Grain growth for astrophysics with discontinuous Galerkin schemes

Maxime Lombart, Guillaume Laibe

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **501,** 4298–4316 (2021) doi:10.1093/mnras/staa3682
Advance Access publication 2020 November 26

Grain growth for astrophysics with discontinuous Galerkin schemes

Maxime Lombart [1]★ and Guillaume Laibe[1,2]★

[1]*Centre de Recherche Astrophysique de Lyon UMR5574, Univ Lyon, Univ Lyon1, Ens de Lyon, CNRS, F-69230 Saint-Genis,-Laval, France*
[2]*Institut Universitaire de France, France*

Accepted 2020 November 23. Received 2020 November 19; in original form 2020 October 20

ABSTRACT

Depending on their sizes, dust grains store more or less charges, catalyse more or less chemical reactions, intercept more or less photons and stick more or less efficiently to form embryos of planets. Hence, the need for an accurate treatment of dust coagulation and fragmentation in numerical modelling. However, existing algorithms for solving the coagulation equation are overdiffusive in the conditions of 3D simulations. We address this challenge by developing a high-order solver based on the discontinuous Galerkin method. This algorithm conserves mass to machine precision and allows to compute accurately the growth of dust grains over several orders of magnitude in size with a very limited number of dust bins.

Key words: methods: numerical – dust, extinction.

1 INTRODUCTION

Solid particles pervade the interstellar medium at all scales. Although they represent a small amount of its total mass, they deeply influence its evolution by setting the local chemical, thermal, and charge balances. Dust plays also a key role in the formation of planets, since solid bodies grow over 30 orders of magnitude in mass to form cores of planets. Spatially resolved observations of young stellar objects strongly suggest that at least some planets have to form in less that one million of years (e.g. ALMA Partnership et al. 2015; Avenhaus et al. 2018; Pinte et al. 2020). Key is to understand how dust growth can be so efficient. However, planet formation is an out-of-equilibrium non-linear multiscales and multiphysics process. For example, dust grains differentiate from the gas as they settle vertically and drift radially in the disc (i.e Testi et al. 2014, and references therein). This creates instabilities which concentrate the solids even more, affecting the collisional rate of the grains, and thus, their growth or fragmentation. Since dust dynamics strongly depends on the grain size, growth operates a strong feed-back on the spatial distribution of the particles.

Hence, 3D dust/gas simulations that include growth and fragmentation are compulsory to understand dust evolution during the early stages of planet formation (e.g. Safronov 1972; Hayashi & Nakagawa 1975; Weidenschilling 1980; Ohtsuki, Nakagawa & Nakazawa 1990; Wetherill 1990; Tanaka, Inaba & Nakazawa 1996; Dominik et al. 2007; Ormel, Spaans & Tielens 2007; Birnstiel, Dullemond & Brauer 2010). The simplest way to formalize the evolution of a local mass distribution of dust grains is by the mean of the deterministic mean-field Smoluchowski equation, which assumes binary collisions (Smoluchowski 1916). This equation does not have generic analytic solutions. *Integrated non-linearities challenge numerical solvers to obtain accurate solutions* (see Fig. 1). As such, this equation has been thoroughly studied since a century (e.g. Müller 1928; Schumann 1940; Chandrasekhar 1943; Melzak 1953; McLeod 1962a; Golovin 1963; Berry 1967; Scott 1968; Trubnikov 1971; Hidy & Brock 1972; Drake 1972; Gillespie 1975b; Silk & White 1978; Silk & Takahashi 1979; Gelbard, Tambour & Seinfeld 1980; Aldous 1999; Friedlander et al. 2000; Ramkrishna 2000; Filbet & Laurencot 2004; Jacobson 2005; Pruppacher & Klett 2010), and applied extensively to several fields such aerosols science, chemistry, meteorology, biology, and astrophysics.

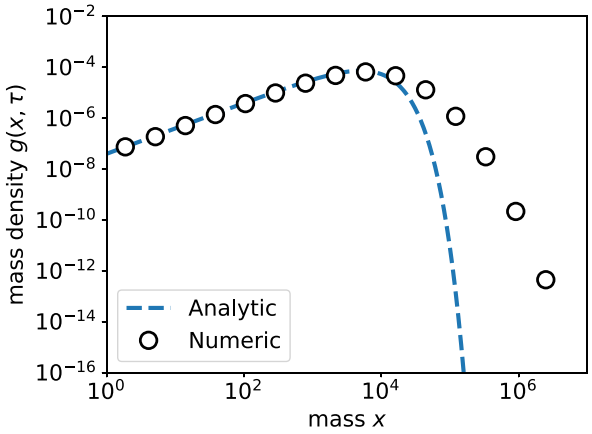

Figure 1 An illustration of the growth overdiffusion problem: numerical schemes of order 0 overestimate the formation of large grains at low resolution. The plot has been realized with the scheme presented in Kovetz & Olund (1969) for the case of a constant kernel $K = 1$ with $N = 15$ logarithmically spaced dust bins.

It has been shown that classical solvers require a sufficient resolution in mass to avoid artificial formation of aggregates of large masses (Soong 1974; Berry & Reinhardt 1974; Trautmann & Wanner 1999; Khain & Pinsky 2018). This artificial diffusion may become particularly important when the mass interval considered is large (Fig. 1). Typically, for planet formation, a few hundreds of mass bins are required to compute dust growth from interstellar

★ E-mail: maxime.lombart@ens-lyon.fr (ML); guillaume.laibe@ens-lyon.fr (GL)

sizes to pebbles. Usually, this fact is of no importance given current computational capacities. However, 3D hydrodynamical simulations can hardly handle more than (a few) ten(s) of mass bins in practice. Compromises have therefore been performed either by simplifying their growth or their dynamics. However, 1D–2 D hydrodynamical codes integrating the Smoluchowski equation (e.g. Birnstiel et al. 2010) provide different results compared to 3D hydrodynamical codes with monodisperse growth models (e.g. Gonzalez, Laibe & Maddison 2017), showing the necessity of a comprehensive approach. This implies to develop a solver which solves accurately the Smoluchowski equation with a limited number of bins, tractable by 3D hydrodynamical codes.

Reaching high accuracy with a low number of bins while conserving mass of a finite interval of mass is a characteristic property of finite volume high-order solvers, which stem therefore as a natural way to address the growth overdiffusion problem. In this study, we present a high-order solver for the Smoluchowski equation based on the Discontinous Galerkin method, following the pioneering work of Liu, Gröpler & Warnecke (2019). Important properties of the Smoluchowski equation discussed in the astrophysical context are presented in Section 2. The novel Discontinous Galerkin numerical scheme is presented in Section 3. The performances of the solver regarding the overdiffusion problem are studied in Section 4. Applicability of the algorithm to young stellar objects or in other astrophysical contexts are discussed in Section 5.

2 SMOLUCHOWSKI EQUATION

2.1 Short summary

The Smoluchowski equation describes mass conservation for a distribution aggregates where mass transfers are allowed. This equation exists under a discrete form (monomers forming polymers) or a continuous limit form when mass quantization becomes negligible (Müller 1928). The Smoluchowski equation is a non-linear integro-differential hyperbolic equation that depend on a collision function called the growth kernel (or kernel) which quantifies the collision rate between two grains. Explicit solutions exist only for the so-called constant (Smoluchowski 1916; Schumann 1940; Scott 1968), additive (Golovin 1963; Scott 1968) and multiplicative kernels (McLeod 1962a; Scott 1968), implying numerical resolution for physical problems. Among the known solutions, self-similar solutions are particularly important since they provide asymptotic behaviour of the mass distribution at large times (Schumann 1940; Friedlander & Wang 1966; Wang 1966; Menon & Pego 2004; Niethammer, Throm & Velázquez 2016; Laurençot 2018). A generic feature of these solutions is the exponentially fast decay of the solution at large masses. Gelation, that is, formation of aggregates of infinite mass form in a finite time for kernels sustaining explosive growth (Leyvraz & Tschudi 1981). In astrophysics, collisions occurs essentially through ballistic impacts modulated by focusing due to long-range interactions (Safronov 1972; Dullemond & Dominik 2005). Kernels are non-explosive and mass remains rigorously conserved during the grow process.

2.2 Conservative form

Mass conservation for a distribution of growing grains has been originally formalized by Smoluchowski (1916). Growth is modelled via binary collisions between spheres having known mean probabilities. The by-products of collisions are called aggregates or polymers. In Smoluchowski (1916), aggregates are assumed to also have spherical

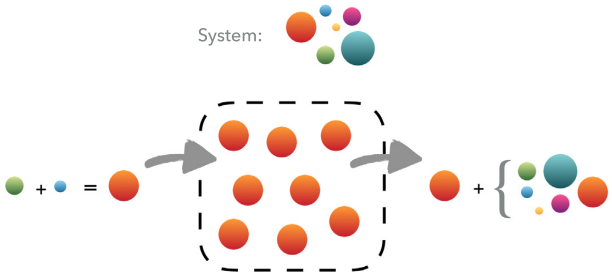

Figure 2. Illustration of the Smoluchowski equation equation (1). Polymers of mass m_i are represented in orange. The green and blue polymers have masses lower than m_i. Creation (respectively, growth) of polymers of mass m_i increases (respectively, decreases) its number density.

shapes. Spatial correlations are neglected. The smallest colliding elements are referred as monomers. For physical systems involving aggregates made of large numbers of monomers, it is often convenient to assume continuous mass distributions. The population density of grains within an elementary mass range dm is characterized by its number density $n(m)$. The continuous Smoluchowski equation is given by

$$\frac{\partial n(m,t)}{\partial t} = \frac{1}{2}\int_0^m K(m-m',m')n(m-m',t)n(m',t)\mathrm{d}m' - n(m,t)\int_0^\infty K(m,m')n(m',t)\mathrm{d}m', \quad (1)$$

where t denotes time and m and m' the *masses* of two colliding polymers. The averaged probabilities of collision are encoded inside the coagulation kernel $K(m, m')$, which is a symmetric function of m and m' for binary collisions (see Section 2.3). Fig. 2 shows the physical meaning of the non-linear integrodifferential equation equation (1). The number of grains encompassed within a given interval of masses varies since (i) binary collisions of aggregates of appropriate masses can increase this population (first term of the right-hand side of equation 1), but (ii) those grains may themselves collide with other grains to form larger aggregates (second term of the right-hand side of equation 1). This equation can be put under a convenient dimensionless form by introducing (Scott 1968; Drake 1972)

$$\begin{cases} x \equiv m/m_0,\ y \equiv m'/m_0,\ \mathcal{K}(x,y) = K(m,m')/K_0, \\ \tau = (K_0 N_0)t,\ f(x,\tau) = m_0\, n(m,t)/N_0. \end{cases} \quad (2)$$

N_0 is the initial total number density of particles, m_0 is the initial mean mass of the particles, and K_0 is a normalizing constant with dimensions [length]^3/time. We adopt the variables x and τ for sake of clarity and homogeneity with the existing literature (e.g. Friedlander et al. 2000; Jacobson 2005, and references therein). x denotes therefore masses. Equation (1) transforms into

$$\frac{\partial f(x,\tau)}{\partial \tau} = \frac{1}{2}\int_0^x \mathcal{K}(y,x-y)f(y,\tau)f(x-y,\tau)\mathrm{d}y - f(x,\tau)\int_0^\infty \mathcal{K}(y,x)f(y,\tau)\mathrm{d}y. \quad (3)$$

Equation (3) is physically ill-posed, since the probability to form aggregates of mass larger than the initial mass of the system may be non-zero. Recently, Tanaka et al. (1996) have shown that equation (3) can be equivalently written under the conservative form

$$\begin{cases} \frac{\partial g(x,\tau)}{\partial \tau} + \frac{\partial F_{\mathrm{coag}}[g](x,\tau)}{\partial x} = 0 \\ F_{\mathrm{coag}}[g](x,\tau) = \int_0^x \int_{x-u}^\infty \mathcal{K}(u,v)g(u,\tau)\frac{g(v,\tau)}{v}\mathrm{d}u\mathrm{d}v, \end{cases} \quad (4)$$

Table 1. Functional form of the different coagulation kernels $\mathcal{K}$ considered in this study.

Kernel	$\mathcal{K}(x, y)$
Size-independent	1
Sum	$x + y$
Product	xy
Ballistic	$\pi(x^{1/3} + y^{1/3})^2 \Delta v$

where $g(x, \tau) \equiv x f(x, \tau)$ is the mass density of polymers per unit mass, and $F_{\rm coag}[g](x, \tau)$ is the flux of mass density across the mass x triggered by coagulation (Filbet & Laurencot 2004). Under this conservative form, the infinite upper bound of the second integral in $F_{\rm coag}$ can simply be replaced by $x_{\rm max} - u$. This prevents the formation of aggregates of masses larger than $x_{\rm max}$ by settling the passing-through mass flux to be rigorously zero.

2.3 Kernels

Physically, the coagulation kernel is defined according to

$$K(m, m') \equiv \beta(m, m', \Delta v)\Delta v(m, m')\sigma(m, m'), \tag{5}$$

where Δv is the mean relative velocity between two aggregates of masses m and m', σ is the mean effective cross-section of collision, and β denotes the mean sticking probability of the grains. The coagulation kernel encodes the microphysics of collisions inside β, σ, and Δv, those parameters depending *a priori* on the sizes of the colliding grains, or the kinetic and thermodynamical parameters of an eventual surrounding flow. A kernel of particular importance for physical problems is the Ballistic kernel (Table 1). In this case, σ corresponds simply to the geometric cross-section of the grains (focusing effects due to electrostatic or gravitational forces being neglected), and β and Δv are treated as constants (which may be a relevant approximation at least over moderate ranges of masses). Coagulation kernel can also be seen as mathematical objects useful to study the properties of the Smoluchowski equation under various conditions or to derive explicit analytic solutions. The expression of the four kernels discussed in this work is given in Table 1.

2.4 Analytic solutions

Explicit analytic solutions exist in the case of simple kernels and specific initial conditions. We review these solutions hereafter since they will be used in Section 4 to benchmark the numerical algorithms.

2.4.1 Constant kernel

For the constant kernel $\mathcal{K}(x, y) = 1$ and the initial condition $f(x, 0) = \exp(-x)$, the solution of equation (3) is (Müller 1928; Schumann 1940; Melzak 1957; Rajagopal 1959; Scott 1968; Silk & Takahashi 1979)

$$\begin{cases} f_1(\tau) \equiv \frac{4}{(2+\tau)^2}, \ f_2(\tau) \equiv \frac{\tau}{2+\tau}, \\ f(x, \tau) = f_1(\tau) \exp(-\{1 - f_2(\tau)\}x). \end{cases} \tag{6}$$

Physically, a constant kernel $\mathcal{K} = 1$ implies that the frequency of collisions between two aggregates is independent of their size.

2.4.2 Additive kernel

The solution for the additive kernel $\mathcal{K}(x, y) = x + y$ with the initial condition $f_0(x, 0) = \exp(-x)$ has been derived by Golovin (1963). Scott (1968) extended the derivation for a general initial condition. For an initial condition under the form $f(x, 0) = \exp(-x)$, the solution of equation (3) is

$$\begin{cases} T \equiv 1 - \exp(-\tau), \\ f(x, \tau) = \frac{(1-T)\exp(-x\{1+T\})}{xT^{1/2}} I_1(2xT^{1/2}), \end{cases} \tag{7}$$

where I_1 is the modified Bessel function of first kind. Physically, the additive kernel implies that the frequency of collisions increases according to the size of the grains. Large aggregates form faster compared to case of a constant kernel, leading to broader dust distributions at large masses. The asymptotic tail presents therefore a smoother decay compared to the case $\mathcal{K} = 1$.

