A Multifluid Dust Module in Athena++: Algorithms and Numerical Tests

Pinghui Huang1 and Xue-Ning Bai1,2

1 Institute for Advanced Study, Tsinghua University, Beijing 100084, People’s Republic of China; phhuang@mail.tsinghua.edu.cn, xbai@tsinghua.edu.cn
2 Department of Astronomy, Tsinghua University, Beijing 100084, People’s Republic of China

Received 2022 April 25; revised 2022 June 1; accepted 2022 June 6; published 2022 August 19

Abstract

We describe the algorithm, implementation, and numerical tests of a multifluid dust module in the Athena++ magnetohydrodynamic code. The module can accommodate an arbitrary number of dust species interacting with the gas via aerodynamic drag (characterized by the stopping time), with a number of numerical solvers. In particular, we describe two second-order accurate, two-stage, fully implicit solvers that are stable in stiff regimes, including short stopping times and high dust mass loading, and they are paired with the second-order explicit van Leer and Runge–Kutta gas dynamics solvers in Athena++, respectively. Moreover, we formulate a consistent treatment of dust concentration diffusion with dust back-reaction, which incorporates momentum diffusion and ensures Galilean invariance. The new formulation and stiff drag solvers are implemented to be compatible with most of the existing features of Athena++, including different coordinate systems, mesh refinement, and shearing box and orbital advection. We present a large suite of test problems, including the streaming instability in linear and nonlinear regimes, as well as local and global settings, which demonstrate that the code achieves the desired performance. This module will be particularly useful for studies of dust dynamics and planet formation in protoplanetary disks.

Unified Astronomy Thesaurus concepts: Hydrodynamics (1963); Protoplanetary disks (1300); Computational methods (1965)

1. Introduction

Protoplanetary disks (PPDs) are composed of gas and dust. Although sharing only about 1% in mass, dust represents a fundamental building block of planets, and it is primarily the thermal radiation from dust that makes PPDs observable in continuum emission from infrared to millimeter wavelengths. Dust is coupled with gas via aerodynamic drag, characterized by the stopping time. Dust particles of small sizes have small stopping times and are strongly coupled to the gas, while larger dust particles are more loosely coupled, and hence do not necessarily trace the gas. This fact is not only important for many processes of planet formation, but also crucial for interpreting disk observations.

The initial stage of planet formation involves dust growth and transport, both of which are sensitive to the disk structure and level of turbulence (Ormel & Cuzzi 2007; Birnstiel et al. 2010). In particular, disk turbulence leads to dust diffusion (Cuzzi et al. 1993; Youdin & Lithwick 2007; Carballido et al. 2010; Zhu et al. 2015), which determines the thickness of the dust layer in the vertical direction, as well as the mixing in the radial direction. Additional “pseudo-diffusion” can result from complex radial gas flow structures, due to, e.g., wind-driven accretion (Hu & Bai 2021). Upon growing to larger sizes, back-reaction from dust to gas leads to dust clumping, due to the streaming instability (SI; Goodman & Pindor 2000; Youdin & Goodman 2005) and, subsequently, planetesimal formation (Johansen et al. 2007). While there has been a large number of further studies (e.g., Bai & Stone 2010a, 2010b; Carrera et al. 2015; Simon et al. 2017; Yang et al. 2017; Li & Youdin 2021), the SI interplay with more realistic gas dynamics is less clear (see Johansen et al. 2011; Schäfer et al. 2020; Xu & Bai 2022). Finally, instead of planetesimal accretion, the growth of planetary cores by pebble accretion has been identified as being more efficient toward higher core mass (Ormel & Klahr 2010; Lambrechts & Johansen 2012). The efficiency of pebble accretion again depends on the disk structure and level of turbulence (e.g., Morbidelli et al. 2015; Xu et al. 2017), and back-reaction from dust to gas may destabilize the feeding zone (Fu et al. 2014; Pierens et al. 2019; Hsieh & Lin 2020; Huang et al. 2020; Surville et al. 2020; Yang & Zhu 2020), which requires careful study considering realistic gas dynamics in 3D.

Over the past decade, thanks to the advent of the Atacama Large Millimeter/submillimeter Array (ALMA), as well as the high-contrast imaging techniques that ground-based telescopes are equipped with, the dramatically improved resolution and sensitivity have led to the discovery of the disk substructures prevalent in PPDs, particularly in the forms of rings and gaps, as well as various forms of asymmetries (see Andrews 2020 for a review). These features are commonly interpreted as a consequence of planet–disk interaction, which can open gaps (Bae et al. 2017; Dong et al. 2017, 2018), create vortices (van der Marel et al. 2013; Zhu et al. 2014; Flock et al. 2015), drive spirals (Dong et al. 2011a, 2011b; Bae & Zhu 2018a, 2018b), etc. At millimeter/submillimeter wavelength, the observed substructures reflect the distribution of millimeter-sized dust particles, which likely substantially amplify the substructures in the gaseous disk, because these particles are not strongly tied to gas and tend to drift toward pressure maxima (Whipple 1972; Weidenschilling 1977). Alternatively, a number of nonplanet mechanisms that lead to substructure formation have been identified, such as processes involving snow lines (Zhang et al. 2015; Okuzumi et al. 2016; Owen 2020) and magnetohydrodynamic (MHD) effects (Suriano et al. 2018; Riols et al. 2020; Cui &
Bai 2021). Some of the mechanisms require active participation from the dust itself, due to its back-reaction (Takahashi & Inutsuka 2014, 2016; Tominaga et al. 2019, 2020). In all these scenarios, it is crucial to coevolve gas and dust in a self-consistent manner, to help constrain the physical mechanisms behind the observations.

Computationally, dust is commonly treated either as Lagrangian (super)particles or as pressureless fluids. The particle methods have been implemented in several MHD codes, including Pencil (Johansen et al. 2007), Athena (Bai & Stone 2010c), FARGO-ADSG (Baruteau & Zhu 2016), and PLUTO (Mignone et al. 2019). It has also been naturally employed in smoothed particle hydrodynamic codes, including PHANTOM (Price et al. 2018). One major advantage of the Lagrangian treatment is being able to properly handle particle crossing, which is more relevant for particles that are marginally or loosely coupled to the gas, and which is important for studying planetesimal formation by the SI. On the other hand, it is generally difficult to handle the highly stiff regime of extreme particle concentration (Bai & Stone 2010c, but see Yang & Johansen 2016; Moseley et al. 2022), and achieving good load balancing can be challenging for very large simulations (but see Johansen et al. 2011). Moreover, it is common to treat the unspecified source of disk turbulence as an effective viscosity in gas dynamic simulations. Doing so for particles can be involved, especially if one were further to consider dust back-reaction.

The alternative fluid treatment of dust is gaining popularity, such as in PIERNIK (Hanasz et al. 2010a, 2010b), MPI-AMRVAC (Porth et al. 2014; Xia et al. 2018), LA-COMPASS (Li et al. 2005, 2009; Fu et al. 2014), and FARGO3D (Benítez-Llambay & Masset 2016; Benítez-Llambay et al. 2019). This approach is more appropriate for relatively strongly coupled dust, as it quickly responds to fluid motion to minimize particle crossing. As separate fluids are colocated with gas in the computational domain, stiffness issues can be overcome by designing fully implicit schemes for the drag source term on gas and dust simultaneously, and load balancing can be trivially satisfied. Dust diffusion can be easily handled, by incorporating a concentration diffusion source term (Cuzzi et al. 1993; Youdin & Lithwick 2007). Finally, this approach is generalizable to further incorporate dust coagulation (Drazkowska et al. 2019; Li et al. 2019, 2020), so that one can self-consistently compute the dust size distribution at every simulation cell.

In this paper, we describe the algorithm, implementation, and numerical tests of a multifluid dust module in the Athena++ MHD code (Stone et al. 2020). Our development features a set of dust integrators, particularly two fully implicit integrators that can handle all stiff regimes while maintaining second-order accuracy, which improve upon previous works that were either explicit, such as MPI-AMRVAC (Porth et al. 2014), FARGO-ADSG (Baruteau & Zhu 2016), and PHANTOM (Price et al. 2018), or implicit, but only first-order accurate (FARGO3D; Benítez-Llambay et al. 2019). With Athena++ being a Godunov MHD code, our implementation naturally conserves total momentum and energy. Moreover, we provide a consistent formulation of dust concentration diffusion, and show that additional correction terms in the momentum equations of dust are necessary to properly conserve total momentum and maintain Galilean invariance. Implementing these terms yields physically sensible results in a number of test problems.

The outline of this paper is as follows. In Section 2, we describe the equations, our numerical schemes, and the implementations. In Section 3, we present the benchmark tests, including collisions between gas and dust, dust diffusion with or without momentum correction, linear and nonlinear tests of the SI, and global curvilinear simulations of the SI, as well as static/adaptive mesh refinement (SMR/AMR) tests. Finally, we summarize and discuss our results in Section 4.

