G. Warnecke, Numerical solution of population balance equations for nucleation growth and aggregation processes, *Comp. & Chem. Eng.*, 31(2007), pp.1576-1589.
  - [23] S. Qamar, G. Warnecke and M.P. Elsner, On the solution of population balances for nucleation, growth, aggregation and breakage processes, *Chem. Eng. Sci.*, 64(2009), pp. 2088-2095.
  - [24] M. Smoluchowski, Drei Vorträge über Diffusion, Brownsche Molekularbewegung und Koagulation von Kolloidteilchen, *Physik. Zeitschr.*, 17(1916), pp. 557-599.
  - [25] H. Tanaka, S. Inaba, K. Nakaza, Steady-state size distribution for the self-similar collision cascade, *Icarus*, 123(1996), pp. 450-455.
  - [26] H.M. Vale, T.F. Mckenna, Solution of the population balance equation for two-component aggregation by an extended fixed pivot technique, *Ind. & Eng. Chem. Res.*, 44(2005), pp. 7885-7891.
  - [27] A. Wiles, Modular elliptic curves and Fermat's last theorem, *Ann. Math.*,141(1995), pp.443-551.