2.4.3 Multiplicative kernel

Originally, McLeod (1962b) derived a solution for the multiplicative kernel $\mathcal{K}(x, y) = xy$ with the initial condition $f_0(x, 0) = x^{-1}\exp(-x)$ only for a small interval of time. The general solution for this problem was later found by Ernst, Ziff & Hendriks (1984)

$$\begin{cases} T \equiv \begin{cases} 1 + \tau & \text{if} \quad \tau \leq 1 \\ 2\tau^{1/2} & \text{otherwise} \end{cases}, \\ f(x, \tau) = \frac{\exp(-Tx) I_1(2x\tau^{1/2})}{x^2 \tau^{1/2}}. \end{cases} \tag{8}$$

The multiplicative kernel is a typical kernel to study the occurrence of gelation, since at $\tau = 1$, aggregates with infinite masses form and mass conservation is mathematically no longer satisfied. Physically, the multiplicative kernel means an explosive increase of the collisional frequencies with respect to grain sizes. Massive grains form faster compared to the case of the additive kernel. In the same time, the mass density of small grains decreases quickly.

2.5 Numerical methods

No known analytic solutions exist for the Smoluchowski coagulation equation with physical kernels, implying numerical resolution. Various numerical schemes have been developed for this purpose. Two classes of algorithms have been developed. A first class of solvers consists of Monte Carlo simulations (e.g. Gillespie 1975a; Liffman 1992; Smith & Matsoukas 1998; Lee & Matsoukas 2000; Debry, Sportisse & Jourdain 2003; Sheng & Shen 2006; Ormel et al. 2007; Zsom & Dullemond 2008). Although convenient, these methods have two principal drawbacks. First, a large number of particles is required to ensure appropriate accuracy of the number density distribution f. Secondly, the scheme is not deterministic and simulations can be reproduced only in a statistical sense, which is not satisfying when interfacing with hydrodynamics. A second class of solvers consist of deterministic algorithms. These methods have been summarized in Kostoglou & Karabelas (1994), Kumar & Ramkrishna (1996), Ramkrishna (2000), Pruppacher & Klett (2010), and Khain & Pinsky (2018). A short but comprehensive summary is given hereafter.

2.5.1 Method of moments

The method of moments seems to be the first numerical method proposed to solve the Smoluchowski equation (Hulburt & Katz 1964). A system of ordinary differential equations (ODEs) is written over the kth moments $M_k \equiv \int_0^\infty x^k f(x, \tau) \mathrm{d}x$ of the number density function. Approximations either for the reconstruction of f (Hulburt & Katz 1964) or for the derivation of fractional moments (Estrada & Cuzzi 2008) are then required to close this system of ODEs. The standard moment method requires an analytical integration of

the kernel. To avoid this difficulty, quadrature moment methods, where integrals are approximated by Gaussian quadrature methods, have been developed. Solutions of moments can be used directly to derive the total number of particles M_0, the total mass M_1 or other physical quantity such as dust opacities (Marchisio, Vigil & Fox 2003; Estrada & Cuzzi 2008). Number densities f are reconstructed using polynomials (Pruppacher & Klett 1980; Piskunov & Golubev 2002).

2.5.2 *Point-based methods*

The number density function f is sampled over a mass grid. The main difficulty lies in representing the continuous distribution f as accurately as possible using the values of f at the sampling points. Different algorithms have been developed using this approach:

2.5.3 *Interpolation method*

This method was developed by Berry (1967) and Berry & Reinhardt (1974). The continuous Smoluchowski equation is written in terms of $g(x, \tau) \equiv xf(x, \tau)$, the mass density function. The mass interval is discretized using a logarithmic grid. A system of ODEs is derived with respect to the variable g evaluated on the grid points. Gain and loss terms are evaluated separately, and integrals are calculated by using high-order Lagrangian interpolations. Middleton & Brock (1976) and Suck & Brock (1979) improved this method by using Simpson's rules for the integrals and cubic splines interpolations.

2.5.4 *Method of orthogonal collocation*

The method of weighted residuals (Finlayson 1972) is a general method for obtaining numerical solutions to differential equations. The unknown solution is tested over a set of weight functions and is adapted to give the best approximated solution to the differential equation. The Smoluchowski equation is multiplied by the weight function ϕ and integrated over all the mass domain to form the residual

$$R \equiv \int_0^\infty \left(\frac{\partial f(x,\tau)}{\partial \tau} - \int_0^x \mathcal{K}(x-y,y) f(x-y,\tau) f(x,\tau) \mathrm{d}y + \int_0^\infty \mathcal{K}(x,y) f(x,\tau) f(y,\tau) \mathrm{d}y \right) \phi(x) \mathrm{d}x = 0. \quad (9)$$

The number density f is approximated by polynomials. The collocation method corresponds to the case where $\phi(x) = \delta(x - x_0)$. The coagulation equation is evaluated at the collocation points x_0. This gives a set of ODEs equal to the degree of freedom of the polynomials used. Integrals are usually performed using Gaussian quadrature rules (Eyre, Wright & Reutert 1988).

2.5.5 *Pair interaction methods*

Numerical integration of the Smoluchowski equation consists of summing contributions of pairwise collisions between all grid points of different masses. For non-regular mass samplings, aggregates do usually not have masses corresponding to an existing grid point. To ensure mass conservation, the mass of the aggregate is distributed over the two relevant adjacent grid points (Fig. 3). The first pair-interaction solver has been developed by Kovetz & Olund (1969). In this algorithm, a system of ODEs is obtained over the quantities $N(x_i) = \int_{a_i}^{b_i} f(x)\mathrm{d}x$ where x_i denotes the mass of individual particles of the ith point, and $a_i \equiv (x_{i+1} - x_i)/2$ and $b_i \equiv (x_i - x_{i-1})/2$. In

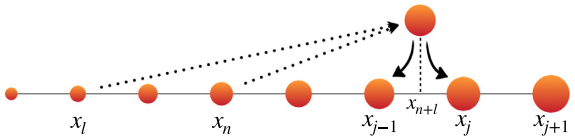

Figure 3. Illustration of the pair interaction methods. A particle of mass $x_{n+l} = x_n + x_l$ forms from collision between particles of masses x_l and x_n. The resulting mass x_{n+l} is distributed on to adjacent bins, generating numerical overdiffusion towards large masses.

practice, logarithmic grids are used to cover wide ranges of masses. In the context of planet formation, widely used solvers follow this approach (e.g. Brauer, Dullemond & Henning 2008; Birnstiel et al. 2010). The principal drawback of this method is that redistribution of mass towards large grains tend to overpredict the number of large aggregates, triggering artificial formation of large bodies (Fig. 1). A large number of grid points is therefore required to avoid an artificial broadening of number density of particles f (Berry & Reinhardt 1974; Soong 1974; Khain & Pinsky 2018). Moreover, a sufficient number of grid points is also needed to avoid difficulties related to collisions that form aggregates of masses larger than the largest mass point. Jacobson (2005) extended the Kovetz & Olund (1969) algorithm by distributing the mass between grid points and writing the scheme in a semi-implicit form. This solver ensures mass conservation to machine precision. Bott (1998), Simmel, Trautmann & Tetzlaff (2002), and Wang, Xue & Grabowski (2007) developed also binary-pairs interaction methods. Mass is advected towards adjacent grid points by a mass flux expressed with a high-order scheme. These methods do not introduce a significant numerical broadening. Other methods have been developed by Hounslow, Ryall & Marshall (1988) and Lister, Smit & Hounslow (1995) where four binary interaction mechanisms of gain and loss of particles are considered to deal correctly the rate of change of particle and mass.

2.5.6 *Finite element methods*

In these methods, the continuous mass distribution is discretized over a finite number of mass elements (intervals, cells, and bins).

2.5.7 *Moments with finite elements*

The first finite-element scheme for coagulation was developed by Bleck (1970) by discretizing mass distributions over logarithmic bins. f is approximated by its moment of order zero over each bin to obtain a system of ODEs. Overdiffusion for large grains is observed with this piecewise constant approximation. A change of variable $x \rightarrow x^{-3}$ is operated to reduce diffusivity at large masses. The method of Soong (1974) follows Bleck (1970). The Smoluchowski equation is written in terms of mass density distributions g and approximated by piecewise exponential functions. This allows to reduce drastically the diffusive effect at large masses. Gelbard et al. (1980) and Landgrebe & Pratsinis (1990) proposed a similar method, where the Smoluchowski equation is decomposed over bins of indices j in terms of $Q_j = \int_{I_j} xf(x,\tau)\mathrm{d}x$. A precise account of gain and loss of particles in terms of fluxes of Q is performed. Trautmann & Wanner (1999) extend the work of Gelbard et al. (1980), also finding numerical diffusion when using piecewise constant approximation, and addressing it by using piecewise exponential approximations. Another moment method that involves polynomial approximations for the first two moments M_0 and M_1 of f has been proposed by

Enukashvily (1980), Kumar & Ramkrishna (1996), and Tzivion, Reisin & Levin (1999).

2.5.8 *Discontinuous Galerkin method*

The discontinuous Galerkin (DG) method is a weighted residual method where the weight $\phi(x)$ consists of orthogonal polynomials (Lagrange polynomials, Legendre polynomials, and cubic splines). The numerical solution of the Smoluchowski equation is decomposed on each bin over this basis and a system of ODEs is obtained for the coefficients (e.g. Pilinis 1990; Erasmus, Eyre & Everson 1994; Mahoney & Ramkrishna 2002, see Section 3). Generally, the integrals are performed by Gaussian quadrature rules (Gelbard & Seinfeld 1978; Rigopoulos & Jones 2003; Sandu 2006).

2.5.9 *Finite-element schemes in the conservative form*

The conservative form equation (4) has been exploited for numerical simulations only lately. Filbet & Laurencot (2004) derived a finite-volume scheme of order zero where volume integrals over flux divergences are replaced by flux terms at the interfaces by the mean of the divergence theorem. This scheme conserves mass exactly and has been further extended by Filbet (2008), Bourgade & Filbet (2008), and Forestier-Coste & Mancini (2012). The mass interval can be sampled uniformly or non-uniformly. Finite-volume schemes of higher orders solving for the conservative form have been investigated recently (Gabriel & Tine 2010; Liu et al. 2019). Gabriel & Tine (2010) used a weighted essentially non-oscillatory (WENO) reconstruction (Jiang & Peng 2000) to approximate the coagulation flux at interfaces. Liu et al. (2019) developed a numerical scheme based on the DG method. This method provides the further advantage to choose the order of the scheme in a flexible manner. Integrals are calculated using Gaussian Quadrature rules, which implies subsampling of the mass intervals.

2.6 Requirements from hydrodynamical simulations

Densities must remain strictly positive and total mass conserved rigorously to ensure the stability of hydrodynamical simulations. These two properties are genuinely ensured by finite-volume methods based on the conservative form equation (4). The double-integral formulation allows to simply quench the formation of aggregates with unphysical masses, by setting for the integral bound the maximum mass allowed. These constrains may not always be satisfied with simple integral formulations.

On the other hand, observational constrains on young stellar objects are essentially provided by high-contrast spectropolarimetry at infrared wavelengths (SPHERE/VLT, GPI, and Subaru/HiCIAO) and millimetre interferometry (ALMA). These observations probe (sub)micron-to-millimetre-in-size dust distributions in discs, which corresponds to 4 orders of magnitude in size, that is, 12 orders of magnitude in mass for compact grains. With current computational capacities, 3D dust/gas simulations of dusty discs can handle $\sim$10–20 dust species simultaneously (e.g. PHANTOM, Price et al. 2018 or RAMSES, Lebreuilly, Commerçon & Laibe 2020). The global accuracy of second-order hydrodynamical solvers is of order $\sim 10^{-3}$. We aim therefore to design a versatile algorithm for coagulation of accuracy $\sim 10^{-3}$ with $\sim$15 dust bins distributed over 12 orders of magnitude in mass that allows tractable simulations. We therefore face the issue of overdiffusion associated to piecewise constant reconstructions with few mass bins, and high-order schemes appear as a natural way to overcome this difficulty. It is much preferable for hydrodynamics to handle a fix grid of sizes, to avoid interpolations when updating forces. We seek therefore for a growth algorithm that works efficiently with a fixed grid.

Additionally, we seek for an algorithm which allows for convergence analysis in 3D hydrodynamical simulations. As explained above, multiplying the number of dust bins provides prohibitive computational costs. Instead, the order of the scheme may be varied, should it be parametrized in a flexible manner. This requirement tends to favour DG schemes with respect to WENO schemes, although they provide in theory equivalent accuracies. Compared to regular Galerkin schemes, DG solvers decompose the solution over several mass bins. This helps to better capture the exponential decay of the solution at large masses and avoid overdiffusion biases. For these reasons, we have chosen to focus on the DG method to solve for the Smoluchowski equation in astrophysical contexts, an approach recently pioneered by Liu et al. (2019).

Monofluid dust/gas hydrodynamical solvers provide a natural architecture to include a coagulation equation. Indeed, relative drifts between grains of different sizes are genuinely computed, eventually in the terminal velocity approximation (e.g. Laibe & Price 2014; Hutchison, Price & Laibe 2018; Lebreuilly, Commerçon & Laibe 2019). Monofluid formalism also ensures exact conservation of momentum, that is, no thrust due to mass transfers propel the mixture. Subgrid fluctuations should be prescribed by an accurate model that describes local turbulence or Brownian motion.

3 DISCONTINUOUS GALERKIN ALGORITHM

3.1 Discontinuous Galerkin method

The DG method is presented for the general scalar hyperbolic conservative equation

$$\begin{cases} \frac{\partial g(x,\tau)}{\partial \tau} + \frac{\partial F[g](x,\tau)}{\partial x} = 0, \\ (x,\tau) \in \mathbb{R}_+, \end{cases} \tag{10}$$

where g is a density of a conservative quantity and $F[g]$ the associated flux.

Let partition the domain of interest $[x_{\min}, x_{\max}] \in \mathbb{R}$ in N subintervals (alternatively, cells or bins), not necessarily of equal sizes. Each cell is defined by $I_j = (x_{j-1/2}, x_{j+1/2}]$, $j \in [[1, N]]$. The size of the jth cell is defined as $h_j = x_{j+1/2} - x_{j-1/2}$. The cell is centred around the position $x_j = (x_{j+1/2} + x_{j-1/2})/2$. We define $\mathcal{V}^k$ the space of polynomials of degree k in each cell I_j

$$\mathcal{V}^k = \{v : v|_{I_j} \in P^k(I_j),\ j \in [\![1, N]\!]\}. \tag{11}$$

We denote $g_j \in \mathcal{V}^k$ the approximate solution of g in the bin I_j. The terminology *DG* comes from the fact that in $\mathcal{V}^k$, the functions are allowed to have jumps at the interfaces $x_{j+1/2}$. One obtains a weak formulation of equation (10) by multiplying by a test function $\phi \in \mathcal{V}^k$, integrating over I_j and finally integrating by parts (Cockburn & Shu 1989)

$$\int_{I_j} \frac{\partial g_j}{\partial t}\phi \mathrm{d}x - \int_{I_j} F[g](x,t)\frac{\partial \phi}{\partial x}\mathrm{d}x + F[g](x_{j+1/2},t)\phi(x_{j+1/2}) - F[g](x_{j-1/2},t)\phi(x_{j-1/2}) = 0. \tag{12}$$

Equation (12) allows to fix unequivocally the degrees of freedom of the function g_j. The residual of equation (10) on bin I_j is defined as

$$R_j \equiv \int_{I_j} \frac{\partial g_j}{\partial t}\phi \mathrm{d}x - \int_{I_j} F[g](x,t)\frac{\partial \phi}{\partial x}\mathrm{d}x + F[g](x_{j+1/2},t)\phi(x_{j+1/2}) - F[g](x_{j-1/2},t)\phi(x_{j-1/2}). \tag{13}$$

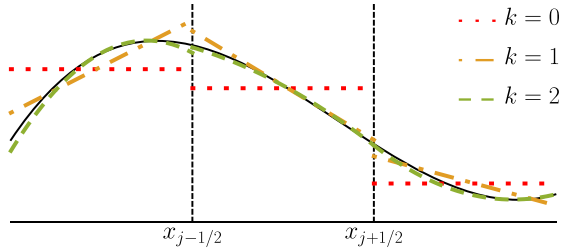

Figure 4. Sketch of the DG method. In each cell, the solution is approximated by high-order polynomials k to increase accuracy.