2. Numerical Scheme

In this section, we describe the basic equations, including the consistent formulation of the dust concentration diffusion, as well as the numerical schemes and the implementation of the multifluid dust module in Athena+++

2.1. General Equations (Conservative Form)

We start by presenting the full set of equations of gas and multifluid dust. We use the subscripts “d” and “g” to denote “dust” and “gas,” respectively. Let there be $N_d$ dust species, each characterized by a stopping time $T_{st,n}$, representing the timescale on which they respond to gas drag, where we use the label “n” for the nth dust species. In conservative form, the equations read:

$$\frac{\partial \rho_d}{\partial t} + \nabla \cdot (\rho_d \mathbf{v}_d) = 0,$$

$$\frac{\partial (\rho_d \mathbf{v}_d)}{\partial t} + \nabla \cdot (\rho_d \mathbf{v}_d \mathbf{v}_d + P_g I) + \mathbf{F}_d = 0,$$

$$\frac{\partial E_g}{\partial t} + \nabla \cdot \left[ (E_g + P_g) \mathbf{v}_g + \mathbf{F}_d \cdot \mathbf{v}_g \right] = \rho_d \mathbf{f}_{g,src} \cdot \mathbf{v}_g + \sum_{n=1}^{N_d} \rho_{d,n} \mathbf{v}_{d,n} - \mathbf{v}_g \left( \omega \sum_{n=1}^{N_d} \rho_{d,n} \frac{\mathbf{v}_{d,n}}{T_{st,n}} \right)^2,$$

$$\frac{\partial \rho_{d,n}}{\partial t} + \nabla \cdot \left( \rho_{d,n} \mathbf{v}_{d,n} + \mathbf{F}_{diff,n} \right) = 0,$$

$$\frac{\partial \rho_{d,n} \mathbf{v}_{d,n}}{\partial t} + \nabla \cdot \left( \rho_{d,n} \mathbf{v}_{d,n} \mathbf{v}_{d,n} + \mathbf{F}_{diff,n} \right) = \rho_{d,n} \mathbf{f}_{d,src,n} + \rho_{d,n} \mathbf{v}_d \frac{\mathbf{v}_{d,n}}{T_{st,n}}.$$

There are $4N_d + 5$ equations in total, where Equations (1) to (3) are the gas continuity, momentum, and energy equations, and Equations (4) and (5) are the continuity and momentum equations for the dust species, which are treated as pressureless fluids (Garaud et al. 2004). In the above, $\rho$ is the density, $\mathbf{v}$ is the velocity, $P_g$ is the gas pressure, I is the identity tensor, and $E_g = \left( \rho_d \mathbf{v}_g^2 / 2 + P_g / (\gamma - 1) \right)$ is the total energy density of the gas, with $\gamma$ being the adiabatic index. Here, we neglect magnetic fields, thermal conduction, etc., as they do not directly couple to dust, and our dust fluid module is fully compatible with these existing features.

We treat dust fluids as neutral, but future extensions may incorporate dust charge, e.g., Hopkins & Squire (2018).
We incorporate gas viscosity, which mimics the presence of external turbulence, described by the viscous stress tensor \( \Pi_v \):

\[
\Pi_v \equiv \rho_v \left( \frac{\partial v_{g,i}}{\partial x_{g,j}} + \frac{\partial v_{g,j}}{\partial x_{g,i}} - \frac{2}{3} \delta_{ij} \nabla \cdot \mathbf{v}_g \right),
\]

where \( \nu_g \) is the kinematic viscosity. Closely related to the gas viscosity is a dust diffusivity \( D_{d,n} \), which leads to concentration diffusion. We treat the diffusivity for each dust species as a free parameter to be specified by the user, while they are usually prescribed as (Youdin & Lithwick 2007):

\[
D_{d,n} = \frac{\nu_g}{1 + \left( T_{g,\text{eddy}}^2 / T_{g,\text{eddys}}^2 \right)^{1/2}},
\]

where \( T_{g,\text{eddy}} \) is the turbulent eddy time of the external turbulence. With this, the dust concentration diffusion flux, acting on the dust continuity equation, is given by

\[
\mathcal{F}_{\text{dif},n} \equiv -\rho_g D_{d,n} \nabla \left( \frac{\rho_{d,n}}{\rho_g} \right) = \rho_{d,n} \mathbf{v}_{d,n} \cdot \nabla, \tag{8}
\]

which also gives the definition of the effective dust drift speed \( \mathbf{v}_{d,n} \), due to concentration diffusion. Associated with this concentration diffusion, correction terms must be incorporated into the dust momentum equation to ensure consistent momentum diffusion flux (Tominaga et al. 2019) and Galilean invariance. The individual components of the momentum diffusion flux tensor are given by

\[
\Pi_{\text{dif},n,i,j} = v_{d,n,j} \mathcal{F}_{\text{dif},n,i} + v_{d,n,i} \mathcal{F}_{\text{dif},n,j}. \tag{9}
\]

Full derivations of the concentration diffusion terms will be presented in Section 2.2.

The last terms of the right-hand sides of the momentum Equations (2) and (5) correspond to the aerodynamic drag between gas and dust. Here, we assume linear drag law, where \( T_{e,n} \) is independent of velocity. An additional two source terms are added to the energy Equation (3), which correspond to the work done by the drag and to frictional heating. We have included a parameter \( \omega \) to control the level of frictional heating, with 0 being turned off and 1 being when all the dissipation is deposited to the gas.4

Other external source terms are denoted by \( f_{\text{src}} \), which may include stellar and/or planetary gravity in disk problems, depending on applications. They are implemented as explicit source terms added on to the momentum Equations (2) and (5), following the standard in Athena++ (Stone et al. 2020). Associated with them is a source term \( W \equiv f_{\text{src}} \cdot \mathbf{v}_g \) in the energy equation, accounting for the work done by the source terms.

Note that our formulation does not contain an energy equation for dust, thus it does not ensure global energy conservation of the composite dust–gas system. While this is not of overwhelming concern in typical applications, future generalization to incorporate a dust energy equation is possible. At algorithmic level, we thus aim at full momentum conservation, and implement energy source terms to match the overall accuracy of the algorithm.

\[\text{In reality, some of the dissipation must lead to the heating of the dust. If assuming that gas and dust should maintain the same temperature, one should assign } \omega = \rho_g \nu_i^2 / (c_{v,d} \rho_g + c_{v,d} \rho_d), \text{ where } c_{v,g} \text{ and } c_{v,d} \text{ are the heat capacity of the gas and dust, respectively, and } \rho_g = \sum_{i=1}^{n} \rho_{g,i}.\]

2.2. Consistent Formulation of Dust Concentration Diffusion

Here we derive dust fluid equations in the presence of turbulent diffusion, following the procedures of Cuzzi et al. (1993) and Tominaga et al. (2019). We use the Reynolds averaging technique with approximate closure relations, to properly account for the role of turbulence at subgrid level, while preserving global conservation laws. In doing so, we are interested in the physics on time (and potentially length) scales, above those for the turbulence, and hence any physical variable \( A \) is decomposed into a time-averaged part \( \overline{A} \) and a fluctuating part \( \Delta A \), i.e., \( A = \overline{A} + \Delta A \).

Without loss of generality, we focus on a single dust species and drop its label \( n \). We start from the standard dust fluid equations in conservation form:

\[
\frac{\partial \rho_d}{\partial t} + \nabla \cdot (\rho_d \mathbf{v}_d) = 0,
\]

\[
\frac{\partial \rho_d \mathbf{v}_d}{\partial t} + \nabla \cdot (\rho_d \mathbf{v}_d \mathbf{v}_d) = \rho_d \mathbf{v}_d - \mathbf{v}_d \cdot \mathbf{a}_g. \tag{10}
\]

Taking averages to the continuity equation, we obtain

\[
\frac{\partial \rho_d}{\partial t} + \nabla \cdot (\rho_d \mathbf{v}_d) + \nabla \cdot (\rho_d \mathbf{v}_d) = 0. \tag{11}
\]

The extra term \( \nabla \cdot (\rho_d \mathbf{v}_d) \), by definition, corresponds to the dust concentration diffusion flux (8):

\[
\nabla \cdot (\rho_d \mathbf{v}_d) = \mathcal{F}_{\text{dif}} = \rho_d \mathbf{v}_{d,n} \cdot \nabla. \tag{12}
\]

Next, by taking the averages to the momentum equation, we obtain

\[
\frac{\partial \rho_d \mathbf{v}_d}{\partial t} + \frac{\partial \Delta \rho_d \Delta \mathbf{v}_d}{\partial t} + \frac{\partial }{\partial x_i} (\rho_d \mathbf{v}_d \cdot \mathbf{v}_d) + \rho_d \mathbf{v}_d \cdot \mathbf{a}_g = \frac{\rho_d \mathbf{v}_d - \mathbf{v}_d}{T_s} \tag{13}
\]

At this stage, it is often argued that one can drop the second term on the left, assuming that the time-dependent diffusion flux is small compared to that of the bulk flow (Cuzzi et al. 1993; Tominaga et al. 2019). However, our analysis shows that this would violate Galilean invariance (see Appendix A and also the numerical tests in Section 2.4.2), and hence it must be kept. The second and third terms in the momentum flux can be reduced using the effective dust drift velocity \( \mathbf{v}_{d,n} \), which leads to the expression of momentum diffusion flux (9).

For the last term in the momentum flux \( \rho_d \Delta \mathbf{v}_{d,i} \Delta \mathbf{v}_{d,j} \), we may use the simple closure relation by Shariff & Cuzzi (2011) and Tominaga et al. (2019), as

\[
\Delta \mathbf{v}_{d,i} \Delta \mathbf{v}_{d,j} = \delta_{ij} c_{v,d}^2, \tag{14}
\]

where \( c_{v,d} \) is the effective dust sound speed. This term can be neglected in the multifluid approach (Garaud et al. 2004). The second term on the right-hand side is also neglected, with the
expectation that the standard drag term dominates, as in Tominaga et al. (2019).

With all these considerations, we recover the dust momentum equation shown in Section 2.1, which is here rewritten as

$$
\frac{\partial \rho_d (v_{d,j} + v_{d,\text{diff}})}{\partial t} + \frac{\partial}{\partial x_i} (\rho_d v_{d,i} v_{d,j} + \rho_d v_{d,\text{diff}} v_{d,j}) + \rho_d v_{d,i} v_{d,\text{diff}} = \frac{\rho_d v_{d,j} - v_{d,j}}{T_d},
$$

where, for notational convenience, we can drop the overline and interpret the dust fluid quantities in the averaged sense. The presence of the time derivative on $\rho_d v_{d,\text{diff}}$ in the momentum equation is the inevitable consequence of this averaging procedure. Missing this term would lead to unphysical behaviors, as we demonstrate in Section 3.2. Implementing this term also requires special care, as will be discussed in Section 2.4.2.

### 2.3. Dust–Gas Drag Integrators

The drag term involves the interactions between gas and all dust species. As a special source term for both gas and dust, the drag integrator aims to solve the following equation:

$$
\frac{\partial M}{\partial t} = [\sum_{n=1}^{N_g} \alpha_n (M_{g,n} - \epsilon_n M_{g})] \equiv f_{\text{drag}} (M, W),
$$

where $f_{\text{drag}}$ is the mutual drag force and $M = [M_g, M_{g,1}, \ldots, M_{g,n}] = [\rho_g v_g, \rho_1 v_1, \ldots, \rho_n v_n]$ is the momentum vector of the gas and dust. The remaining variables are denoted as $W$, given by $(\epsilon, \alpha)$, where $\epsilon = [\epsilon_1, \ldots, \epsilon_n] \equiv [\rho \partial_1 / \rho g, \ldots, \rho \partial_n / \rho g]$ and $\alpha = [\alpha_1, \ldots, \alpha_n] \equiv [T_{s,1}, \ldots, T_{s,n}]$. They are treated as constant parameters in the integrator.

The drag term is potentially stiff in two regimes. First, when the dust stopping time $T_d$ is very small, and stiffness arises when $T_d < \Delta t \equiv h$, the hydrodynamic time step. Second, when $\sum_{n=1}^{N_g} \alpha_n \gg 1$, which arises when the dust is strongly concentrated. The stiff regimes should be handled by fully implicit integrators, for stability. We note that for particle-based methods, handling the first regime of stiffness is relatively straightforward (Bai & Stone 2010c; Fung & Muley 2019; Mignone et al. 2019), whereas handling the second regime requires extra care, as one either artificially reduces the particle back-reaction (Bai & Stone 2010c) or sacrifices the time step (Li & Youdin 2021), and more rigorous treatment demands substantially more computational cost (Yang & Johansen 2016). With the fluid treatment of dust, one can directly solve the above equation implicitly, which automatically handles both stiffness regimes (Benítez-Llambay et al. 2019). In doing so, we need to evaluate the Jacobian of $f_{\text{drag}}$, and for brevity we drop the subscript “drag”:

$$
\frac{\partial f}{\partial M} = \begin{bmatrix}
-\sum_{n=1}^{N_g} \epsilon_1 \alpha_1 & -\alpha_1 & 0 & \cdots & 0 \\
\epsilon_1 \alpha_1 & -\alpha_1 & 0 & \cdots & 0 \\
\epsilon_2 \alpha_2 & 0 & -\alpha_2 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\epsilon_n \alpha_n & 0 & 0 & \cdots & -\alpha_n \\
\end{bmatrix}.
$$

Note that for the linear drag law, this Jacobian applies to individual dimensions (which are independent of each other).

We have implemented a number of different drag integrators. Here, we describe the algorithms of fully implicit integrators that we develop for achieving numerical stability and toward higher-order accuracy. The implementations of other simpler integrators, which are explicit or semi-implicit, and are useful for nonstiff problems, are described in Appendix C.

#### 2.3.1. First-order Fully Implicit Method

We start from the standard backward Euler method, which is a single-stage integrator to be combined with the Runge-Kutta 1 (RK1, or forward Euler) time integrator in Athena++. Integrating from step $n$ to $n+1$, the format is given by

$$
M^{(n+1)} = M^{(n)} + hf(M^{(n+1)}, W^{(n)}).
$$

Note that this numerical format guarantees momentum conservation. Substituting

$$
f(M^{(n+1)}, W^{(n)}) = f(M^{(n)}, W^{(n)}) + \frac{\partial f}{\partial M} (M^{(n+1)} - M^{(n)}),
$$

we can update the momentum $\Delta M \equiv M^{(n+1)} - M^{(n)}$ by

$$
\Delta M = \left(1 - h \frac{\partial f}{\partial M} \right)^{-1} hf(M^{(n)}, W^{(n)}),
$$

where $I$ is the identity matrix and evaluating $\Delta M$ involves matrix inversion. This is the main integrator implemented in FARGO3D (Benítez-Llambay et al. 2019), which makes the mutual drag interaction unconditional stable, despite only being first-order accurate in time. With the simple form of the Jacobian (17), the matrix in the backward Euler method can be solved efficiently on the order $\sim O(N_g)$, instead of $\sim O(N_g^2)$, as in standard lower–upper (LU) decomposition (Krapp & Benítez-Llambay 2020).

The energy source term on the gas has two parts. The first arises from the work done by the drag force. To better preserve energy conservation, this term should be implemented as the change in the gas kinetic energy due to gas drag:

$$
\Delta E_{g,1} = \Delta M_g \cdot \frac{(v_{g}^{(n)} + v_{g}^{(n+1)})}{2}/2.
$$

The second part is from frictional heating (Marble 1970; Laibe & Price 2014; Mignone et al. 2019), which is associated with the reduction of the total kinetic energy in the gas–dust system. This can be calculated by

$$
\Delta E_{g,2} = \Delta M_g \cdot \frac{v_{g}^{(n)} + v_{g}^{(n+1)}}{2} + \sum_{n=1}^{N_g} \Delta M_{d,n} \cdot \frac{v_{d,n}^{(n)} + v_{d,n}^{(n+1)}}{2}.
$$

### References

- Tominaga et al. (2019)
- Li & Youdin (2021)
- Bai & Stone (2010c)
- Fung & Muley (2019)
- Mignone et al. (2019)
- Laibe & Price (2014)
- Mignone et al. (2019)
- Marble (1970)
- Li & Youdin (2021)
- Bai & Stone (2010c)
- Fung & Muley (2019)
- Mignone et al. (2019)
- Yang & Johansen (2016)
- Benítez-Llambay et al. (2019)
The source terms for the energy equation should thus be
\[ E_{g}^{(n+1)} = E_{g}^{(n)} + \Delta E_{g,1} - \omega \Delta E_{g,2}. \]  

(23)

2.3.2. Second-order Fully Implicit Methods

Next, we build two fully implicit drag integrators to be combined with the van Leer 2 (VL2) and the Runge–Kutta 2 (RK2) time integrators in Athena++. We refer to them as the “VL2-Implicit” and the “RK2-Implicit” integrators, respectively. Both integrators involve two stages. Here, we describe their implementation, while the derivation of the algorithm can be found in Appendix B.

2.3.3. VL2-Implicit

Stage I: we apply the backward Euler method to update the system momenta from step \( n \) to half a time step \( h/2 \), denoted by a prime \( \prime \):

\[ \Delta M' = \left( 1 - \frac{h}{2} \frac{\partial f}{\partial M} \right)^{-1} \frac{h}{2} f(M^{(n)}, W^{(n)}), \]

(24)

Matrix inversion at this stage can be similarly achieved on order \( O(N_{p}) \). The update in gas energy at this stage is exactly analogous to that in the backward Euler method, which we do not repeat.

Stage II: the momentum at stage \( n \) to \( n+1 \) using the following:

\[ \Delta M = \Lambda^{-1} \left[ \left( 1 - \frac{h}{2} \frac{\partial f}{\partial M} \right)^{-1} \frac{h}{2} f(M^{(n)}, W^{(n)}) \right], \]

(25)

where

\[ \Lambda \equiv 1 - \left( 1 - \frac{h}{2} \frac{\partial f}{\partial M} \right)^{\prime} \frac{h}{2} \frac{\partial f}{\partial M}^{\prime}. \]

(26)

Note that this matrix is more complex and should be inverted by LU decomposition (Press et al. 1986). The update in gas energy has exactly the same form as Equations (21) to (23), which we do not repeat.

2.3.4. RK2-Implicit

Stage I: we use the backward Euler method with time step \( h \) to calculate the momentum at step \( n+1 \), which is exactly the same as described in Section 2.3.1. We still denote the quantities at the end of this stage using a prime \( \prime \).

Stage II: the momentum at stage \( n+1 \) is:

\[ \Delta M = \Lambda^{-1} \left[ b f(M^{(n)}, W^{(n)}) \right. \]

\[ + \left( 1 - \frac{h}{2} \frac{\partial f}{\partial M} \right)^{\prime} \frac{h}{2} f(M^{(n)}, W^{(n)}) \right], \]

(27)

where

\[ \Lambda \equiv 1 - \left( 1 - \frac{h}{2} \frac{\partial f}{\partial M} \right)^{\prime} \frac{h}{2} \frac{\partial f}{\partial M}^{\prime}. \]

(28)

Similarly, matrix inversion is solved by LU decomposition. The update in gas energy also has exactly the same form as Equations (21) to (23), which we do not repeat.