DG schemes consist of choosing a local orthogonal polynomials basis on I_j to replace the test function and to approximate the solution. Residuals R_j are therefore null in the sense of orthogonalization on the basis. In practice, Legendre polynomials are used (Cockburn & Shu 1989). We denote hereafter the ith Legendre polynomial by $\phi_i(\xi)$, where $\xi \in [-1, 1]$. Polynomial functions $\phi_i(\xi)$ are orthogonal in $L^2([-1, 1])$ with respect to the inner product with weight unity. Fig. 4 shows a sketch of the DG method. In each cell, the function g is approximated by Legendre polynomials. The accuracy of the approximation increases with respect to the order of the polynomials. The approximation of g in cell I_j writes

$$\forall x \in I_j,\ g(x) \approx g_j(x,t) = \sum_{i=0}^{k} g_j^i(t)\phi_i(\xi_j(x)),$$
$$g_j(x,t) = \boldsymbol{g}_j^T(t)\cdot\boldsymbol{\phi}(\xi_j(x)),\ \text{with } \boldsymbol{g}_j = \begin{bmatrix} g_j^0 \\ \vdots \\ g_j^k \end{bmatrix} \text{and } \boldsymbol{\phi} = \begin{bmatrix} \phi_0 \\ \vdots \\ \phi_k \end{bmatrix}, \tag{14}$$

where g_j^i is the component of g_j on the Legendre polynomials basis. The function $\xi_j(x) \equiv \frac{2}{h_j}(x - x_j)$ is used to map the interval I_j on to the interval $[-1, 1]$. Normalizing the Legendre basis gives

$$\int_{-1}^{1} \boldsymbol{\phi}(\xi)\boldsymbol{\phi}^T(\xi)\mathrm{d}\xi = d_i\delta_{ik} \text{ with } d_i \equiv \frac{2}{2i+1}, \tag{15}$$

where d_i is the coefficient of normalization. By combining equations (12), (14), and (15) one obtains

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{g}_j(t)}{\mathrm{d}t} &= \boldsymbol{L}[g] \text{ with}\\ \boldsymbol{L}[g] \equiv \frac{2}{h_j}&\begin{bmatrix} 1/d_0 & & \\ & \ddots & \\ & & 1/d_i \end{bmatrix}\Bigg(\int_{I_j} F[g](x,t)\partial_x\boldsymbol{\phi}(\xi_j(x))\mathrm{d}x\\ &- [F[g](x_{j+1/2},t)\boldsymbol{\phi}(\xi_j(x_{j+1/2}))\\ &- F[g](x_{j-1/2},t)\boldsymbol{\phi}(\xi_j(x_{j-1/2}))]\Bigg),\end{aligned} \tag{16}$$

where $\boldsymbol{L}$ is the operator that results from applying the DG procedure to equation (10) with a Legendre polynomials basis. With the procedure described above, the original system of partial differential equation (12) is transformed into a system of ODE (16) on to the coefficients $g_j^i(t)$. The initial condition $g_j(x, 0)$ is generated by the piecewise L^2 projection of an initial mass density distribution $g_0(x)$ on each bin, i.e.

$$\forall j \in [\![1, N]\!],\ \int_{I_j} (g_j(x,0) - g_0(x))\boldsymbol{\phi}^T(\xi_j(x))\mathrm{d}x = \boldsymbol{0}. \tag{17}$$

Orthogonality of Legendre polynomials ensures

$$\begin{aligned}\int_{I_j} g_j\boldsymbol{\phi}^T\mathrm{d}x &= \frac{h_j}{2}\int_{-1}^{1}\boldsymbol{\phi}(\xi)\boldsymbol{\phi}^T(\xi)\mathrm{d}\xi\,\boldsymbol{g}_j(t)\\ &= \frac{h_j}{2}\mathrm{diag}[d_0, ..., d_k]\boldsymbol{g}_j(t).\end{aligned} \tag{18}$$

Then, the components of $\boldsymbol{g}_j$ are given by

$$\forall j \in [\![1, N]\!], \forall i \in [\![0, k]\!],$$
$$g_j^i(0) = \frac{2}{h_j d_i}\int_{-1}^{1} g_0\left(\frac{h_j}{2}\xi_j + x_j\right)\phi_i(\xi_j)\mathrm{d}\xi_j. \tag{19}$$

Hence, the DG method consists in solving the following Cauchy problem

$$\begin{cases}\forall j \in [\![1, N]\!], \forall i \in [\![0, k]\!],\\ \frac{\mathrm{d}\boldsymbol{g}_j(t)}{\mathrm{d}t} = \boldsymbol{L}[g],\\ g_j^i(0) = \frac{2}{h_j d_i}\int_{-1}^{1} g_0\left(\frac{h_j}{2}\xi_j + x_j\right)\phi_i(\xi_j)\mathrm{d}\xi_j,\end{cases} \tag{20}$$

where $\boldsymbol{L}$ is detailed in equation (16).

3.2 Evaluation of the flux

3.2.1 *Regularized flux*

The continuous Smoluchowski coagulation equation (3) is defined over an unbounded interval of masses $x \in \mathbb{R}_+$. Before applying the DG procedure, equation (3) is restrained to a physical mass interval. Moreover, growth from a gaseous reservoir is excluded, meaning that $x > 0$. The mass interval is therefore reduced to the interval $[x_{\min} > 0, x_{\max} < +\infty]$ (Filbet & Laurencot 2004; Liu et al. 2019). The coagulation flux can be truncated according to two procedures (Filbet & Laurencot 2004). On the one hand

$$\begin{aligned}F_{\mathrm{coag}}^{\mathrm{c}}[g](x,\tau) = &\int_{x_{\min}}^{x}\int_{x-u+x_{\min}}^{x_{\max}-u+x_{\min}} \mathcal{K}(u,v)g(u,\tau)\\ &\times\frac{g(v,\tau)}{v}\mathrm{d}v\mathrm{d}u,\end{aligned} \tag{21}$$

where $F_{\mathrm{coag}}^{\mathrm{c}}$ is the conservative flux, meaning that no particle of mass larger than $x_{\max}$ is allowed to form. On the other hand

$$F_{\mathrm{coag}}^{\mathrm{nc}}[g](x,\tau) = \int_{x_{\min}}^{x}\int_{x-u+x_{\min}}^{x_{\max}} \mathcal{K}(u,v)g(u,\tau)\frac{g(v,\tau)}{v}\mathrm{d}v\mathrm{d}u, \tag{22}$$

where $F_{\mathrm{coag}}^{\mathrm{nc}}$ is the non-conservative flux which allows formation of particles of mass $x > x_{\max}$. $F_{\mathrm{coag}}^{\mathrm{c}}$ is useful in realistic simulations of growth, whereas $F_{\mathrm{coag}}^{\mathrm{nc}}$ should be used to compare numerical solution to analytic solutions of equation (1).

3.2.2 *Method for evaluating the flux*

A crucial difference between this scheme and usual DG solvers is that the coagulation flux $F_{\mathrm{coag}}^{\mathrm{nc}}$ is non-local. The evaluation of the numerical flux $F_{\mathrm{coag}}^{\mathrm{nc}}[g]$ at the interface $x_{j+1/2}$ depends on the evaluation of g_j in all cells. Mathematically, $F_{\mathrm{coag}}^{\mathrm{nc}}$ is a double integral of a product of polynomials. Then, the flux is a continuous function of mass x. At the interface $x_{j+1/2}$, the numerical flux reduces to $F_{\mathrm{coag}}^{\mathrm{nc}}[g] = F_{\mathrm{coag}}^{\mathrm{nc}}[g](x_{j+1/2}, t)$. In usual DG solvers, the numerical flux is a discontinuous function and must be reconstructed at the interfaces (e.g Cockburn & Shu 1989; Zhang & Shu 2010).

The principal difficulty lies in carefully evaluating the flux at interfaces. This relies on handling the numerical integration of the polynomials g_j in every relevant cell. Liu et al. (2019) use a Gaussian quadrature method with a Legendre polynomials basis to

approximate the flux. The lower bound of the inner integral $x - u$ does usually not correspond to a grid point. To accurately perform the Gauss quadrature, some grid elements must be subdivided, increasing drastically the cost of the numerical procedure, especially for high-order polynomials. To avoid prohibitive computational costs due to cell oversampling, we take advantage of the polynomial approximation by calculating integrals analytically. This requires integrable kernels, which is the case for the four kernels presented in this study. This approach maintains a reasonable computational cost by not multiplying the number of sampling points. This also avoid to add errors due to the numerical integration and to approximate kernels by piecewise constant functions.

3.2.3 Mathematical procedure

To integrate analytically the numerical flux, let define the function $\tilde{g}$ that approximates the function g over the entire mass interval

$$\forall x \in [x_{\min}, x_{\max}],$$
$$\tilde{g}(x,\tau) \equiv \sum_{l=1}^{N}\sum_{i=0}^{k} g_l^i(\tau)\phi_i(\xi_l(x))[\theta(x - x_{l-1/2}) - \theta(x - x_{l+1/2})]. \tag{23}$$

We assume that the kernel function is explicitly integrable and can be written as $\mathcal{K}(u,v) = \mathcal{K}_1(u)\mathcal{K}_2(v)$, which is effectively the case for the three simple kernels and the ballistic kernel (see Section 2.3). For instance, the additive kernel writes $\mathcal{K}_{\rm kadd}(u,v) = u + v = \mathcal{K}_1^1(u)\mathcal{K}_2^1(v) + \mathcal{K}_1^2(u)\mathcal{K}_2^2(v)$, where $\mathcal{K}_1^1(u) = u$, $\mathcal{K}_2^1(v) = 1$, $\mathcal{K}_1^2(u) = 1$ and $\mathcal{K}_2^2(v) = v$. The numerical flux is split in two terms. The numerical flux writes

$$\begin{aligned} F^{\rm nc}_{\rm coag}[\tilde{g}](x,t) = &\sum_{l'=1}^{N}\sum_{i'=0}^{k}\sum_{l=1}^{N}\sum_{i=0}^{k} g_{l'}^{i'}(t) g_l^i(t) \\ &\int_{x_{\min}}^{x}\int_{x-u+x_{\min}}^{x_{\max}} \frac{\mathcal{K}(u,v)}{v} \\ &\phi_{i'}(\xi_{l'}(u))[\theta(u - x_{l'-1/2}) - \theta(u - x_{l'+1/2})] \\ &\phi_i(\xi_l(v))[\theta(v - x_{l-1/2}) - \theta(v - x_{l+1/2})]\mathrm{d}v\mathrm{d}u. \end{aligned} \tag{24}$$

In the DG equation (12), the numerical flux is evaluated on grid points $x_{j+1/2}$ and $x_{j-1/2}$ with $j \in [[1, N]]$. k is the order of the Legendre polynomials to approximate the solution. Therefore, $F^{\rm nc}_{\rm coag}$ depends on j and k. The flux is sampled over a 2D array $(N, k+1)$ in order to use vectorial operations to reduce the computational time. The numerical flux is

$$\begin{cases} F^{\rm nc}_{\rm coag}[\tilde{g}](x,t) = \\ \displaystyle\sum_{l'=1}^{N}\sum_{i'=0}^{k}\sum_{l=1}^{N}\sum_{i=0}^{k} g_{l'}^{i'}(t) g_l^i(t) T(x, x_{\min}, x_{\max}, i', i, l', l), \\ T(x, x_{\min}, x_{\max}, i', i, l', l) = \\ \displaystyle\int_{x_{\min}}^{x} \mathcal{K}_1(u)\phi_{i'}(\xi_{l'}(u))[\theta(u - x_{l'-1/2}) - \theta(u - x_{l'+1/2})] \\ \displaystyle\int_{x-u+x_{\min}}^{x_{\max}} \frac{\mathcal{K}_2(v)}{v}\phi_i(\xi_l(v)) \\ \qquad [\theta(v - x_{l-1/2}) - \theta(v - x_{l+1/2})]\mathrm{d}v\mathrm{d}u. \end{cases} \tag{25}$$

A priori, the boundaries for the intervals of integration can be arbitrarily large. We therefore rescale these intervals to avoid any numerical issues related to large numbers when calculating the terms T in the variables $\xi_{l'}$ and ξ_l. To avoid critical typos, the term T is derived with MATHEMATICA by starting with the inner integral on ξ_l and then the integral on $\xi_{l'}$. Further details about the derivation of the algorithm are given in Supporting Information on GitHub (see Data Availability Section 6) for reproducibility. These integrals do not commute. The high-order solver is written in FORTRAN. Reducing the number of integrals is key to avoid numerical issues with differences of large numbers. For this purpose, the expression of T is split in several terms provided on GitHub (see Data Availability Section 6). For robustness, all these integrals are calculated with MATHEMATICA. The MATHEMATICA function `FortranForm` is used to translate integral expressions to FORTRAN. For large expressions, it is necessary to split them with the function `MonomialList`. The scheme to evaluate $T(x, x_{\min}, x_{\max}, i', i, l', l)$ in FORTRAN is given on GitHub (see Data Availability Section 6).

A 4D array with element $T(x, x_{\min}, x_{\max}, i', i, l', l)$ and a 4D array with element $g_{l'}^{i'}(t)g_l^i(t)$ are computed. The element (j, k) of the 2D array corresponding to the flux is obtained by multiplying these two 4D arrays and summing over of all elements. $F^{\rm nc}_{\rm coag}[\tilde{g}]$ is then evaluated in $x_{j-1/2}$ and $x_{j+1/2}$ for all j.

3.3 Evaluation of the integral of the flux

Let denote $\mathcal{F}^{\rm nc}_{\rm coag}$ the term of equation (12) corresponding to the integral of the numerical flux. $\mathcal{F}^{\rm nc}_{\rm coag}$ writes

$$\begin{cases} \mathcal{F}^{\rm nc}_{\rm coag}[\tilde{g}, j, k](t) = \\ \displaystyle\sum_{l'=1}^{N}\sum_{i'=0}^{k}\sum_{l=1}^{N}\sum_{i=0}^{k} g_{l'}^{i'}(t)\, g_l^i(t)\mathcal{T}\left(x_{\min}, x_{\max}, j, k, i', i, l', l\right) \\ \mathcal{T}\left(x_{\min}, x_{\max}, j, k, i', i, l', l\right) \equiv \\ \displaystyle\int_{I_j}\int_{x_{\min}}^{x}\int_{x-u+x_{\min}}^{x_{\max}} \frac{\mathcal{K}(u,v)}{v}\,\partial_x\phi_k(\xi_j(x)) \\ \qquad \phi_{i'}(\xi_{l'}(u))[\theta(u - x_{l'-1/2}) - \theta(u - x_{l'+1/2})] \\ \qquad \phi_i(\xi_l(v))[\theta(v - x_{l-1/2}) - \theta(v - x_{l+1/2})]\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}x. \end{cases} \tag{26}$$

$\mathcal{F}^{\rm nc}_{\rm coag}[\tilde{g}]$ is evaluated similarly to the flux. A triple integral is derived with MATHEMATICA with the changes of variables

$$\xi_l \equiv \frac{2}{h_l}(v - x_l),\ \xi_{l'} \equiv \frac{2}{h_{l'}}(u - x_{l'}),\ \xi_j \equiv \frac{2}{h_j}(x - x_j). \tag{27}$$

To derive tractable equations for the integrals involving Heaviside distributions, we start to compute integrals over the variable ξ_l, then calculating the integral over $\xi_{l'}$ and finally, over x. The details of the calculations and the scheme in FORTRAN to evaluate $\mathcal{T}(x_{\min}, x_{\max}, j, k, i', i, l', l)$ are given in Supporting Information on GitHub (see Data Availability Section 6) for completeness. $\mathcal{F}^{\rm nc}_{\rm coag}$ is computed as a product of 4D arrays similarly to $F^{\rm nc}_{\rm coag}$. Accuracy on T and $\mathcal{T}$ depends only the quality of the polynomial approximation of g by $\tilde{g}$, since the integrals corresponding to $F^{\rm nc}_{\rm coag}[\tilde{g}]$ and $\mathcal{F}^{\rm nc}_{\rm coag}[\tilde{g}]$ are calculated analytically.

3.4 Slope limiter

For most of astrophysical kernels, the solution of the Smoluchowski coagulation equation has been mathematically shown to decays with an exponential tail in at large masses (Schumann 1940; Menon & Pego 2004). This part is challenging to approximate with polynomials, and numerical estimates g_j of g in the bin I_j can lead to negative values, which is not acceptable physically.

To preserve the positivity of solution, the requirement $g_j(x, t) \geq 0$ for $x \in I_j$ needs to be enforced. The idea is to use a scaling limiter which controls the maximum/minimum of the reconstructed

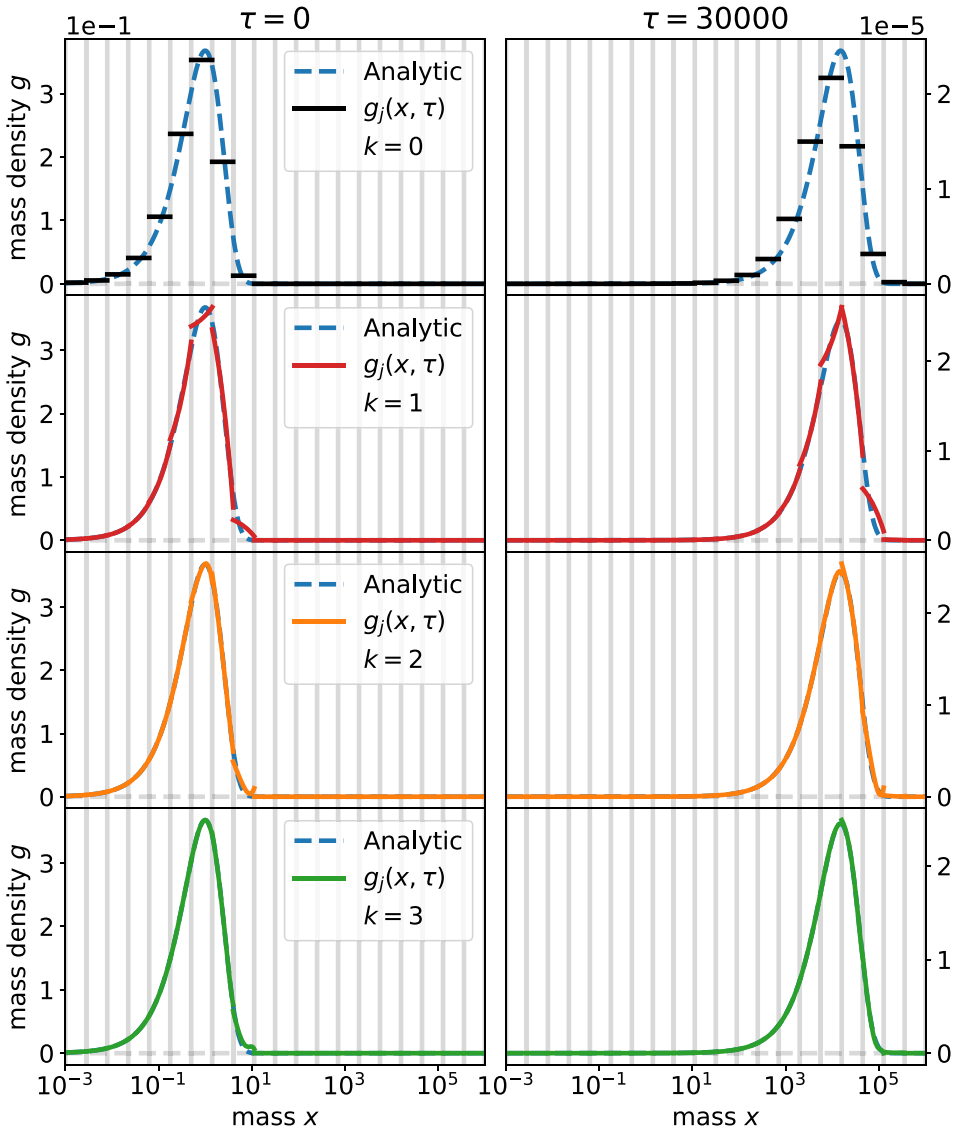

Figure 5. Test case, constant kernel: the numerical solution $g_j(x, \tau)$ is plotted for $N = 20$ bins and $k = 0$, 1, 2, and 3 from $\tau = 0$ to 30 000, and compared to the analytic solution $g(x, \tau)$. Vertical grey lines delimit the bins. The accuracy improves for larger values of k. Order 3 approximates the bump where the major part of the mass is concentrated with accuracy of order ∼0.1 per cent.

polynomials (Liu & Osher 1996; Zhang & Shu 2010; Liu et al. 2019). This is achieved by a reconstruction step based on cell averaging. Let us consider the polynomials $g_j(x)$ of order k that approximates $g(x)$ on I_j. Let denote m and M two positive reals $M_j \equiv \max_{x\in I_j} g_j(x)$, $m_j \equiv \min_{x\in I_j} g_j(x)$ and define the scaled polynomials

$$p_j(x) \equiv \gamma_j(g_j(x) - \overline{g}_j) + \overline{g}_j,$$
$$\gamma_j = \min\left\{\left|\frac{M - \overline{g}_j}{M_j - \overline{g}_j}\right|, \left|\frac{m - \overline{g}_j}{m_j - \overline{g}_j}\right|, 1\right\}. \quad (28)$$

where $\overline{g}_j$ refers to the cell average of g in I_j

$$\overline{g}_j \equiv \frac{1}{h_j}\int_{I_j} g_j(x,t)\mathrm{d}x. \quad (29)$$

For all j, we assume $\overline{g}_j \in [m, M]$. $p_j(x)$ is a polynomial of order k such as $\overline{p}_j = \overline{g}_j$. Liu & Osher (1996) proved that $\forall x \in I_j, p_j(x) \in [m, M]$. This scaling limiter allows to build a maximum-principle-satisfying DG scheme, in the sense that the numerical solution never goes out of the range $[m, M]$. The main difficulty is to ensure the property $\overline{g}_j \in [m, M]$ during the evolution without loosing high accuracy.

In the DG scheme given by equation (16), polynomials $g_j(x)$ are replaced by the scaled polynomials $p_j(x)$ such as

$$p_j(x) = \gamma_j\left(g_j(x) - \overline{g}_j\right) + \overline{g}_j$$
$$= \sum_{i=0}^{k} \gamma_j g_j^i(t)\phi_{1,i}(\xi_j(x)) + \sum_{i=0}^{k} g_j^i(t)\phi_{2,i}(\xi_j(x)) \quad (30)$$

with

$$\phi_{1,i}(\xi_j(x)) \equiv \left(\phi_i(\xi_j(x)) - \frac{1}{2}\int_{I_j}\phi_i(\xi_j(x))\mathrm{d}x\right),$$
$$\phi_{2,i}(\xi_j(x)) \equiv \frac{1}{2}\int_{I_j}\phi_i(\xi_j(x))\mathrm{d}x. \quad (31)$$

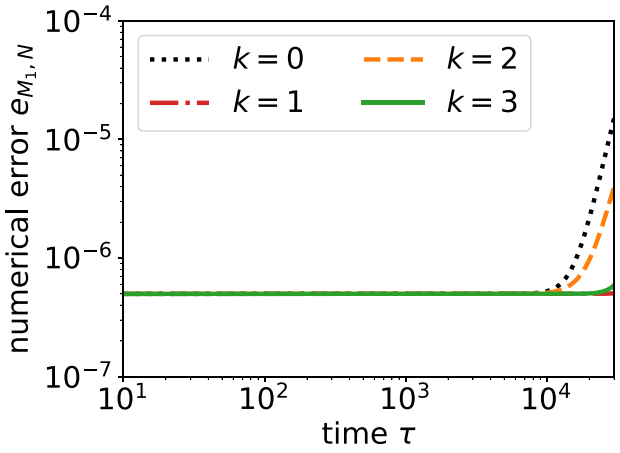

Figure 6 Test case, constant kernel: evolution of the numerical absolute error $e_{M_{1,N}}$ on the moment $M_{1,N}$ for $N = 20$ bins. The divergence at long times is explained by accumulation of errors due to numerical diffusion for even orders $k = 0$ and 2. Total mass is conserved at machine precision until $\tau = 10^4$.

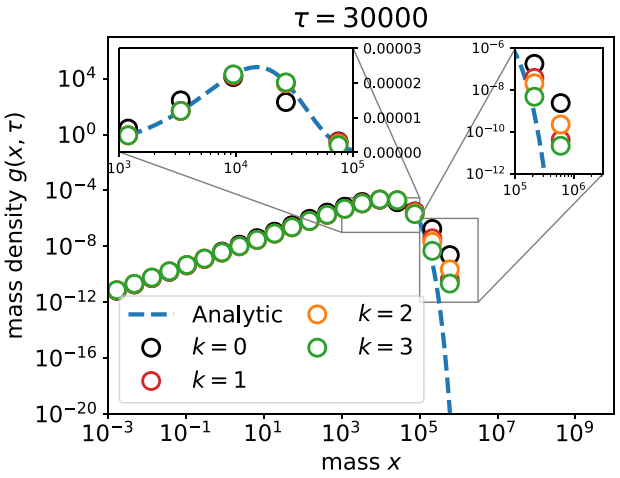

Figure 7. Test case, constant kernel: numerical solution $g_j(x, \tau)$ evaluated with the geometric mean $\hat{x}_j$ over each bin I_j. At the location of the maximum, orders $k = 1$, 2, and 3 achieve an absolute error of $\sim 0.1 - 1$ per cent, to be compared with 30 per cent obtained with $k = 0$. Accuracy in the exponential tail is improved by a factor 100 with $k = 3$ compared to $k = 0$.

Replacing g_j by p_j in equation (25) gives four terms for the function T: $T_{11}[\phi_{1,i'}\phi_{1,i}]$, $T_{12}[\phi_{1,i'}\phi_{2,i}]$, $T_{21}[\phi_{2,i'}\phi_{1,i}]$ and $T_{22}[\phi_{2,i'}\phi_{2,i}]$. For each term, a corresponding coefficient $g_{l',i'}(t)g_{l,i}(t)$ is associated, namely $\gamma_{l'}g_{l',i'}(t)g_{l,i}(t)$, $\gamma_{l'}g_{l',i'}(t)g_{l,i}(t)$, $\gamma_l g_{l',i'}(t)g_{l,i}(t)$, and $g_{l',i'}(t)g_{l,i}(t)$ (no γ in the last term). $F^{\rm nc}_{\rm coag}$ is evaluated by summing over those four terms. The same procedure is applied for $\mathcal{F}^{\rm nc}_{\rm coag}$. Therefore, the positivity of $\tilde{g}$ is ensured in each cell.

3.5 High-order time-stepping

3.5.1 CFL condition

Forward Euler discretization of equation (12) gives

$$\overline{g}_j^{n+1} = \overline{g}_j^n - \frac{\Delta t}{\Delta x_j}\left[F^{\rm nc}_{\rm coag}[g_j](x_{j+1/2},t) - F^{\rm nc}_{\rm coag}[g_j](x_{j-1/2},t)\right], \tag{32}$$

for the nth time-step. The Courant–Friedrichs–Lewy condition (CFL) of the scheme is chosen to guarantee the positivity of the cell average $\overline{g}_j^{n+1} > 0$ (Filbet & Laurencot 2004), that is,

$$\Delta t < \frac{\Delta x_j \overline{g}_j^n}{|F^{\rm nc}_{\rm coag}[g_j](x_{j+1/2},t) - F^{\rm nc}_{\rm coag}[g_j](x_{j-1/2},t)|}. \tag{33}$$

This CFL condition associated with the slope limiter (see Section 3.4) ensures the positivity of the global scheme. The CFL condition is initially dominated by small grains and softens as grains grow.

3.5.2 Strong stability preserving Runge–Kutta method

In equation (4), the spatial derivative $\partial_x F_{\rm coag}[g]$ is approximated by the non-linearly stable operator $-L[g]$ given in equation (16). For hyperbolic conservation laws, non-linear stability is characterized by the total variation diminishing (TVD) seminorm

$$\mathrm{TV}(g) \equiv \sum_j |\overline{g}_{j+1} - \overline{g}_j|. \tag{34}$$

The spatial discretiZation $-L[g]$ has the property that the total variation of the numerical solution does not increase for a forward Euler integration

$$g^{n+1} = g^n + \Delta t\, L[g], \;\; \Delta t \le \Delta t_{\rm FE}, \tag{35}$$

when $\Delta t_{\rm FE}$ the CFL condition determined in equation (33), that is, $\mathrm{TV}(g^{n+1}) \le \mathrm{TV}(g^n)$. TVD property can be generalized to high-order time discretization with a strong stability preserving (SSP) scheme (Shu & Osher 1988; Gottlieb, Shu & Tadmor 2001; Zhang & Shu 2010; Liu et al. 2019). The method is SSP if the following condition holds

$$\mathrm{TV}(g^{n+1}) \le \mathrm{TV}(g^n), \tag{36}$$

and the time-step satisfies

$$\Delta t_{\rm SSP} \le c\,\Delta t_{\rm FE}, \tag{37}$$

where c is a positive coefficient. Stability arguments are based on convex decomposition of high-order methods in term of the first-order Euler elements. This ensures that SSP preserves high-order accuracy in time for any convex functional (e.g. TV). In practice, errors are dominated by mass discretization. We use an SSP Runge-Kutta (SSPRK) third-order method (Gottlieb, Ketcheson & Shu 2009; Zhang & Shu 2010; Liu et al. 2019) which writes, with $c = 1$,

$$\begin{aligned}
\boldsymbol{g}_j^{(1)} &= \boldsymbol{g}_j^n + \Delta t_{\rm SSP}\boldsymbol{L}[g_j^n],\\
\boldsymbol{g}_j^{(2)} &= \frac{3}{4}\boldsymbol{g}_j^n + \frac{1}{4}\left(\boldsymbol{g}_j^{(1)} + \Delta t_{\rm SSP}\boldsymbol{L}\left[g_j^{(1)}\right]\right),\\
\boldsymbol{g}_j^{n+1} &= \frac{1}{3}\boldsymbol{g}_j^n + \frac{2}{3}\left(\boldsymbol{g}_j^{(2)} + \Delta t_{\rm SSP}\boldsymbol{L}\left[g_j^{(2)}\right]\right).
\end{aligned} \tag{38}$$

This SSPRK third-order method ensures that $\overline{g}_j \in [m, M]$ for $(m, M) \in \mathbb{R}_+$ at all times. Hence, under a suitable CFL condition, SSP high-order time discretization preserves the property $\overline{g}_j \in [m, M]$ of the DG scheme and the linear scaling presented in Section 3.4 satisfies a maximum principle.

3.6 Algorithm flowchart

Associating SSPRK with a DG scheme provides overall an high-order scheme that maintains overall a uniform high-order accuracy of the solution (Zhang & Shu 2010; Liu et al. 2019). We use the

SSPRK of third order given by equation (38). Splitting the algorithm into the following steps ensures positivity:

(i) Initialization: from the initial data $g_0(x)$,

(a) generate $\forall j \in [\![1, N]\!]$, $g_j(x, 0) \in \mathcal{V}^k$ by piecewise L^2 projection and get the components on Legendre basis equation (19),
(b) define $[m, M]$ for which $\overline{g}_j(x, 0) \in [m, M]$,
(c) replace g_j by p_j.