Note that this integration scheme is in essence the same as the fully implicit particle integrator in Bai & Stone (2010c).

2.3.5. Coupling Explicit Hydrodynamic Integrators with Implicit Drag Integrators

Special care must be taken when combining implicit integrators with explicit hydrodynamic integrators and source terms. When they are treated separately, the combined algorithm would only be first-order accurate, and the implicit drag integrator cannot maintain exact equilibrium solutions.

To overcome these issues, we may consider advection, diffusion, and other hydrodynamic source terms as “add-ons” to \( f \). In other words, \( f \) in the drag algorithms above represents the combination of the drag force \( f_{\text{drag}} \), treated implicitly, as well as other explicit terms, including advection and other source terms \( G_{M} \), acting on the gas and dust momenta:

\[ f \equiv f_{\text{drag}}(M, W) + G_{M}(U), \]

(29)

where \( G_{M} \) is expressed in terms of conserved variables \( U \). By adding an overline on \( U \), we treat these other explicit terms as known constants, readily obtained in the hydrodynamic integrator. In the hydrodynamic integration from step 1 to step 2 over time interval \( \Delta t \), we estimate \( G_{M} \) to be the momentum update from the explicit terms:

\[ G_{M}(U) = \frac{M^{(2)} - M^{(1)}}{\Delta t}, \]

(30)

where \( M^{(2)}, M^{(1)} \) represent the momenta before and after the explicit integration steps (advection, diffusion, and other explicit source terms). By treating this term as a constant, the Jacobian and the \( \Lambda \) matrices described in the previous subsection remain unchanged.

Implementing the above requires extra storage to store the momentum updates and that we must finish all explicit steps in the hydrodynamic integration before entering the drag integrator. We will show that our approach successfully achieves second-order accuracy when using VL2-Implicit and RK2-Implicit integrators, and that it also allows us to achieve exact equilibrium solutions involving the drag force.

2.4. Integration of Multifluid Dust Equations

The integration of dust fluid is divided into several parts (advection, diffusion, source terms, and drag). Except for the drag term (described in the previous subsection), the other terms are treated independently and explicitly, and we describe their implementation in this subsection.

2.4.1. General Procedures

Following the standard routine in Athena++, each integration time step is divided into a number of stages, depending on the time integrator employed (see the detailed descriptions in Stone et al. 2020). At each stage, the integration procedures involve updating conserved variables based on primitive variables, by evolving the fluid equations by \( dt \). Our multifluid dust module supports Athena++ time integrators up to second order, including first-order Runge–

The implicit–explicit Runge–Kutta schemes are viable choices (Pareschi & Russo 2005); they usually involve more stages of integration than the order of accuracy achieved, and do not necessarily match the existing hydrodynamic integrators in Athena++.
Kutta (RK1), second-order Runge–Kutta (RK2), and the van Leer integrator (VL2).

For each dust species, the primitive \( (W_d) \) and conserved \( (U_d) \) variables are

\[
W_d = \begin{bmatrix} \rho_d \\ \mathbf{v}_d \end{bmatrix}, \quad U_d = \begin{bmatrix} \rho_d \\ \rho_d (\mathbf{v}_d + \mathbf{v}_{d,dif}) \end{bmatrix}.
\]

Note that the presence of the time derivative on \( \rho_d \mathbf{v}_{d,dif} \) in the momentum equation suggests that the concentration diffusion flux should be considered as part of the conserved dust momentum. The total momentum is thus \( \rho \mathbf{v}_d + \sum \rho_{d,i} (\mathbf{v}_{d,i} + \mathbf{v}_{d,dif,i}) \).

Integrating the bulk part of the dust fluid is very similar to that for hydrodynamics in Athena++. The main procedure involves the reconstruction of primitive variables at cell interfaces, followed by solving a Riemann problem to obtain the mass and momentum fluxes, after which we update the dust fluid quantities from the flux gradients. As in Athena++, the multifluid dust module supports spatial reconstructions up to third order.

As pressureless fluids, the Riemann problem for dust fluids is greatly simplified. In one dimension along the \( x \)-direction, given the left/right states \( W_{d,l/r} \), we provide the Riemann flux for conserved variables as follows. The density flux reads:

\[
F_\rho(\rho_d) = \begin{cases} 
\rho_d \mathbf{v}_{d,l} & \mathbf{v}_{d,l} > 0, \mathbf{v}_{d,r} > 0, \\
\rho_d \mathbf{v}_{d,r} & \mathbf{v}_{d,l} < 0, \mathbf{v}_{d,r} < 0, \\
0 & \mathbf{v}_{d,l} < 0, \mathbf{v}_{d,r} > 0, \\
\rho_d \mathbf{v}_{d,l} + \rho_d \mathbf{v}_{d,r} & \mathbf{v}_{d,l} > 0, \mathbf{v}_{d,r} < 0.
\end{cases}
\]

Similar expressions hold for the momentum flux for all three directions. Essentially, we use the upwind flux when the normal velocities in the left/right states are the same, set the flux to be zero when the left/right normal velocities diverge, and sum up the fluxes from the two sides when the left/right normal velocities converge. The last treatment reflects that, as pressureless fluids, the flows on the two sides can penetrate each other, just as particles do.\(^6\)

The implementations of other source terms on dust, such as stellar gravity and source terms in a shearing box, as well as geometric source terms in cylindrical and spherical coordinates, are the same as that for gas, which is treated explicitly.

### 2.4.2. Dust Diffusion

The implementation of dust concentration diffusion starts by computing the concentration diffusion flux according to Equation (8). The fluxes are computed by standard finite differencing, and are located at the cell interfaces. Next, we calculate the momentum diffusion flux according to Equation (9). This term contains two parts. The first part, \( \mathbf{v}_{d,n,i} \mathbf{F}_{d,dif,n,i} \), describes the diffusion of the \( j \)-momentum in the \( i \)-direction. At the implementation level, its value is obtained by averaging from the upwind side, based on the sign of the concentration diffusion flux \( \mathbf{F}_{d,dif,n,i} \). The second part, \( \mathbf{v}_{d,n,i} \mathbf{F}_{d,dif,n,j} \), represents the advection of the \( j \)-diffusion flux in the \( i \)-direction. Its value is obtained by averaging from the upwind side, based on the sign of the advection velocity \( \mathbf{v}_{d,n,i} \).

In addition, we note that in cylindrical/spherical coordinates, we need to add extra diffusive geometric sources terms on the momentum and energy equations (Skinner & Ostriker 2010).

Finally, we compute the concentration diffusion momenta and compare to the original concentration diffusion flux, from which we can estimate the contribution from the \( \partial (\rho_d \mathbf{v}_{d,dif,n,i}) / \partial t \) term. The concentration diffusion momenta are stored in the cell centers and are averaged by the nearby face-centered concentration diffusion fluxes. We note that although our formulation is Galilean invariant, it is not invariant to machine precision at the implementation level.

---

\(^6\) Note that penetration is still prohibited within each cell, in which the dust fluid velocities get well mixed. Alternatively, one may set the flux to zero in this case. We do not find many practical differences in test problems by using different Riemann solvers for dust.
but the incorporation of this term is important for ensuring approximate Galilean invariance in simulations.

2.5. Flow Chart

Figure 1 shows the flow chart of our multifluid dust module in Athena++, and we summarize the main steps over one integration stage below.

Step 1: backup the primitive variables for both gas and dust and calculate the dust stopping time. The backed-up primitive variables are used in the semi-implicit and fully implicit drag integrators to ensure the higher-order accuracy of the combined algorithm, as discussed in Section 2.3.5.

Step 2: calculate the diffusion processes of gas and dust, when applicable, including viscosity, thermal conduction, and resistivity on the gas, and concentration diffusion and momentum correction on the dust fluids.

Step 3: calculate the Riemann fluxes of both gas and dust, and integrate the gas and dust fluids by applying flux divergence. Send and receive flux corrections when necessary for mesh refinement.

Step 4: add explicit source terms on gas and dust, including geometric source terms for curvilinear coordinates.

Step 5: apply any of the drag integrators, and use the backed-up variables to enhance the accuracy in implicit schemes.

Step 6: do orbital advection when necessary (for disk problems).

Step 7: send and receive boundary data, set boundary conditions, and do prolongation/restriction for mesh refinement.

Step 8: convert conserved variables to primitive variables. When the dust momentum correction is turned on, the concentration diffusion flux calculated by step 2 will be subtracted from the dust momenta.

After finishing all stages of an integration cycle, we calculate the new time step based on the Courant–Friedrichs–Lewy (CFL) condition for both gas and dust. The dust CFL condition is set according to the maximum dust velocity and dust diffusion coefficient $D_d$, in the same way as the gas velocity and viscosity. As a dust fluid module, it has a fixed amount of floating-point operations per meshblock per integration cycle, as opposed to particle-based approaches. Taking advantage of the task-based execution model, with the excellent scalability of Athena++, our dust fluid module primarily adds a fixed fraction of computational cost. Such cost increases with $N_d$ nonlinearly when using higher-order fully implicit drag solvers, due to matrix inversion, the cost of which scales as $O((N_d + 1)^3)$. In practice, we find that linear scaling approximately applies for $N_d \lesssim 5$, and the cost of the drag solver is no more than the cost of the rest of the dust integration scheme for $N_d \lesssim 10$. Further details about code performance are provided in Appendix D.