(ii) Evolution: use the scheme equation (38) to compute $\forall j \in [\![1, N]\!], \forall i \in [\![1, k]\!]$, $(g_j^i)^{n+1}$.
(iii) Reconstruction: use equation (30) to reconstruct $p_j(x, t)$.

4 NUMERICAL TESTS

The high-order solver presented in Section 3 is benchmarked against the analytical solutions presented in Section 2.4, similarly to Liu et al. (2019). Accuracy tests are performed with a small number of bins, consistently with hydrodynamical requirements.

4.1 Error measurements

Numerical simulations are carried out to (i) investigate the experimental order of convergence (EOC, Kumar, Kumar & Warnecke 2014; Liu et al. 2019) , and (ii) determine the efficiency of the algorithm. Relative errors are measured using a continuous norm and a discrete norm. The L^1 norm is a natural choice for equations of conservation. The continuous L^1 norm can be approximated by using a high-order Gaussian quadrature rule

$$\begin{aligned}\|f\|_1 &\equiv \int_{x_{\min}}^{x_{\max}} |f(x)|\mathrm{d}x \\ &= \sum_{j=1}^{N} \int_{I_j} |f(x)|\mathrm{d}x \approx \sum_{j=1}^{N} \frac{h_j}{2} \sum_{\alpha=1}^{R} \omega_\alpha |f\left(x_j^\alpha\right)|,\end{aligned} \tag{39}$$

where N is the number of bins, h_j is the size of bin I_j, ω_α are the weights, and x_j^α are the corresponding Gauss points in cell I_j. We use $R = 16$ for sufficient accuracy. The numerical error $e_{c,N}$ is measured with the continuous L^1 norm as

$$e_{c,N}(\tau) \equiv \sum_{j=1}^{N} \frac{h_j}{2} \sum_{\alpha=1}^{R} \omega_\alpha \left|g_j\left(x_j^\alpha, \tau\right) - g\left(x_j^\alpha, \tau\right)\right|, \tag{40}$$

where g and g_j are the analytic and the numerical solutions of the Smoluchowski equation. Equation (40) is computed with MATHEMATICA using 16 digits for sufficient precision. The discrete L^1 norm is defined by evaluating g_j and g at the geometric mean $\hat{x}_j \equiv \sqrt{x_{j-1/2} x_{j+1/2}}$ of the bin I_j. The numerical error measured with this discrete L^1 norm is

$$e_{d,N}(\tau) \equiv \sum_{j=1}^{N} h_j |g_j(\hat{x}_j, \tau) - g(\hat{x}_j, \tau)|. \tag{41}$$

We follow Liu et al. (2019) to calculate the EOC

$$\mathrm{EOC} \equiv \frac{\ln\left(\frac{e_N(\tau)}{e_{2N}(\tau)}\right)}{\ln(2)}, \tag{42}$$

where e_N is the error evaluated for N cells and e_{2N} for $2N$ cells. For the calculation of the EOC, the numerical errors are calculated at time $\tau = 0.01$ for the order of convergence of the DG scheme not to be altered by time-stepping errors. The moments of the numerical

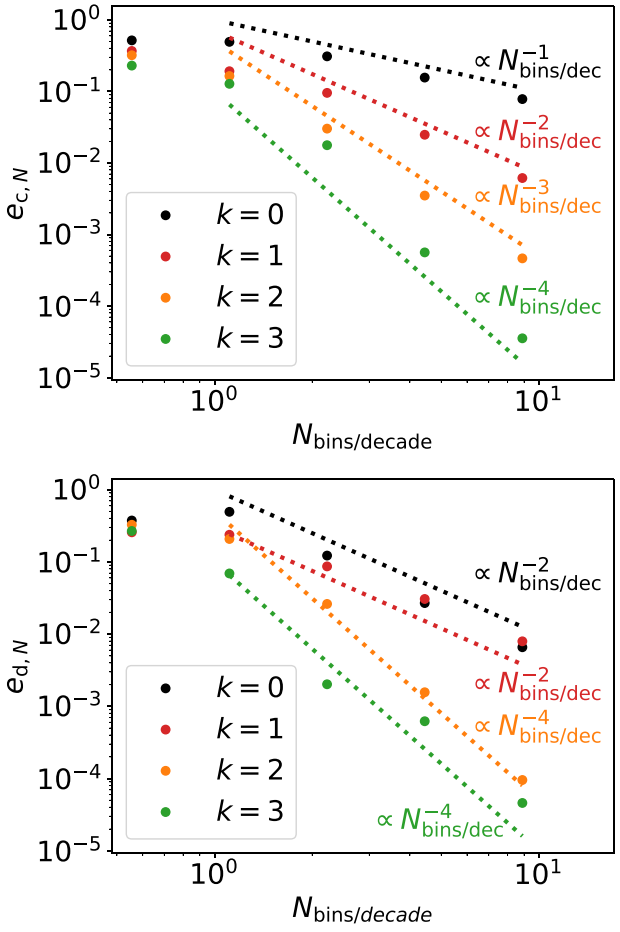

Figure 8. Test case, constant kernel: the continuous L^1 error $e_{c,N}$ and the discrete L^1 error $e_{d,N}$ are plotted as functions of the number of bins per decade. With $e_{c,N}$, the experimental order of convergence is EOC $= k + 1$. With $e_{d,N}$, EOC $= k + 1$ for polynomials of odd orders and EOC $= k + 2$ for polynomials of even orders. The DG scheme achieves on $e_{d,N}$ an accuracy of 0.1 per cent with more than 10 bins per decade for $k = 0$ and 1, with ∼9 bins per decade for $k = 2$ and with ∼5 bins per decade for $k = 3$. An accuracy of 1 per cent is achieved with ∼9 bins per decade for $k = 0$ and 1, with ∼5 bins/decade for $k = 2$ and ∼2 bins per decade for $k = 3$.

solution are defined according to

$$\begin{aligned}M_{p,N}(\tau) &= \int_{x_{\min}}^{x_{\max}} x^{p-1} \tilde{g}(x, \tau)\mathrm{d}x \\ &= \sum_{j=1}^{N} \int_{I_j} x^{p-1} g_j(x, \tau)\mathrm{d}x \\ &= \sum_{j=1}^{N} \sum_{i=0}^{k} g_j^i(\tau) \int_{I_j} x^{p-1} \phi_i(\xi_j(x))\mathrm{d}x.\end{aligned} \tag{43}$$

The total mass of the system writes

$$M_{1,N}(\tau) = \sum_{j=1}^{N} \sum_{i=0}^{k} g_j^i(\tau) \frac{h_j}{2} \underbrace{\int_{-1}^{1} \phi_i(\xi_j)\mathrm{d}\xi_j}_{=\delta_{00}=2} = \sum_{j=1}^{N} h_j g_j^0(\tau). \tag{44}$$

Absolute errors on moments are given by

$$e_{M_{p,N}}(\tau) \equiv \frac{|M_{p,N}(\tau) - M_p(\tau)|}{M_p(\tau)}, \tag{45}$$

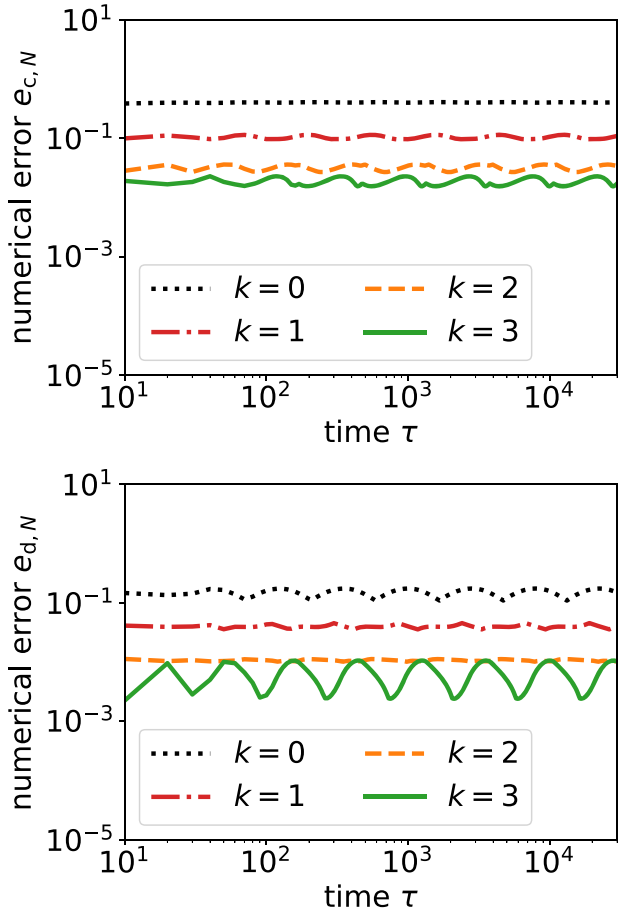

Figure 9. Test case, constant kernel: numerical errors $e_{c,N}$ with the L^1 continuous norm, $e_{d,N}$ with the discrete L^1 norm. All these errors are calculated for $N = 20$. Errors remain bounded at large times.

where $M_p(\tau)$ is the moment of order p at time τ for the exact solution. In usual convergence tests, errors are normalized with respect to the number of degrees of freedom of the algorithm. This is not the case here, since we compare absolute gains for the purpose interfacing it with an hydrodynamical solver.

4.2 Practical implementation of the tests

Numerical tests are performed by comparing numerical solutions the constant, additive and multiplicative kernels to the solutions given in equations (6)–(8). Solutions are integrated over the intervals $x \in [10^{-3}, 10^{6}]$ for the constant and the additive kernels, and $x \in [10^{-3}, 10^{3}]$ for the multiplicative kernel. Tests are performed with FORTRAN, errors are calculated with MATHEMATICA at machine precision. Quadruple precision is required for the additive kernel with $k = 2$, and for all kernels with $k = 3$. The results are shown for Legendre polynomials of order $k = 0$, 1, 2, and 3. Above order 3, numerical errors due to arithmatics of large numbers are not negligible anymore. A safety coefficient of 1/2 is applied on the CFL condition, that is, the coagulation time-step used in practice is $\mathrm{d}\tau_{\mathrm{coag}} = 1/2\,\mathrm{d}\tau_{\mathrm{CFL}}$. Initial conditions are set to satisfy the analytic solution at initial time $\tau = 0$. The analytical and numerical solutions are compared when particles of large masses are formed at final times τ that depend on the kernels. Simulations are performed by

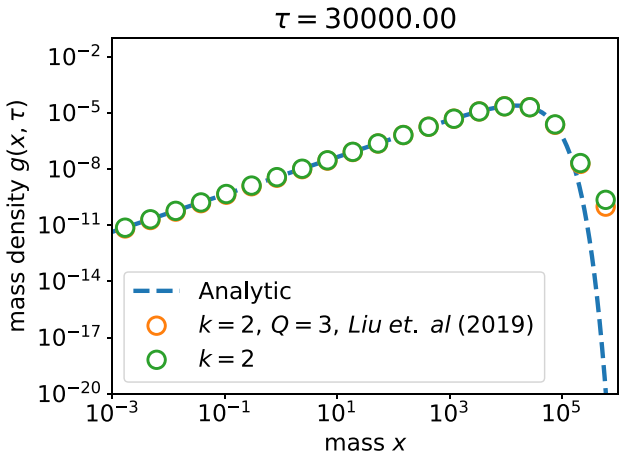

Figure 10. Test case, constant kernel: comparison with the scheme of Liu et al. (2019). Similar accuracies are reached, but being ∼4 times more effective due to numerical integration.

dividing τ into constant dumps of value $\mathrm{d}\tau$ (300 for the constant and the additive kernels, 10 000 for the multiplicative kernel). Each dump is subdivided in several coagulation steps satisfying the CFL condition. The analytical derivation of the coagulation flux allows the algorithm to be efficient, that is, to reach desired accuracy with a low computational time. To quantify efficiency, the computational time is compared to the one obtained with the scheme of Liu et al. (2019) with a number of Gauss points $Q = k + 1$ on a simulation in double precision with $N = 20$ bins, $k = 1$ for the additive kernel and $k = 2$ for the constant and multiplicative kernels. The Liu scheme is implemented by following the description of Liu et al. (2019) step-by-step, without additional optimizations. Simulations are performed in sequential on an Intel Core i7 2.8GHz. We use the GFORTRAN v9.2.0 compiler. Such a comparison is delicate to perform and interpret, since it is implementation-dependent. Should the number of Gauss points in the Liu algorithm be increased to better approximate the integral terms calculated here analytically, this may result in an increase of computational time by several orders of magnitudes, giving the false impression that the Liu algorithm is not performant. Hence the choice $Q = k + 1$. Qualitatively, our scheme is more effective by a factor of several unities for same precision and without requiring sub-binning, except for the additive kernel for which the Liu scheme exhibits serendipitous superconvergence (Liu et al. 2019).

4.3 Constant kernel

4.3.1 Positivity and mass conservation

Fig. 5 shows the numerical solutions obtained for $N = 20$ bins, varying the order of the polynomials k. The analytical and numerical solutions are compared at time $\tau = 30\,000$. As expected, the solution remains positive, as a result from combining the slope limiter (see Section 3.4) and the SSP Runge–Kutta time-stepping (see Section 3.5.2). The piecewise linear solution ($k = 1$) appears curved due to the logarithmic scale of the x-axis. Fig. 6 shows the numerical absolute error $e_{M_1,N}$ on the moment $M_{1,N}$ for $N = 20$ bins from $\tau = 0$ to 30 000. The total mass remains conserved to machine precision until $\tau = 10^4$.

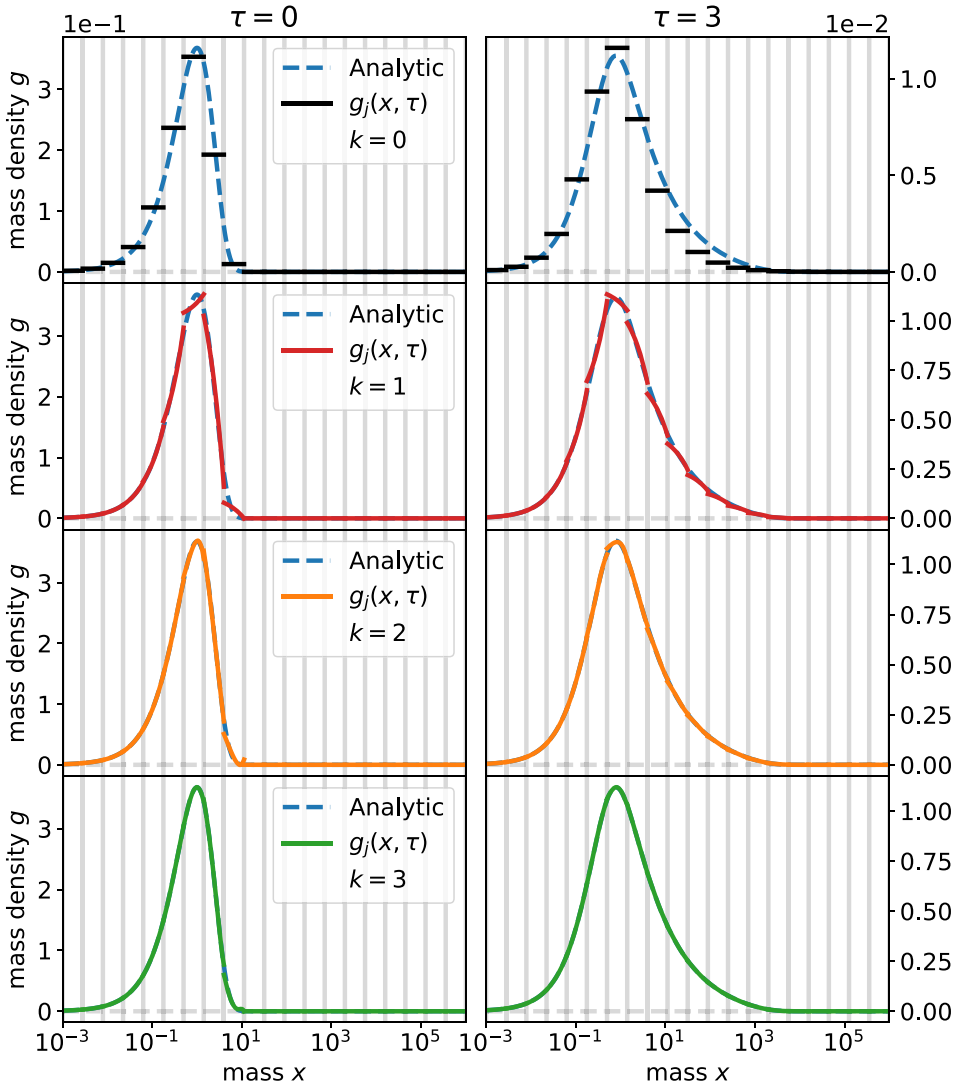

Figure 11. Test case, additive kernel: the numerical solution $g_j(x, \tau)$ is plotted for $N = 20$ bins and $k = 0$, 1, 2, and 3 from $\tau = 0$ to 3, and compared to the analytic solution $g(x, \tau)$. Vertical grey lines delimit the bins. The accuracy improves for larger values of k. Order 3 approximates the bump where the major part of the mass is concentrated with accuracy of order ∼0.1 per cent.

4.3.2 *Accuracy of the numerical solution*

As expected, the accuracy of the numerical solution improves with the order of the scheme. Fig. 7 shows the numerical solution obtained at $\tau = 30\,000$ (note the 16 orders of magnitude in mass on the y- axis in log). The major part of the total mass of the system is located around the maximum of the curve. Fig. 7 shows that around this maximum, schemes of order $k = 1$, 2, and 3 provide errors of order $\sim 0.1 - 1$ per cent when $k = 0$ generates errors of order ∼30 per cent. Fig. 7 also shows that numerical diffusion is drastically reduced in the exponential tail as the order of the scheme increases, since a gain of a factor ∼100 is obtained with order 3 compared to order 0.

4.3.3 *Convergence analysis*

Numerical errors introduced in Section 4.1 are shown on Fig. 8 at $\tau = 0.01$. $e_{c,N}$ and $e_{d,N}$ are plotted as a functions of the number of bins per decade $N_{\rm bin/dec}$, to infer the EOC independently from the global mass interval. With the continuous L^1 norm, the EOC is of order $k + 1$ on a geometric grid, similarly to Liu et al. (2019). With the discrete L^1 norm, the EOC is of order $k + 2$ for odd polynomials, and $k + 1$ for even polynomials. We recover second order of convergence (EOC=2) for the finite-volume scheme with $k = 0$ that was predicted by Filbet & Laurencot (2004). Fig. 8 shows that the expected accuracy of order ∼0.1 per cent on $e_{d,N}$ is achieved with more than 10 bins per decade for orders 0 and 1, with ∼9 bins per decade for order 2 and with ∼5 bins per decade for order 3. Accuracy of order ∼1 per cent is achieved with ∼9 bins per decade for orders 0 and 1, with ∼5 bins per decade for order 2, and with ∼2 bins per decade for order 3.

4.3.4 *Stability in time*

Time evolution of the numerical errors $e_{c,N}$ and $e_{d,N}$ are shown in Fig. 9. The results are shown for $N = 20$ bins for $k = 0$, 1, 2, and 3 at time $\tau = 30\,000$, when particles of large masses have formed. We verify that $e_{c,N}$ and $e_{d,N}$ remain bounded.

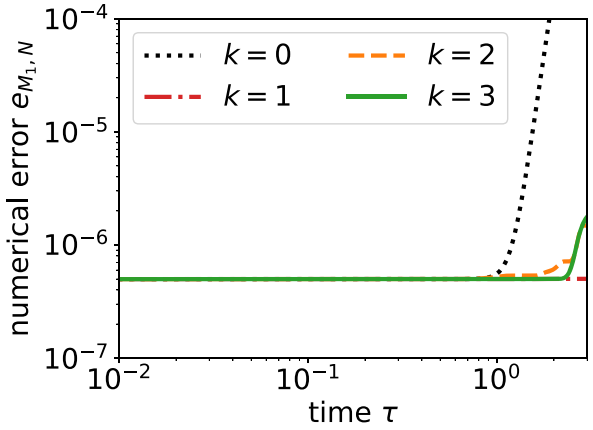

Figure 12. Test case, additive kernel: evolution of the numerical absolute error $e_{M_1,N}$ on the moment $M_{1,N}$ for $N = 20$ bins. The divergence at long times is explained by accumulation of errors due to numerical diffusion for orders $k = 0$, 2, and 3. Total mass is conserved at machine precision until $\tau = 1$.

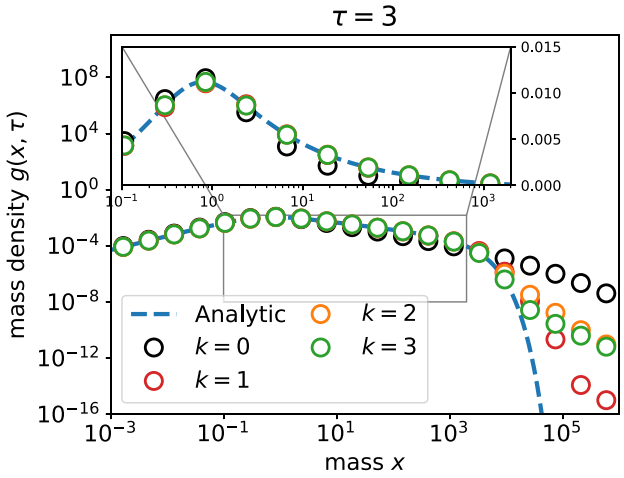

Figure 13. Test case, additive kernel: at the location of the maximum, orders $k = 1$, 2, and 3 achieve an absolute error of $\sim 0.1 - 1$ per cent, to be compared with 10 per cent obtained with $k = 0$. Accuracy in the exponential tail is improved by a factor 10 000 by $k = 3$ compared to $k = 0$.

4.3.5 *Computational efficiency*

Fig. 10 shows that similar accuracies are obtained with this scheme and the scheme described in Liu et al. (2019). Computational time is compared on a simulation with $N = 20$ bins, $k = 2$, and a final time $\tau = 30\,000$ after $\sim 10^3$ time-steps. The computational time for the Liu et al. (2019) scheme is around 16 s (real time). The computational time for this scheme is around 4 s (real time). An improvement of factor 4 is therefore achieved for the computational time by estimating integrals analytically.

4.4 Additive kernel

4.4.1 *Positivity and mass conservation*

Fig. 11 shows numerical solutions obtained for $N = 20$ bins and $k = 0$, 1, 2, and 3 at time $\tau = 3$. The numerical solutions remains positive as grains grow. Fig. 12 shows the evolution of the numerical absolute error $e_{M_1,N}$ on the first moment $M_{1,N}$. The total mass remains conserved to machine precision until $\tau = 1$.

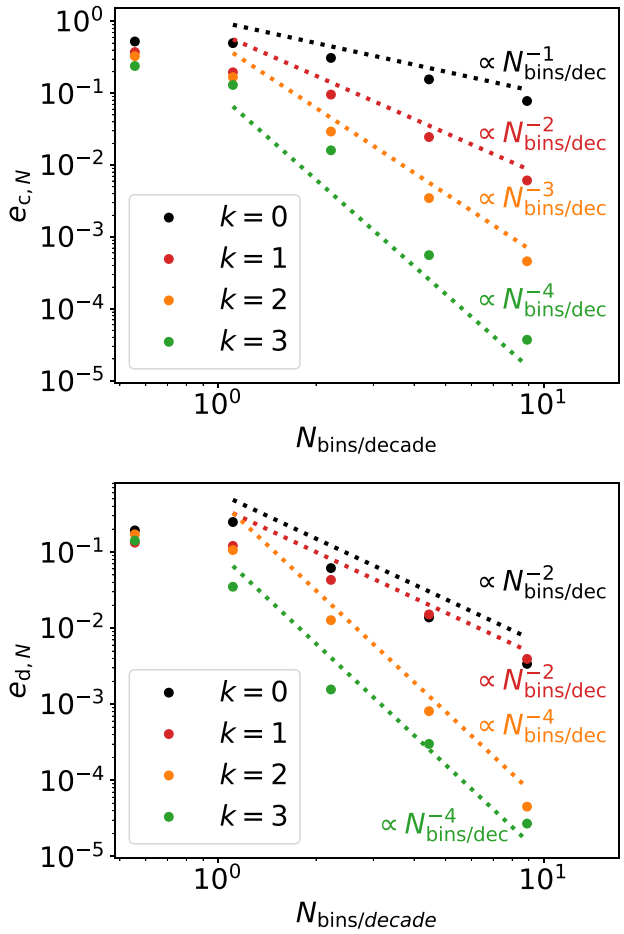

Figure 14. Test case, additive kernel: similar to Fig. 8. The DG scheme achieves on $e_{d,N}$ an accuracy of order 0.1 per cent with more than 10 bins per decade for $k = 0$ and 1, with ~ 5 bins per decade for $k = 2$ and 3. An accuracy of order 1 per cent is achieved with ~ 9 bins per decade for $k = 0$ and 1, with ~ 5 bins per decade for $k = 2$ and with ~ 2 bins per decade for $k = 3$.

4.4.2 *Accuracy of the numerical solution*

Fig. 13 shows numerical solutions obtained at $\tau = 3$ on a logarithmic scale. Fig. 13 reveals a strong numerical diffusion for order 0. Numerical errors are indeed integrated and diffused extremely efficiently towards large masses by the additive kernel. In this case, the mass density for large-masses particles is overestimated by several orders of magnitude. High-order schemes reduce this numerical diffusion as expected. Fig. 13 shows that around the maximum, schemes of order $k = 1$, 2, and 3 provide errors of order $\sim 0.1 - 1$ per cent when $k = 0$ generates errors of order ~ 10 per cent. Numerical diffusion is reduced in the exponential tail as the order of the scheme increases, up to reaching a gain of a factor $\sim 10\,000$ with order 3 compared to order 0.

4.4.3 *Convergence analysis*

Numerical errors are shown on Fig. 14 at $\tau = 0.01$. Accuracy of order ~ 0.1 per cent on $e_{d,N}$ errors are achieved with more than 10

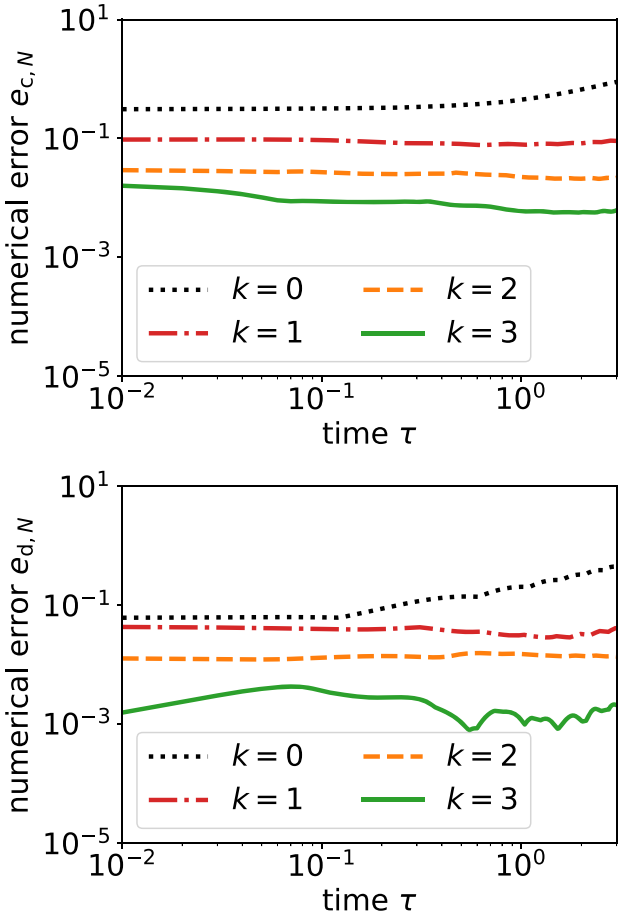

Figure 15. Test case, additive kernel: numerical errors $e_{c,N}$ with the L^1 continuous norm, $e_{d,N}$ with the discrete L^1 norm. All these errors are calculated for $N = 20$. Errors remain bounded at large times for orders $k = 1$, 2, and 3.

bins per decade for order 0 and 1, with ∼9 bins per decade for orders 2 and 3. Accuracy of order ∼1 per cent is achieved with ∼9 bins per decade for orders 0 and 1, with ∼5 bins per decade for order 2 and ∼2 bins per decade for order 3.

4.4.4 Stability in time

Evolution of the numerical errors $e_{c,N}$ and $e_{d,N}$ are shown in Fig. 15. The results are shown for $N = 20$ bins for $k = 0$, 1, 2, and 3 at $\tau = 3$, when particles with large masses have formed. At order 0, $e_{c,N}$ (respectively, $e_{d,N}$) increases significantly after $\tau \approx 5 \times 10^{-1}$ (respectively, $\tau \approx 10^{-1}$). On the contrary, $e_{c,N}$ and $e_{d,N}$ remain bounded for longer times at orders 1, 2, and 3.

4.4.5 Computational efficiency

Computational time is compared to Liu et al. (2019) on a simulation with $N = 20$ bins, $k = 1$ and a final time $\tau = 3$. Fig. 10 shows similar accuracy for both schemes. The computational time for the Liu et al. (2019) scheme is around 3 s (real time) for a number of Gauss quadrature points $Q = 2$. The computational time for this scheme is 1 s, providing an improvement by a factor 3. Fig. 16 also shows that for the additive kernel, the Liu scheme with $Q = 2$ is counterintuitively more accurate than for $Q = 16$ and the DG scheme. This result

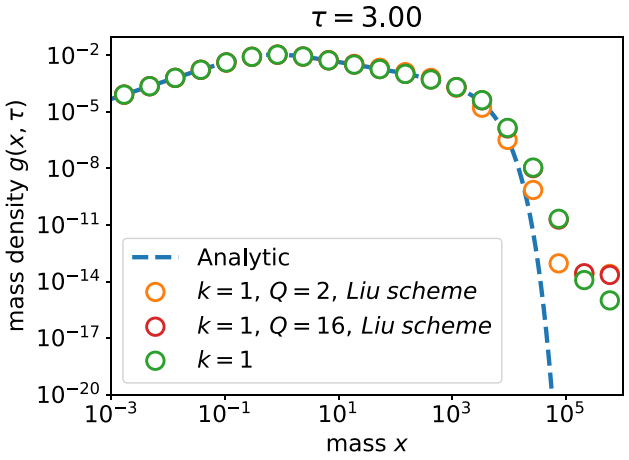

Figure 16. Test case, additive kernel: comparison with the scheme of Liu et al. (2019). Unexpected accuracy occurs for integral estimates with $Q = 2$ Gauss points due to serendipitous error compensations. Our algorithm is ∼3 times more effective due to analytical integration compared to the Lui scheme with $Q = 2$.

can be explained by a serendipitous compensation of errors when approximating the integrals with a Gauss quadrature of low order.

4.5 Multiplicative kernel

4.5.1 Positivity and mass conservation

Fig. 17 shows the numerical solutions obtained for $N = 20$ bins and $k = 0$, 1, 2, and 3 after $\tau = 100$. The numerical solutions remain positive as grain grow. Fig. 18 shows the evolution of $e_{M_1,N}$. Total mass remains conserved to machine precision until $\tau < 1$. At $\tau = 1$, gelation occurs, particles with infinite mass are formed (McLeod 1962b; Ernst et al. 1984; Filbet & Laurencot 2004) and total mass is no longer conserved anymore.