3. Code Tests

In this section, we show benchmark numerical tests of our multifluid dust module. They include the collisions between gas and dust, dust diffusion with momentum correction, linear/ nonlinear SI, and (static/adaptive) mesh refinement. We also follow the same dusty sound wave and dusty shock tests in Sections 3.2 and 3.3 of Benítez-Llambay et al. (2019). To avoid repetitions, we show the test results of the dusty sound wave and dusty shock in Appendices F.1 and F.2. They demonstrate that our multifluid dust code achieves full second-order accuracy when coupled with hydrodynamics, and that it is excellent at shock capturing.

3.1. Collisions

We start by conducting the 1D dust–gas collision test as a benchmark, similar to Section 3.1 of Benítez-Llambay et al. (2019). We consider two dust species with constant stopping times $T_{d,1}$, $T_{d,2}$, and set three collision tests named A, B, and C. The gas and all the dust species are homogeneous, each having its own density $(\rho_g, \rho_{d,1}, \rho_{d,2})$ and velocity $(v_g, v_{d,1}, v_{d,2})$. The system then evolves under the mutual aerodynamic drag forces, characterized by two eigenvalues $\lambda_1, \lambda_2$, in the form of

$$v = v_{\text{COM}} + c_1 \exp(\lambda_1 t) + c_2 \exp(\lambda_2 t),$$

where $v_{\text{COM}}$ is the center-of-mass velocity of the system. Their initial conditions, as well as the associated coefficients and eigenvalues, are given by shown in Table 1, and we provide the calculation procedures in Appendix E. The three tests are designed to test the nonstiff case (Test A), the stiff case with small stopping time (Test B), and the stiff case with large dust-to-gas ratios (Test C). These tests are conducted in 1D Cartesian coordinates with a periodic boundary condition. We use the adiabatic equation of state, with the adiabatic index being $\gamma = 1.4$, and an initial gas sound speed is set as $c_s^2 \equiv \gamma \rho_{\text{gIni}} = 1.4$ for all three tests. We include the work and friction heating from the drags in the energy equation. We test eight drag integrators (explicit: “RK1-Explicit,” “RK2-Explicit,” and “VL2-Explicit”; semi-implicit: “Trapezoid” and “TrBDF2”; and fully implicit: “RK1-Implicit,” “RK2-Implicit,” and “VL2-Implicit”), and the main results are discussed below.

The top eight panels of Figure 2 show the temporal evolution of the velocities and energy of both the gas and dust in Tests B and C, with the numerical time steps $\Delta t = 0.005$ and 0.05, respectively. The results are to be compared with the analytic solution. Table 1 shows the largest eigenvalue, $|\lambda_2| \approx 1058$ and 106 for Tests B and C. Therefore, the drag terms become stiff when $\Delta t > 1/|\lambda_2| \approx 0.001$ and 0.01 for B and C. The explicit integrators are unstable in Tests B and C, with $\Delta t = 0.005$ and 0.05. The semi-implicit methods are stable in the stiff drags, but the numerical updates oscillate around the analytic solutions artificially, which is not unexpected, as similar behavior was observed in Bai & Stone (2010c). Our fully implicit methods, “RK1-Implicit,” “VL2-Implicit,” and “RK2-Implicit,” handle these two stiff regimes (small stopping times and large dust-to-gas ratios) very well, and the two second-order integrators are clearly seen to be more accurate.

To test the numerical convergence, we calculate the relative error $\Delta E$ as a function of the numerical time step $\Delta t$, with different drag integrators. The $\Delta E$ is calculated by

$$\Delta E(\Delta t) = \frac{1}{t_{\text{max}} - t_{\text{min}}} \sum \frac{[U_{\text{num}}(\Delta t) - U_{\text{ana}}]}{U_{\text{ana}}} \Delta t,$$  

where $U$ represents the momentum or the gas energy, and the subscripts “num” and “ana” represent the numerical and analytic solutions, respectively. The scaling of the total relative error $\Delta E_{\text{total}} = \Delta E_{\text{num}} + \Delta E_{\text{erg}}$ with different time steps $\Delta t$ is shown in the bottom three panels of Figure 2 for different drag
integrators. We vary $\Delta t$ from $10^{-4}$ to $10^{-1}$, which are nonstiff for Test A, but get increasingly stiff for Tests B and C. The total relative errors are calculated between $t_{\text{min}} = 0$ and $t_{\text{max}} = 10$.

We see that in the nonstiff regime of Test A, all the drag integrators achieve first- or second-order accuracy in time, as desired, and there is no significant difference between integrators of the same order. In the stiff regime of Tests B and C, we see that the errors in the explicit and semi-implicit integrators diverge when $\Delta t \gtrsim (1/|\lambda_2|)$. The threshold for error divergence is only slightly higher for semi-implicit integrators. The fully implicit integrators, on the other hand, achieve the desired level of accuracy at small $\Delta t$, while remaining stable at large $\Delta t$ for both tests. Among them, the second-order fully implicit integrators show error levels that are at least one order of magnitude smaller, and are the preferred choices that we generally recommend.

Our drag integrators conserve the total momentum of the dust–gas system by construction. While not shown in the figures, we have verified that in all three tests, the total momentum is conserved to the fractional level of $\lesssim 10^{-14}$ (i.e., approaching machine precision), within the duration of the simulations.

### 3.2. Momentum Correction in Dust Diffusion

Novel additions in our dust concentration diffusion formulation include the time derivative of the $\rho_d \nu_d,\diff$ term and the $\Pi_t,\diff$ in the dust momentum Equation (5). We refer to them as “momentum correction.” They reflect the facts that the concentration diffusion flux carries momentum that backreacts to the gas and that the new formulation is Galilean invariant. To test these aspects of our implementation, we consider the following simple test problems without/momentum correction in 1D, 1.5D, and 2D Cartesian coordinates, where 1.5D means a 1D test in the presence of a transverse velocity.

#### 3.2.1. Initial Setsups

We give the parameters of our tests in Table 2, with more details below. The 1D and 1.5D tests are carried out in Cartesian coordinates, with 256 uniform grids, covering $x \in [0, 20]$. The gas has a uniform initial density $\rho_{g0} = 1$, and a single dust species with a constant stopping time $T_s = 10^{-2}$ is included with an initial Gaussian density profile:

$$\rho_{d0} = A \exp \left[-\frac{(x - x_0)^2}{2\sigma_x^2}\right] + \rho_{g0},$$

where $A = 5$, $x_0 = 10$, and $\sigma_x = 2$ in both cases. We include gas viscosity and dust diffusion with the coefficients $\nu = D_{\nu} = 1$. In the 1D test, the initial gas and dust velocities are zero, whereas in the 1.5D test, the gas and dust have the constant transverse velocities $v_{g,x} = v_{d,x} = 1$.

In the 2D test, we use 2D Cartesian coordinates in the domain $x, y \in [0, 20]$, with 256$^2$ cells, and we set the initial 2D Gaussian dust density profile in the center of the box:

$$\rho_{d0,2D} = A \exp \left[-\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2}\right] + \rho_{g0},$$

where $y_0 = 10$ and $\sigma_y = 2$, and the rest of parameters are same as in the 1.5D test.

We note that when including momentum correction, there is an additional contribution $\nu_d,\diff$ to the conserved dust momentum. This leads to two possible initial settings. One is to make the initial conserved momentum zero. By the conversion relation (31), the primitive velocity thus equals $- \nu_d,\diff$. Alternatively, one can choose the primitive velocity to be zero (i.e., zero mean dust velocity), so that the conserved momentum becomes $\rho_v,\diff$ (i.e., nonzero mean dust momentum). This ambiguity reflects the initial condition itself being physically unrealistic to build up, without involving additional source terms. As a test problem, we choose the latter, which we consider to be physically more natural and intuitive (the
alternative choice would lead to different interpretable outcomes that we omit here for brevity). Note that without momentum correction, we set $v_d = 0$, so that the two setups share the same initial conditions of primitive variables.

In all these tests, dust back-reaction is included. We use an isothermal equation of state with sound speed $c_{s,iso} = 1$, applying periodic boundary conditions. We use the piecewise linear method (PLM) spatial reconstruction for both gas and dust, and a CFL number of 0.3. The “VL2-Implicit” drag integrator is used to calculate the mutual drags.

3.2.2. Results

In the 1D tests shown in the top eight panels of Figure 3, we note that the correction term $\Pi_{dif}$ is zero, thus only the time derivative term $\frac{\partial (\rho u_n u_n)}{\partial t}$ is effective. When the correction is not included, dust proceeds as normal concentration diffusion, whereas the gas is totally intact. This is unphysical, because turbulent mixing is a two-way process that not only mixes dust with gas, but should also mix gas with dust. When the momentum correction is included, we see that the gas density...
exhibits a central deficit and two outside bumps. This is essentially the outcome of the gas being dragged by the outward diffusion flow of dust, as can be seen in the central region of the middle panel. The additional structures in the gas act to slow down the dust concentration diffusion, and we can see that without incorporating momentum correction, dust diffuses more than two times more rapidly.