4.5.2 Accuracy of the numerical solution

Fig. 19 shows the numerical solution for the multiplicative kernel at $\tau = 100$. Accuracy of order ∼0.1 per cent is obtained at all orders, even $k = 0$. Physically, growth is effective enough for advection in the mass space to be more efficient than numerical diffusion.

4.5.3 Convergence analysis

Numerical errors are shown on Fig. 20 at $\tau = 0.01$. Accuracy of order ∼0.1 per cent on $e_{d,N}$ errors are achieved with ∼15 bins per decade for orders 0 and 1, with ∼7 bins per decade for order 2, and with ∼4 bins per decade for order 3. Accuracy of order ∼1 per cent is achieved with ∼7 bins per decade for orders 0 and 1, with ∼2 bins per decade for order 2, and with ∼1 bins per decade for order 3.

4.5.4 Stability in time

The evolution of the numerical errors $e_{c,N}$ and $e_{d,N}$ are shown in Fig. 21. The results are shown for $N = 20$ bins fo $k = 0$, 1, 2, and 3 at time $\tau = 100$, when particles with large masses have formed. We observe that $e_{c,N}$ and $e_{d,N}$ remain bounded, even after the occurrence of gelation at $\tau = 1$.

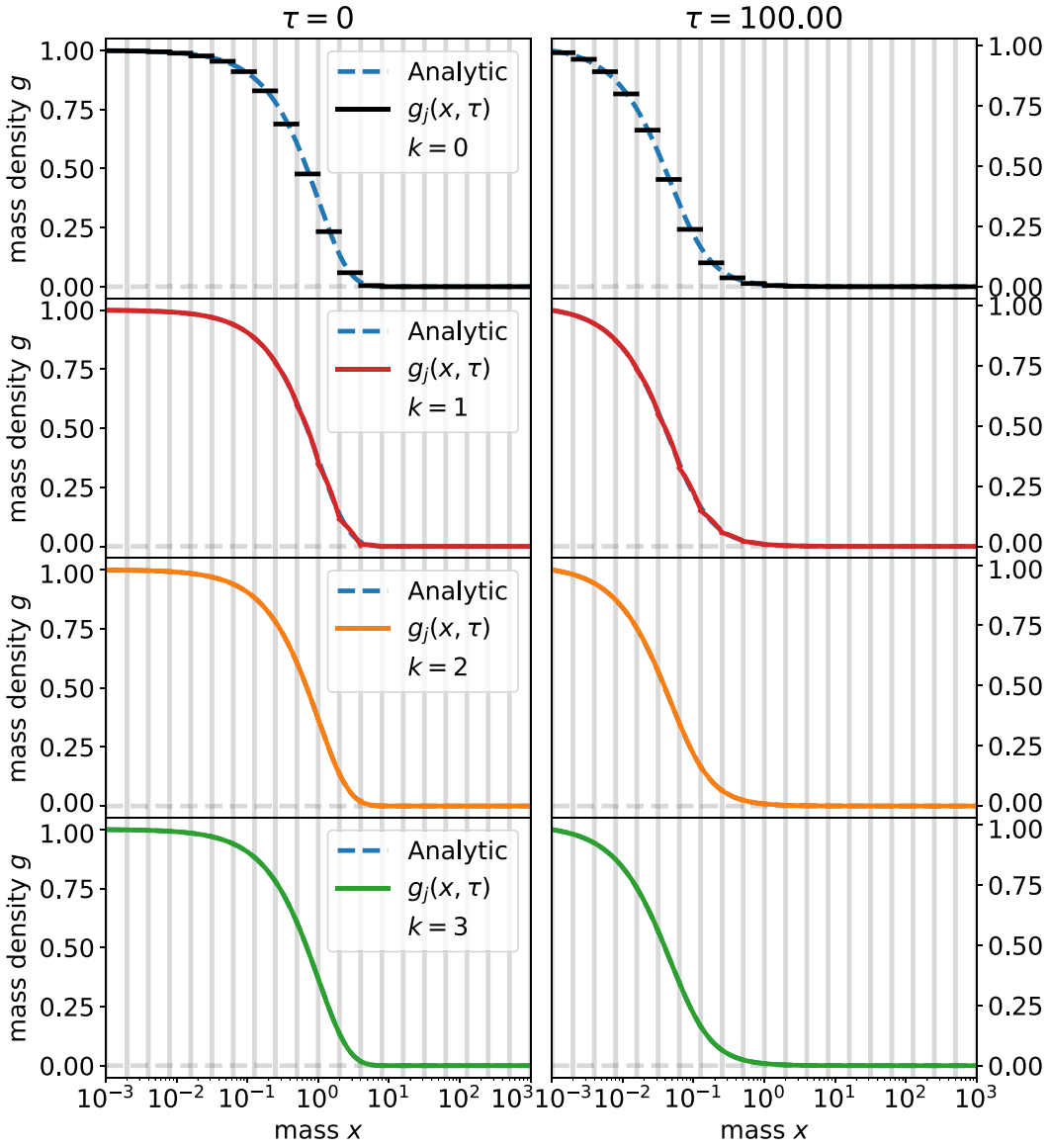

Figure 17. Test case, multiplicative kernel: numerical solution $g_j(x, \tau)$ is plotted for $N = 20$ bins for $k = 0$, 1, 2, and 3 from $\tau = 0$ to 100 and compared to the analytic solution $g(x, \tau)$. Vertical grey lines delimit the bins. Accuracy of order ∼0.1 per cent is achieved at all orders.

4.5.5 Computational efficiency

Fig. 22 shows similar accuracies for the Liu et al. (2019) scheme and our implementation. With $k = 2$, the computational time for the Liu et al. (2019) scheme is around 8 min for a number of Gauss quadrature points $Q = 3$. The computational time is for this scheme 1 min and 40 s, providing an improvement by a factor 5.

5 DISCUSSION

The DG scheme presented in Section 3 involves polynomials of high-order, implying issues with differences of large real numbers. Order $k = 3$ appears as a maximum limit for the order of the scheme in its current form in practice. So far, the ratio betwen 10^6 coagulation time-steps and one hydrodynamical time-step with PHANTOM using 10^6 smoothed particle hydrodynamics particles is of order ∼10–100. We are confident the we can reach a one-to-one ratio by (i) taking advantage of more ingenious time-stepping (e.g. Carrillo & Goudon 2004; Goudon, Lagoutière & Tine 2013, (ii) adopt a more relevant choice for the basis (e.g. Soong 1974), and (iii) use GPU parallelization, since calls to the coagulation solver by the hydrodynamical code are independent. These strategies to further gain accuracy and computational efficiency will be tested in a next future.

The most relevant kernel for astrophysics is the ballistic kernel (Section 2.3). Large-scale values of Δv are provided by 2D piecewise constant functions from hydrodynamic codes. In discs, the Δv function encompasses radial drift, vertical settling, and turbulence at large scales. The ballistic kernel splits in three terms

$$\begin{aligned}\mathcal{K}_b(u, v) &= \pi(u^{2/3} + 2u^{1/3}v^{1/3} + v^{2/3})\Delta v(u, v) \\ &= \mathcal{K}_{b,1}(u, v) + \mathcal{K}_{b,2}(u, v) + \mathcal{K}_{b,3}(u, v),\end{aligned} \tag{46}$$

$\mathcal{K}_{b,1}(u, v) \equiv \pi u^{2/3}\Delta v(u, v)$, $\mathcal{K}_{b,2}(u, v) \equiv \pi 2u^{1/3}v^{1/3}\Delta v(u, v)$, and $\mathcal{K}_{b,3}(u, v) \equiv \pi v^{2/3}\Delta v(u, v)$. The numerical flux is also split in three terms that are evaluated analytically. Models of differential

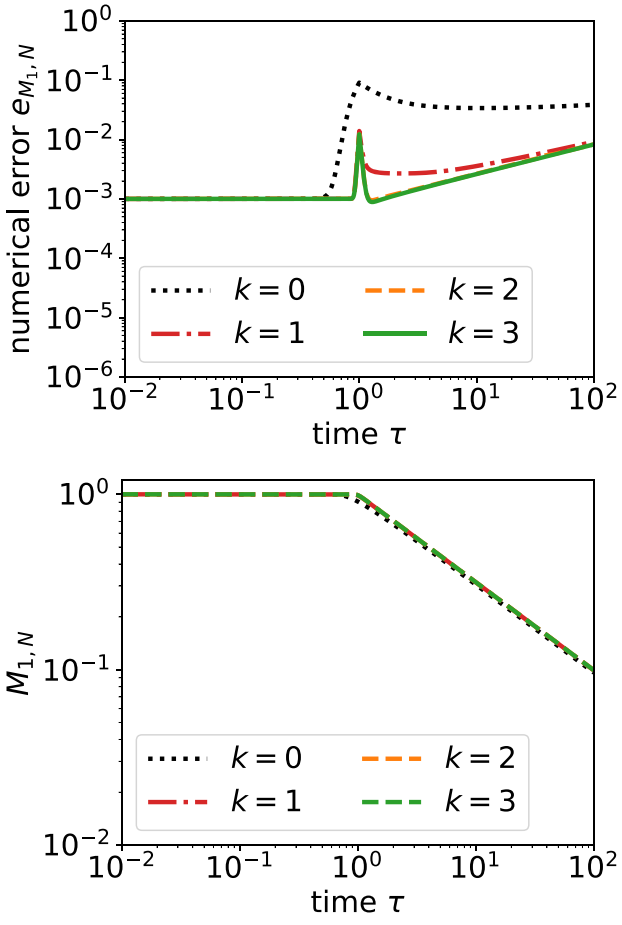

Figure 18. Test case, multiplicative kernel: evolution of the numerical absolute error $e_{M_1,N}$ on the moment $M_{1,N}$ for $N = 20$ bins. Mass is conserved anymore when gelation occurs at $\tau = 1$.

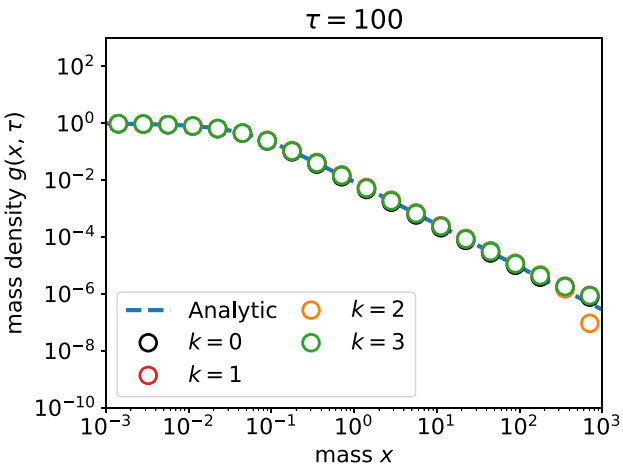

Figure 19. Test case, multiplicative kernel: accuracy of order ∼0.1 per cent is achieved at any order. Growth is so efficient than is overtakes numerical diffusion.

velocities are also used to model subgrid small-scale values of Δv (Brownian motion, dusty turbulence at small scales). Shall these kernels not be integrable, we will estimate them with an appropriate interpolation.

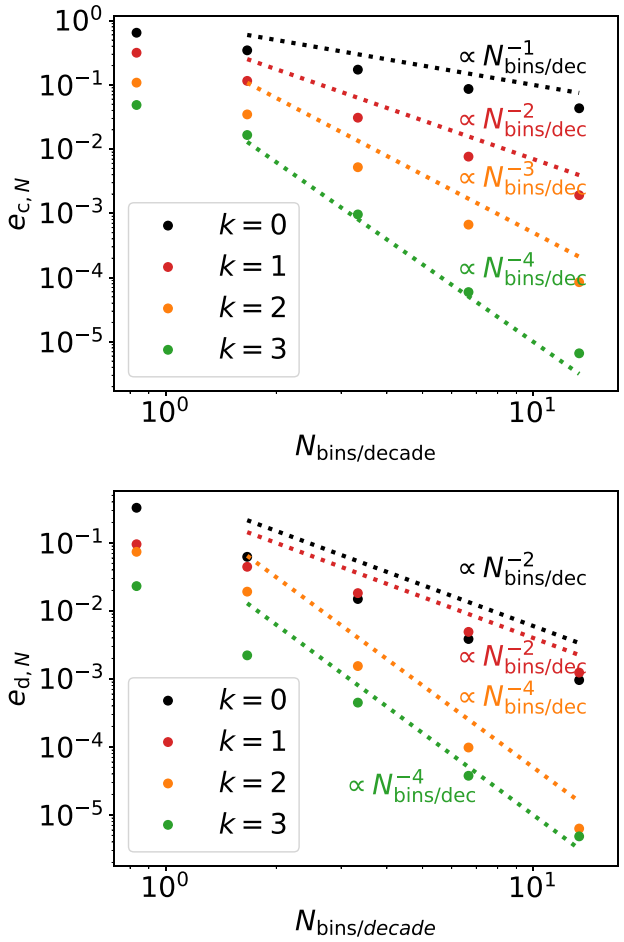

Figure 20. Test case, multiplicative kernel: the continuous L^1 error $e_{c,N}$ and the discrete L^1 error $e_{d,N}$ are plotted as functions of the number of bins per decade. With $e_{c,N}$, the experimental order of convergence is EOC $= k + 1$. With $e_{d,N}$, EOC $= k + 1$ for polynomials of odd orders and EOC $= k + 2$ for polynomials of even orders. The DG scheme achieves on $e_{d,N}$ an accuracy of 0.1 per cent with ∼15 bins per decade for $k = 0$ and 1, with ∼7 bins per decade for $k = 2$ and with ∼4 bins per decade for $k = 3$. An accuracy of 1 per cent is achieved with ∼7 bins per decade for $k = 0$ and 1, with ∼2 bins per decade for $k = 2$ and with ∼1 bins per decade for $k = 3$.

Moreover, the algorithm presented above solves for the Smoluchowski equation with pure growth. Although fragmentation plays a key role in regulating the number of small grains and preventing the formation of large bodies, it has not being included in the solver yet. The algorithm presented in Section 3 has been designed to incorporate fragmentation genuinely by adding the extra fragmentation flux (Paul & Kumar 2018)

$$F_{\text{frag}}[g](x,\tau) \equiv \int_0^\infty \int_x^\infty \int_0^x \frac{w}{yz} b(w,y,z)\mathcal{K}(y,z)g(y,\tau) \times g(z,\tau)\mathrm{d}w\mathrm{d}y\mathrm{d}z, \quad (47)$$

similarly, for example, to Birnstiel et al. (2010). The kernel $\mathcal{K}$ provides the fragmentation rate between two particles of masses x and y. The function b is the breakage rate related to the formation of a particle of mass x from particles of mass y and w. Known functional forms of the fragmentation kernel should authorize direct analytic integrations, similarly to the derivations performed in Section 3.2. For peculiar regimes, fragmentation kernels can alternatively be interpolated. Astrophysical mass distributions are expected to be dominated

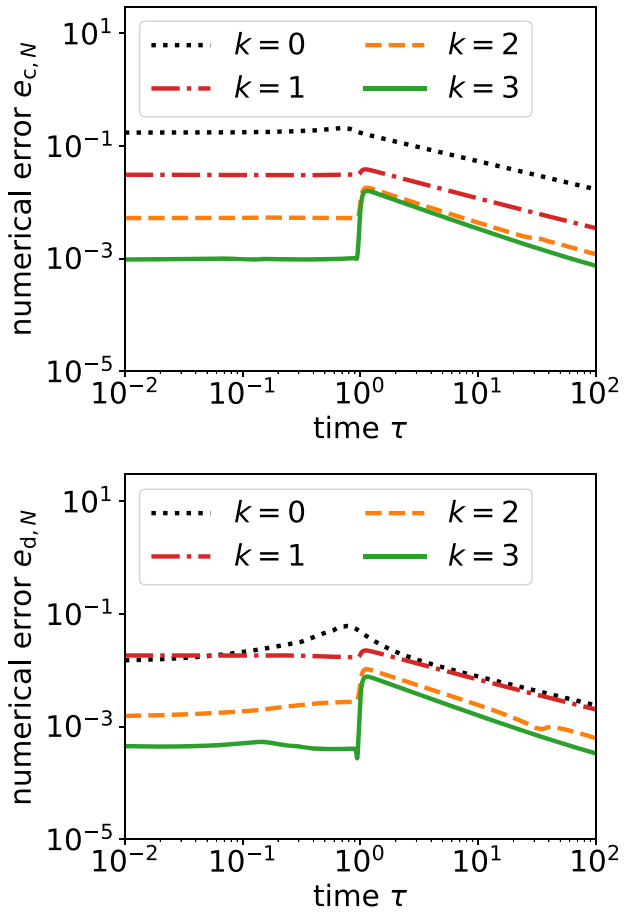

Figure 21. Test case, multiplicative kernel: numerical errors $e_{c,N}$ with the L^1 continuous norm, $e_{d,N}$ with the discrete L^1 norm. All these errors are calculated for $N = 20$. Errors remain bounded at large times.