When adding a transverse velocity in the 1.5D test, one should expect identical results as in the 1D test, except for a velocity shift in the y-direction. However, without momentum correction, we see in the two right panels at the bottom of Figure 3 that the system develops artificial variations in $v_y$. This is because concentration diffusion changes the dust density profile, but without properly altering the momentum profile. When momentum corrections are included, we see that the dust and gas momenta are properly advected to ensure Galilean invariance.

The Astrophysical Journal Supplement Series, 262:11 (25pp), 2022 September Huang & Bai

**Table 2**

| Correction | $v_{g,x}$ | $v_{g,y}$ | $v_{d,x}$ | $v_{d,y}$ |
|------------|-----------|-----------|-----------|-----------|
| 1D         | No        | 0         | 0         | 0         |
|            | Yes       | 0         | 0         | $v_{d,dif,x}$ |
| 1.5D       | No        | 0         | 1         | 0         |
|            | Yes       | 0         | 1         | $v_{d,dif,x}$ |
| 2D         | No        | 1         | 1         | 1         |
|            | Yes$^a$   | 1         | 1         | $1+v_{d,dif,x,2D}$ |

**Notes.**

$^a$ The correction terms by default include both $\Pi_{dif}$ and $\partial(\rho d v_{d,dif})/\partial t$.

$^b$ The diffusion velocity $v_{d,dif}$ is calculated by Equation (8).

$^c$ In the 2D cases, we have two corrections tests: one is with only $\Pi_{dif}$ and the other is with both $\Pi_{dif}$ and $\partial(\rho d v_{d,dif})/\partial t$.
Figure 4. 2D dust diffusion tests without correction (top eight panels), with only $\Pi_{d/d}$ correction (middle eight panels), and with both $\Pi_{d/d}$ and $\partial(\rho_d\rho_{d,dif})/\partial t$ corrections (bottom eight panels). From left to right, the panels are at time $t = 0.0$, 1.0, 3.0, and 5.0. The first, third, and fifth rows are for the gas density ($\rho_g$), while the second, fourth, and sixth rows are for the dust concentration ($\rho_d/\rho_g$). The black (white) lines represent the velocity streamlines of gas (dust).
Therefore, both correction terms are essential for ensuring Galilean invariance. Overall, these tests demonstrate the importance of properly incorporating the momentum correction terms for a consistent treatment of dust concentration diffusion, especially when dust feedback is taken into account.

### 3.3. Streaming Instability

SI is a stringent test for two-way gas drag, and its nonlinear evolution with strong dust climbing up represents a further test of the code’s capability of handling sharp discontinuities. We adopt the linear tests given by Youdin & Johansen (2007) and the nonlinear runs from Johansen & Youdin (2007), which have become the standard test problems for codes with particle-based treatments of dust (Bai & Stone 2010c), and more recently for multi-fluid dust as well (Benítez-Llambay et al. 2019). We will further extend the nonlinear tests to cylindrical coordinates (Section 3.3.4) and incorporate mesh refinement (Section 3.4.1).

#### 3.3.1. Shearing Box Equations and Equilibrium State

Most of our SI tests are carried out in the local shearing box framework, which follows a local patch of a disk at some fiducial radius $r_0$ in the corotating frame with orbital frequency $\Omega_0 = \Omega(r_0)$ (Goldreich & Lynden-Bell 1965; Hawley et al. 1995). The equations of gas and dust are written in a Cartesian coordinate system $(x, y, z)$ for radial, azimuthal, and vertical directions, with Coriolis and centrifugal source terms (Stone & Gardner 2010). The unit vectors along these three directions are denoted as $(\hat{i}, \hat{j}, \hat{k})$. We do not consider viscosity, diffusion, magnetic field, and self-/vertical gravity. Adopting an isothermal equation of state with isothermal sound speed $c_s$, the continuity and momentum equations of gas and dust are:

\[
\frac{\partial \rho_g}{\partial t} + \nabla \cdot (\rho_g \mathbf{v}_g) = 0, \quad (37)
\]

\[
\frac{\partial (\rho_g \mathbf{v}_g)}{\partial t} + \nabla \cdot (\rho_g \mathbf{v}_g \mathbf{v}_g + \rho_g \mathbf{P}_g) = 2 \rho_g \eta \Omega_0^2 \hat{i} \times \mathbf{v}_g - \sum_{k=1}^{n} \rho_d \mathbf{v}_d - \rho_{d,k} T_{d,k}, \quad (38)
\]

\[
\frac{\partial \rho_{d,k}}{\partial t} + \nabla \cdot (\rho_{d,k} \mathbf{v}_{d,k}) = 0, \quad (39)
\]

\[
\frac{\partial (\rho_{d,k} \mathbf{v}_{d,k})}{\partial t} + \nabla \cdot (\rho_{d,k} \mathbf{v}_{d,k} \mathbf{v}_{d,k} + \rho_{d,k} \mathbf{P}_{d,k}) = 2 \rho_d \eta \Omega_0^2 \hat{i} \times \mathbf{v}_{d,k} - \rho_{d,k} \mathbf{v}_{d,k} - \rho_{d,k} T_{d,k}, \quad (40)
\]

where $q = -d \ln \Omega_0/\partial r$ is the shear rate and $q = 3/2$ for the Keplerian disks that we adopt. Ignoring vertical gravity (i.e., an unstratified disk), dust and gas can achieve the so-called Nakagawa–Sekiya–Hayashi (NSH) equilibrium (Nakagawa et al. 1986). The equilibrium is a force balance between background pressure gradient, centrifugal force, Coriolis force, and mutual aerodynamic drags between gas and dust in the horizontal plane. Here, we consider a Keplerian disk with angular speed $\Omega_0 = \Omega_K$. In the absence of dust, the gas rotates slower than the Keplerian speed by a small amount $\eta v_K \equiv 1/c_s$, due to the background pressure gradient, and $\Pi$ is $\lesssim 0.1$ under typical disk conditions. $\eta$ represents the strength of the radial gas pressure gradient:

\[
\eta \equiv \frac{1}{2} \frac{\ln P_r}{\ln r} \left( \frac{h}{r} \right)^2 = \frac{1}{2} \frac{\ln P_r}{\ln r} \left( \frac{c_s}{v_K} \right)^2. \quad (41)
\]

The original NSH solution considered a single dust species. Generalized to multiple dust species with different stopping times, the velocities of gas and dust in the multispecies equilibrium are given by Benítez-Llambay et al. (2019, and also see Tanaka et al. 2005):

\[
v_{g0,x} = 2 \eta v_K \frac{A}{A + B}, \quad v_{d0,k,x} = \frac{v_{g0,x} + 2 St_k v_{g0,y}}{1 + St_k^2}, \quad (42)
\]

\[
v_{g0,y} = -\eta v_K \frac{B}{A + B}, \quad v_{d0,k,y} = \frac{v_{g0,y} - St_k v_{g0,x}/2}{1 + St_k^2},
\]

where $St_k \equiv \Omega_K T_{d,k}$ is the dimensionless stopping time of the $k$th dust species, the velocities with a prime have a Keplerian shear subtracted $v_{g0,y} = v_{g0,y} + (3/2) \Omega_K x$, $v_{d0,k,y} = v_{d0,k,y} + (3/2) \Omega_K x$, and

\[
A = \sum_{k=1}^{n} \frac{\epsilon_k St_k}{1 + St_k^2}, \quad B = 1 + \sum_{k=1}^{n} \frac{\epsilon_k}{1 + St_k^2}. \quad (43)
\]

In our numerical setup, we add an additional outward force $f = 2 \rho_d \eta v_K \Omega_0 q$ on the gas, to mimic the radial pressure gradient. Note that this differs from Bai & Stone (2010c) and Benítez-Llambay et al. (2019), who add this force on the dust component. The two approaches are equivalent, except for a constant velocity shift. Realizing the exact analytic solution of the multispecies NSH equilibrium is straightforward when using explicit integrators. Our numerical implementation in Section 2.3.5 also ensures that such an exact solution can be realized using the semi-implicit and implicit drag integrators as well.

#### 3.3.2. Linear SI Modes and Growth Rates

The NSH equilibrium is subject to the SI (Youdin & Goodman 2005). The growth rate $\sigma$ of the SI is a function of the initial dust-to-gas ratio (or metallicity) $\epsilon_0 \equiv \rho_d 0/\rho_g 0$, the dimensionless stopping time $St$, and the two dimensionless wavenumbers $K_x \equiv k x \rho_0 = k v_K / \Omega_K$ and $K_z \equiv k z \rho_0 = k v_K / \Omega_K$; $s = (\epsilon_0, St, K_x, K_z)$. In this work, we choose the Lin-A and Lin-B tests in Table 1 of Youdin & Johansen (2007), which consist of gas and one dust species, and the Lin-3 test in Section 3.5 of Benítez-Llambay et al. (2019), which consists of gas and two dust species.

The numerical setups of the linear tests are similar to those of Youdin & Johansen (2007), Bai & Stone (2010c), and Benítez-Llambay et al. (2019). The numerical domain is a square box along the $x$- and $z$-directions, with $-L_x/2 < x < L_x/2$, $-L_z/2 < z < L_z/2$, and $L_x = L_z = 1$. We choose $\Omega_0 = 1.0$ and $\eta v_K = 0.05 c_s$. On top of the multispecies NSH equilibrium from Equation (42), we add perturbations of the following form on the densities and velocities of both gas and dust, following...
Younid & Johansen (2007):

\[ \delta \rho = [\Re(\tilde{\rho}) \cos \phi - \Im(\tilde{\rho}) \sin \phi] \cos(k_z z), \]
\[ \delta v_x = [\Re(\tilde{v}_x) \cos \phi - \Im(\tilde{v}_x) \sin \phi] \cos(k_z z), \]
\[ \delta v_z = [\Re(\tilde{v}_z) \cos \phi + \Im(\tilde{v}_z) \sin \phi] \sin(k_z z), \]

(44)

where \( \phi \equiv k_x x - \omega t \) and the density and velocity perturbations, denoted by \( \tilde{\rho} \) for gas and individual dust species, are given by the respective eigenvectors of the specific linear modes, which are listed in Table 1 of Youdin & Johansen (2007) and Table 4 of Benítez-Llambay et al. (2019).

In these tests, we fit one eigenmode in the simulation box. As our box size is fixed, the sound speed \( c_s \) is no longer a free parameter, and it depends on the dimensionless wavenumbers \( (K_x, K_z) \). Because of \( K_x = K_z = k_x \eta_0 = 2\pi \times 0.05 c_s / \Omega_0 \), we obtain \( c_s = K_x \Omega_0 / (0.05 \times 2\pi) \). In the Lin-A test, \( K_x = K_z = 30 \), thus we have \( c_s = 95.49296585 \). A similar setup can be done for the Lin-B and Lin-3 tests, with \( c_s = 19.09859317 \) in Lin-B and \( c_s = 159.1549431 \) in Lin-3. We use all the second-order drag integrators, with the PLM and piecewise parabolic method (PPM) for the spatial reconstructions on both the gas and dust, and the HLLE Riemann solver. The numerical resolution spans from 8 to 256 cells per wavelength in all directions, with the CFL number being 0.3.

We measure the growth rate of the kinetic energy \( E_{\text{kinetic}} \) of both the gas and dust in the linear tests. Because the initial kinetic energy is dominated by radial drift, we expect the temporal variation of the kinetic energy to grow as \( \delta E_{\text{kinetic}} \propto \exp(s t) \). We fit the spatial standard deviation \( \sqrt{\delta E_{\text{kinetic}}^2} \) over the whole mesh as a function of time, and we show the fitted growth rate \( s / \Omega_0 \) in Figure 5 as a function of resolution in the Lin-A, Lin-B, and Lin-3 tests. As these test problems are nonstiff, different drag integrators generally show similar results. With PPM reconstruction, 16 cells per wavelength is generally sufficient to accurately capture the growth in all three tests. When using the PLM reconstructions, on the other hand, about 128–256 cells per wavelength are needed for similar accuracy, and the requirement for Lin-B is the most stringent. When compared with the measured growth rates in Table 5 of Benítez-Llambay et al. (2019), it appears that we need more grid points to achieve similar accuracy as FARGO3D. On the other hand, the level of accuracy that we achieve is similar to those obtained in other finite-volume method codes, such as Athena (Bai & Stone 2010c) and PLUTO (Mignone et al. 2019), with PPM spatial reconstructions. We thus attribute the difference primarily to the different nature of the base code. On the other hand, we will show that our code shows similar outcomes in the nonlinear regime at a given resolution.

3.3.3. The SI in the Nonlinear Regime

The nonlinear SI runs are also carried out in the 2D shearing box in \( x-z \). We select two nonlinear tests of the SI, namely, the AB and BA tests from Johansen & Youdin (2007). In the AB test, the domain is \( -\eta_0 \leq x \leq \eta_0, -\eta_r \leq z \leq \eta_0 \), with the parameters being \( c_0 = 1.0, \, \Omega_0 = 0.1 \). In the BA test, the domain is \( -20 \eta_0 \leq x \leq 20 \eta_0, -20 \eta_r \leq z \leq 20 \eta_0 \), with the parameters being \( c_0 = 0.2 \) and \( \Omega_0 = 1.0 \). The parameters and simulation setups are similar to Table 1 of Johansen & Youdin (2007), Section 5 of Bai & Stone (2010c), and Section 3.5.6 of Benítez-Llambay et al. (2019). We use the “VL2-Implicit” drag integrator and the Roe Riemann solver, with PPM reconstruction for the gas and PPM reconstruction for the dust, in order to more robustly follow the dramatic variation of dust density over space in the nonlinear stage with strong particle clumping. We use an isothermal equation of state, where the sound speed is \( c_s = 1.0 \) and the initial gas density is \( \rho_0 = 1.0 \). We also set a density floor \( \rho_{\text{floor}} = 10^{-6} \) on dust. Gas viscosity and dust diffusion are not included. The simulations are initiated from the NSH equilibrium, on top of which we add white-noise velocity perturbations with an amplitude of \( \langle A \rangle \sim 0.02 c_s \) on both gas and dust.

The saturated states of the AB and BA tests, following the evolutions of \( 40 \Omega_0^{-1} \) and \( 800 \Omega_0^{-1} \), are shown in Figure 6, for simulations with different resolutions. They are to be compared with Figures 8 and 9 of Benítez-Llambay et al. (2019). The AB test is characterized by the development of thin filaments and cavitation toward smaller scales at higher resolution, well consistent with the results in previous works (Johansen & Youdin 2007; Bai & Stone 2010c; Benítez-Llambay et al. 2019). In the BA test, the system develops long dusty stripes and valleys nearly aligned with the \( z \)-direction and tilted toward the \( x \)-direction. Different from the
AB test, these general features are similar at all resolutions, again consistent with previous studies.

We further investigate the convergence of dust clumping by calculating the cumulative dust density distributions (CDFs). Following the same procedures as described in Section 3.5.6 of Benítez-Llambay et al. (2019), there are two ways of calculating the CDFs. One is based on the probability that the local dust density exceeds a certain threshold, obtained by counting the number of cells whose dust density exceeds the threshold. The other reflects the probability of a particle residing in regions whose particle density exceeds a certain threshold, obtained by weighting the first probability by the local dust density. We refer to the two CDFs as being obtained by counting cell numbers and by counting dust density, respectively. The overall results are shown in Figure 7.

The CDFs obtained by counting cell numbers are very similar to those obtained by Benítez-Llambay et al. (2019) for both the AB and BA tests. The CDFs of the AB test systematically vary with resolution at both the low-density and high-density ends, in line with the nonconvergent behaviors revealed in Figure 6. The CDFs of the BA test show convergence at the low-density end up to \( P \sim 10^{-3} \), but show more significant clumping at higher resolution (BA-1024\(^2\) and BA-2048\(^2\)).

Our CDFs by counting dust density, on the other hand, show more clumping than FARGO3D. The clumping is also more significant than that obtained using the particle module of Athena (see Figure 6 of Bai & Stone 2010c). This might be related to the higher-order drag integrators adopted here compared to FARGO3D. We also note that in the original Athena code, some artificial reduction of dust feedback was applied in strong dust clumps to alleviate the stiffness in the system, which is circumvented in our approach. We thus leave this as an open issue. We also anticipate that our dust fluid module generally finds most of the applications in regimes with \( St < 1 \), instead of \( St \gtrsim 1 \), as in the BA test.

We also examine the properties of gas turbulence triggered by the SI. We calculate the turbulent Mach numbers along three directions \((x, y, z)\), the Reynolds stress, and the mean radial drift velocities of dust fluids. The mach numbers are calculated by \( Ma = \sqrt{\left(v_y - \bar{v}_y\right)^2/\bar{v}_x} \). The Reynolds stress is calculated by \( \mathcal{R}_e = \rho \left(v_{xy} (v_{xy} - v_{xy})\right) \). The mean radial drift velocities are computed by dividing the mean dust momentum over the mean density. These properties are all spatially and time-averaged, indicated by angle brackets, based on many snapshots (30 \( \Omega_0^{-1} \sim 40 \Omega_0^{-1} \) in the AB test, 60 \( \Omega_0^{-1} \sim 800 \Omega_0^{-1} \) in the BA test), and the results are shown in Table 3. These diagnostic quantities are in broad agreement with the values obtained in Johansen & Youdin (2007) and Bai & Stone (2010c), all saturated into highly subsonic anisotropic turbulence, with enhanced radial drift and Reynolds stress in the AB test and reduced radial drift and Reynolds stress in the BA test.

### 3.3.4. Global Curvilinear Run of the BA test

In order to test our multifluid module in curvilinear coordinates, we run a global unstratified SI test in cylindrical coordinates \((r, \phi, z)\), similar to earlier investigations (Kowalik et al. 2013; Mignone et al. 2019). We choose to adopt parameters close to the BA test here, which is less demanding in resolution and has better convergence properties. The computational domains in three directions are \( r \in [0.2, 2.6] \),
Figure 7. The dust CDFs of the AB test (top) and the BA test (bottom), similar to Figure 10 of Benítez-Llambay et al. (2019). The left panels are the CDFs calculated by counting the number of cells whose dust density exceeds a certain threshold. The right panels are the CDFs calculated by additional weighting by dust density. The color-shaded regions are the temporal standard deviations based on many snapshots (from 30 \( \Omega_0^{-1} \) to 40 \( \Omega_0^{-1} \) for the AB test, and from 600 \( \Omega_0^{-1} \) to 800 \( \Omega_0^{-1} \) for the BA test). The initial dust densities are \( \rho_{d,0} = 1.0 \) and \( \rho_{d,0} = 0.2 \) for the AB and BA tests, respectively.