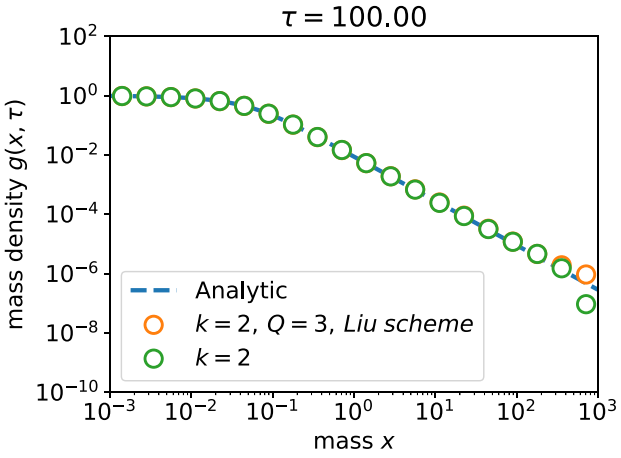

Figure 22. Test case, multiplicative kernel: comparison between the numerical solutions provided by this scheme and the scheme of Liu et al. (2019). Similar accuracies are reached, but being ∼5 times more effective due to analytical integration.

by large grains. Hence, the CFL condition for fragmentation should be similar to the one for growth (Vericel & Gonzalez 2020). If so, numerical integration will be performed explicitly. If not, implicit time-stepping can be implemented in a manageable way since the number of dust bins has been kept minimal with analytic integrations (i.e. linear algebra with $\sim 15 \times 15$ matrices).

Equation (1) restrains dust interactions to binary collisions between aggregates of spherical shapes. Multiple collisions are not expected to play a critical role in astrophysics, since dust volume densities are extremely low. On the other hand, dust aggregates are expected to be porous or have fractal structures. In particular, small bodies that have not been recompacted by collisions are expected to be fluffy. Equation (1) also reduces probability distributions of velocities to their mean values. This approximation may quench grain growth occurring through rare collisional events, for example, between large bodies having low relative velocities (Windmark et al. 2012; Garaud et al. 2013). Finally, growth is in essence stochastic, but fluctuations of the solution cannot be computed with equation (1). This is not critical, those being hardly constrained by observations. Although the solver presented in Section 3 cannot be used directly to treat the additional physical processes described above, the method could be adapted to do so. Lastly, extending equation (1) to multiple compositions, without or with change of states has been done in other communities. This comes to the cost of multiplying the number of variables by the number of materials considered. The algorithm presented in Section 3 is a first step towards reducing the number of dust bins to allow for solving for multiple compositions in 3D. This would have strong implications for planet formation, for example, by handling snow lines consistently and providing constrains for meteoritic data.

6 CONCLUSION

We have presented an high-order algorithm that solves accurately the coagulation equation with a limited number of dust bins (∼15). Specifically:

(i) mass is conserved to machine precision for astrophysical kernels,
(ii) positivity is guaranteed by combining an appropriate slope-limiter to a TVD time-stepping,
(iii) creating aggregates of masses larger that the mass of the reservoir is mathematically excluded by a control of the growth flux,
(iv) errors of order $0.1 - 1$ per cent are achieved by high-order discretization in time and space that can be modulated for convergence purpose. They shall not dominate the error budget over hydrodynamics,
(v) combining a low number of bins and analytic integrations allows manageable costs in memory and time,
(vi) additional physics should be implementable in a versatile way.

The next step consists of performing 3D hydrodynamical simulations of star and planet formation with accurate dust growth. The design of the algorithm allows to implement additional processes such as fragmentation in a genuine way. This solver encourages the reduction of CO_2 emissions related to computational astrophysics.

ACKNOWLEDGEMENTS

GL acknowledges funding from the ERC CoG project PODCAST no. 864965. This project has received funding from the European Union's Horizon 2020 research and innovation programme under

the Marie Skłodowska-Curie grant agreement no. 823823. This project was partly supported by the IDEXLyon project (contract nANR-16-IDEX-0005) under the auspices University of Lyon. We acknowledge financial support from the national programmes (PNP, PNPS, and PCMI) of CNRS/INSU, CEA, and CNES, France. We used MATHEMATICA (Wolfram Research 14). We thank L. Tine, E. Déléage, D. Price, D. Mentiplay, T. Guillet, S. Charnoz, R. Teyssier, and the anonymous referee for useful comments and discussions.

DATA AVAILABILITY

The data and supplementary material underlying this article are available in the repository 'growth' on GitHub at https://github.com/mlombart/growth.git. Figures can be reproduced following the file `README.md`. The repository contains data and PYTHON scripts used to generate figures.

REFERENCES

Aldous D. J., 1999, Bernoulli, 5, 3
ALMA Partnership et al., 2015, ApJ, 808, L3
Avenhaus H. et al., 2018, ApJ, 863, 44
Berry E. X., 1967, J. Atmos. Sci., 24, 688
Berry E. X., Reinhardt R. L., 1974, J. Atmos. Sci., 31, 1814
Birnstiel T., Dullemond C. P., Brauer F., 2010, A&A, 513, A79
Bleck R., 1970, J. Geophys. Res., 75, 5165
Bott A., 1998, J. Atmos. Sci., 55, 2284
Bourgade J.-P., Filbet F., 2008, Math. Comput., 77, 851
Brauer F., Dullemond C. P., Henning T., 2008, A&A, 480, 859
Carrillo J. A., Goudon T., 2004, J. Sci. Comput., 20, 69–113
Chandrasekhar S., 1943, Rev. Mod. Phys., 15, 1
Cockburn B., Shu C.-W., 1989, Math. Comput., 52, 411
Debry E., Sportisse B., Jourdain B., 2003, J. Comput. Phys., 184, 649
Dominik C., Blum J., Cuzzi J. N., Wurm G., 2007, Protostars and Planets, University of Arizona Press, Tucson, p. 783
Drake R. L., 1972, J. Atmos. Sci., 29, 537
Dullemond C. P., Dominik C., 2005, A&A, 434, 971
Enukashvily I. M., 1980, J. Atmos. Sci., 37, 2521
Erasmus L., Eyre D., Everson R., 1994, Comput. Chem. Eng., 18, 775
Ernst M. H., Ziff R. M., Hendriks E. M., 1984, J. Colloid Interf. Sci., 97, 266
Estrada P. R., Cuzzi J. N., 2008, ApJ, 682, 515
Eyre D., Wright C. J., Reutert G., 1988, J. Comput. Phys., 78, 288
Filbet F., 2008, in Simos T. E., Ch. Tsitouras eds, AIP Conf. Proc., Vol. 1048, International Conference on Numerical Analysis and Applied Mathematics, Am. Inst. Phys., p. 921
Filbet F., Laurencot P., 2004, SIAM J. Sci. Comput., 25, 2004
Finlayson B., 1972, The Method of Weighted Residuals and Variational Principles, Mathematics in Science and Engineering, Academic Press, London
Forestier-Coste L., Mancini S., 2012, SIAM J. Sci. Comput., 34, B840
Friedlander S. K. et al., 2000, Smoke, Dust, and Haze. Vol. 198. Oxford University Press, New York
Friedlander S. K., Wang C. S., 1966, J. Colloid Interf. Sci., 22, 126
Gabriel P., Tine L. M., 2010, in Bresch D., Calvez V., Grenier E., Vigneaux P., Gerbeau J.-F., eds, CEMRACS 2009: Mathematical Modelling in Medicine, Vol. 30, EDP Sciences, France, p. 53
Garaud P., Meru F., Galvagni M., Olczak C., 2013, ApJ, 764, 146
Gelbard F., Seinfeld J. H., 1978, J. Comput. Phys., 28, 357
Gelbard F., Tambour Y., Seinfeld J. H., 1980, J. Colloid Interf. Sci., 76, 541
Gillespie D. T., 1975a, J. Atmos. Sci., 32, 600
Gillespie D. T., 1975b, J. Atmos. Sci., 32, 1977
Golovin A., 1963, Izv. Geophys. Ser., 5, 482
Gonzalez J. F., Laibe G., Maddison S. T., 2017, MNRAS, 467, 1984
Gottlieb S., Shu C.-W., Tadmor E., 2001, SIAM Rev., 43, 89
Gottlieb S., Ketcheson D. I., Shu C.-W., 2009, J. Sci. Comput., 38, 251
Goudon T., Lagoutière F., Tine L. M., 2013, Math. Models Methods Appl. Sci., 23, 1177
Hayashi C., Nakagawa Y., 1975, Progress of Theoretical Physics, 54, 93
Hidy G., Brock J., 1972, Topics in Current Aerosol Research. Part 2. Pergamon Press, Oxford
Hounslow M. J., Ryall R. L., Marshall V. R., 1988, AIChE J., 34, 1821
Hulburt H., Katz S., 1964, Chem. Eng. Sci., 19, 555
Hutchison M., Price D. J., Laibe G., 2018, MNRAS, 476, 2186
Jacobson M. Z., 2005, Fundamentals of Atmospheric Modeling, Cambridge University Press, Cambridge
Jiang G.-S., Peng D., 2000, SIAM J. Sci. Comput., 21, 2126
Khain A. P., Pinsky M., 2018, Physical Processes in Clouds and Cloud Modeling. Cambridge University Press, Cambridge
Kostoglou M., Karabelas A. J., 1994, J. Colloid Interf. Sci., 163, 420
Kovetz A., Olund B., 1969, J. Atmos. Sci., 26, 1060
Kumar S., Ramkrishna D., 1996, Chem. Eng. Sci., 51, 1311
Kumar R., Kumar J., Warnecke G., 2014, Kinetic Relat. Models, 7, 713
Laibe G., Price D. J., 2014, MNRAS, 440, 2136
Landgrebe J. D., Pratsinis S. E., 1990, J. Colloid Interf. Sci., 139, 63
Laurençot P., 2018, J. Stat. Phys., 171, 484
Lebreuilly U., Commerçon B., Laibe G., 2019, A&A, 626, A96
Lebreuilly U., Commerçon B., Laibe G., 2020, A&A, 641, A112
Lee K., Matsoukas T., 2000, Powder Technol., 110, 82
Leyvraz F., Tschudi H. R., 1981, J. Phys. A Math. Gen., 14, 3389
Liffman K., 1992, J. Comput. Phys., 100, 116
Lister J. D., Smit D. J., Hounslow M. J., 1995, AIChE J., 41, 591
Liu H., Gröpler R., Warnecke G., 2019, SIAM J. Sci. Comput., 41, B448
Liu X.-D., Osher S., 1996, SIAM J. Numer. Anal., 33, 760
Mahoney A. W., Ramkrishna D., 2002, Chem. Eng. Sci., 57, 1107
Marchisio D. L., Vigil R. D., Fox R. O., 2003, J. Colloid Interf. Sci., 258, 322
McLeod J. B., 1962a, Proc. London Math. Soc., s3-14, 445
McLeod J. B., 1962b, Proc. London Math. Soc., s3-14, 445
Melzak Z. A., 1953, Quart. Appl. Math., 11, 231
Melzak Z., 1957, Trans. Am. Math. Soc., 85, 547
Menon G., Pego R. L., 2004, Commun. Pure Appl. Math., 57, 1197
Middleton P., Brock J., 1976, J. Colloid Interf. Sci., 54, 249
Müller H., 1928, Fortschrittsberichte über Kolloide und Polymere, 27, 223
Niethammer B., Throm S., Velázquez J. J. L., 2016, J. Stat. Phys., 164, 399
Ohtsuki K., Nakagawa Y., Nakazawa K., 1990, Icarus, 83, 205
Ormel C. W., Spaans M., Tielens A. G. G. M., 2007, A&A, 461, 215
Paul J., Kumar J., 2018, Math. Methods Appl. Sci., 41, 2715
Pilinis C., 1990, Atmos. Environ., 24, 1923
Pinte C. et al., 2020, ApJ, 890, L9
Piskunov V. N., Golubev A. I., 2002, J. Aerosol Sci., 33, 51
Price D. J. et al., 2018, PASA, 35, e031
Pruppacher H. R., Klett J. D., 1980, Nature, 284, 88
Pruppacher H. R., Klett J. D., 2010, in Microphysics of Clouds and Precipitation. Springer, Berlin, p. 10
Rajagopal E., 1959, Kolloid-Z., 167, 17
Ramkrishna D., 2000, Population Balances: Theory and Applications to Particulate Systems in Engineering. Elsevier, Amstredam
Rigopoulos S., Jones A. G., 2003, AIChE J., 49, 1127
Safronov V. S., 1972, Evolution of the Protoplanetary Cloud and Formation of the Earth and Planets, Israel Program for Scientific Translations, Keter Publishing House, Jerusalem, Israel
Sandu A., 2006, Aerosol Sci. Technol., 40, 261
Schumann T. E. W., 1940, Quart. R. Meteorol. Soc., 66, 195
Scott W. T., 1968, J. Atmos. Sci., 25, 54
Sheng C., Shen X., 2006, J. Aerosol Sci., 37, 16
Shu C.-W., Osher S., 1988, J. Comput. Phys., 77, 439
Silk J., Takahashi T., 1979, ApJ, 229, 242
Silk J., White S. D., 1978, ApJ, 223, L59
Simmel M., Trautmann T., Tetzlaff G., 2002, Atmos. Res., 61, 135
Smith M., Matsoukas T., 1998, Chem. Eng. Sci., 53, 1777

Smoluchowski M. V., 1916, Z. Phys., 17, 557
Soong S.-T., 1974, J. Atmos. Sci., 31, 1262
Suck S. H., Brock J. R., 1979, J. Aerosol Sci., 10, 581
Tanaka H., Inaba S., Nakazawa K., 1996, Icarus, 123, 450
Testi L. et al., 2014, Protostars and Planets VI, University of Arizona Press, Tucson, p. 339
Trautmann T., Wanner C., 1999, Atmos. Environ., 33, 1631
Trubnikov B. A., 1971, Sov. Phys. Dokl., 16, 124
Tzivion S., Reisin T. G., Levin Z., 1999, J. Comput. Phys., 148, 527
Vericel A., Gonzalez J.-F., 2020, MNRAS, 492, 210
Wang C.-S., 1966, PhD thesis. California Institute of Technology
Wang L.-P., Xue Y., Grabowski W. W., 2007, J. Comput. Phys., 226, 59
Weidenschilling S. J., 1980, Icarus, 44, 172
Wetherill G. W., 1990, Icarus, 88, 336
Windmark F., Birnstiel T., Güttler C., Blum J., Dullemond C. P., Henning T., 2012, A&A, 540, A73
Wolfram Research I., 14, Mathematica, Version 14.0. Available at: https://www.wolfram.com/mathematica
Zhang X., Shu C.-W., 2010, J. Comput. Phys., 229, 3091
Zsom A., Dullemond C. P., 2008, A&A, 489, 931

This paper has been typeset from a TEX/LATEX file prepared by the author.