Table 3
Turbulence Properties of the AB and BA Tests with Different Resolutions

| Run     | \( M_a \) | \( M_{\infty} \) | \( M_{\infty} \) | \( \mathcal{R}_s / \mathcal{R}_{NSH} \) | \( V_{d,drift}/V_{d,drift,NSH} \) |
|---------|-----------|-----------------|-----------------|----------------------------------------|----------------------------------|
| AB-128^2 | 1.39(06) \times 10^{-2} | 8.97(55) \times 10^{-3} | 1.18(08) \times 10^{-2} | 2.15(22)                              | 1.74(07)                         |
| AB-256^2 | 1.46(05) \times 10^{-2} | 9.30(49) \times 10^{-3} | 1.10(03) \times 10^{-2} | 2.56(10)                              | 2.03(08)                         |
| AB-512^2 | 1.37(05) \times 10^{-2} | 7.32(39) \times 10^{-3} | 1.07(06) \times 10^{-2} | 2.66(12)                              | 2.19(07)                         |
| AB-1024^2 | 1.24(02) \times 10^{-2} | 6.01(24) \times 10^{-3} | 8.95(28) \times 10^{-3} | 2.55(07)                              | 2.15(05)                         |
| AB-2048^2 | 1.10(02) \times 10^{-2} | 4.96(17) \times 10^{-3} | 7.61(15) \times 10^{-3} | 2.38(02)                              | 2.07(02)                         |
| AB-4096^2 | 1.07(01) \times 10^{-2} | 4.93(14) \times 10^{-3} | 7.26(14) \times 10^{-3} | 2.34(04)                              | 2.03(02)                         |
| BA-64^2  | 1.03(15) \times 10^{-2} | 1.69(14) \times 10^{-2} | 3.92(33) \times 10^{-2} | 0.74(08)                              | 0.74(08)                         |
| BA-128^2 | 1.46(23) \times 10^{-2} | 2.03(20) \times 10^{-2} | 4.92(26) \times 10^{-2} | 0.61(07)                              | 0.63(06)                         |
| BA-256^2 | 1.78(11) \times 10^{-2} | 2.09(15) \times 10^{-2} | 4.91(45) \times 10^{-2} | 0.54(08)                              | 0.58(07)                         |
| BA-512^2 | 1.59(21) \times 10^{-2} | 2.08(14) \times 10^{-2} | 5.02(16) \times 10^{-2} | 0.62(06)                              | 0.65(05)                         |
| BA-1024^2 | 1.57(15) \times 10^{-2} | 2.07(07) \times 10^{-2} | 4.83(13) \times 10^{-2} | 0.58(04)                              | 0.62(04)                         |
| BA-2048^2 | 1.61(15) \times 10^{-2} | 2.12(20) \times 10^{-2} | 4.95(24) \times 10^{-2} | 0.61(05)                              | 0.64(05)                         |

Note. The numbers in parentheses quote the 1σ uncertainty of the last two digits.
\( \phi \in [0, 2\pi] \), and \( z \in [-0.15, 0.15] \). The numerical resolutions are 4096 \times 4 \times 512 cells along \( r, \phi, \) and \( z \), where we use a reduced \( \phi \)-resolution to preserve axisymmetry.

We set the central star mass \( GM = 1 \) and we set the gas radial density profile to be \( \rho_0(r) = \rho_0(r/r_0)^{-0.5} \), with \( \rho_0 = 1 \) at \( r_0 = 1 \). We adopt a vertically isothermal equation of state, with \( P(r) = c_s(r)^2 \rho_0^2(r) \). The sound speed is chosen so that the disk aspect ratio is \( h/r = c_s/v_{\text{K}} = 0.1 \) at all radii, which gives \( c_s = 0.1(r/r_0)^{-0.5} \). From Equation (41), we obtain \( \eta = 0.0075 \) (\( \Pi = 0.075 \)). The effective resolutions along the \( r \)- and \( z \)-directions are 12.8/\( n_{\phi} \), the same as BA-512\(^2 \). The initial dust density follows that of the gas, with a uniform dust-to-gas density ratio \( \epsilon_0 = 0.2 \). The Stokes number of the dust is \( St = T_0 \Omega_k = 1.0 \). The initial velocities of both the gas and dust are set by the NSH equilibrium. Periodic boundary conditions are applied along the \( \phi \)- and \( z \)-directions. The radial boundary condition is fixed by the NSH solution. To minimize the unphysical wave reflection, we apply wave-damping zones near the inner and outer radial boundaries (de Val-Borro et al. 2006, 2007), located at \( 0.20 < r < 0.32 \) and \( 2.44 < r < 2.60 \), where the gas and dust density and velocities (represented by \( x \)) are relaxed, according to:

\[
\frac{dx}{dt} = -\frac{x - x_{\text{init}}}{\tau} R(r),
\]

where \( x_{\text{init}} \) is the initial value and \( \tau \) is the damping rate. We adopt \( \tau = 1 \) in our simulation, and \( R(r) \) is a smoothing parabolic function that transitions from 0 in the active zones to 1 in the ghost zones. We also use the orbital advection scheme (Masset 2000, 2002; the FARGO algorithm) to reduce truncation errors. At the beginning, we add small velocity perturbations in white noise, with an amplitude of 0.02\( c_s \), to seed the instability.

Figure 8 shows different snapshots of the dust density in the global simulations of this global BA test. We can clearly see the progressive development of the SI from small radii to large radii, as the inner region has a shorter dynamical time. The simulation reaches its saturated state over the entire domain after about \( t = 600 \Omega_0^{-1} \), where \( \Omega_0 = \Omega_k \) at \( r = r_0 \), and we see the characteristic long dusty stripes and valleys, with a maximum of \( \rho_d^{1.5} \), significantly amplified by a factor of \( \geq 100 \).

More quantitatively, we have also examined the dust-to-gas ratio \( \epsilon \) and the dust CDFs of the global run, and compared them with those from the local BA-512\(^2 \) shown in Figure 9. The maximum \( \epsilon \) in the global simulation is higher than that in the local simulation by a small margin. Note that due to the reduction of the mean radial drift speed in the BA test as the SI saturates, and due to the SI developing faster in the inner region than in the outer region, the mean dust-to-gas ratio \( \epsilon \) in the global run increases with time, and reaches about 0.3 (rather than the initial value of 0.2) at \( t = 800 \Omega_0^{-1} \). This is likely the main cause of the stronger dust clumping found in the global run. On the other hand, from the dust CDFs, we see that the dust density distributions in the global and local runs converge within the error bars at relatively high densities with large \( \rho_{\text{threshold}} \) (\( \geq 5 \rho_0 \), both by counting numbers and counting density), suggesting that dust clumping is well captured in both the local and global simulations. However, there are some deviations at relatively small dust density. The cause of this deviation is unknown and may require further investigations beyond the scope of this work, but we speculate that it may be related to a combination of a higher pressure gradient, \( \Pi = 0.075 \) instead of 0.05, a higher mean \( \epsilon \), and the global nature of the simulation.

### 3.4. Mesh Refinement

In this subsection, we present additional tests to demonstrate the compatibility of our multifluid dust module with SMR/AMR. Following the convention, the root mesh here is called level 0, and each level of refinement doubles the resolution and is called level 1, 2, etc.

#### 3.4.1. SMR Run of the AB Test

We first rerun the AB test of the SI (see Section 3.3.3), but with two levels of SMR. The resolution of the root mesh is 256\(^2 \), and each level of refinement doubles the resolution in the central region along the \( z \)-direction. Because the AB test is sensitive to the amplitude of the initial perturbations, we use perturbations 10 times smaller (0.002\( c_s \)) than those in the uniform runs. The results are shown in Figure 10. As noted earlier, the outcome of the AB test depends on the resolution. Indeed, we see that the SI is first developed at the finest level, and quickly becomes nonlinear well within one orbital time, while the SI is developed more slowly in coarser meshblocks, and it is not until after about \( \tau \approx 15 \Omega_0^{-1} \) that the SI is fully developed in the entire domain. The overall pattern at each refinement level closely resembles those shown in Figure 6 with the same resolution (256\(^2 \)-1024\(^2 \)), and there are no abrupt features seen along the coarse-\(-1024\(^2 \) meshblock boundaries, which testifies to the compatibility of our multifluid dust module with SMR in a shearing box. We have also examined the CDFs of this SMR run, and found that the CDFs at a given level are largely consistent with the CDFs in the corresponding uniform-level runs discussed earlier, within the 1\( \sigma \) uncertainties.

#### 3.4.2. AMR Test of Kelvin–Helmholtz Instability

We next conduct the standard test problem of the Kelvin–Helmholtz instability (KHI in Athena++ with AMR, following the problem setup described in Section 3.4.3 of Stone et al. (2020)) exactly, but adding two dust species. The resolution of the root mesh is 256\(^2 \), with each meshblock’s size being 8\(^2 \), and the refinement condition is determined by

\[
g = \max(|\partial_t v_g|, |\partial_x v_g|, |\partial_t v_d|, |\partial_x v_d|),
\]

which represents the maximum spatial velocity gradients in gas and all dust fluids. Meshblocks with \( g > 0.1 \) will be refined, and those with \( g < 0.005 \) will be derefined. We set a maximum of three refinement levels. We add two dust species with stopping times \( T_{s,1} = 10^{-2} \), \( T_{s,2} = 10^{-8} \), but no dust diffusion. We consider the cases without and with dust feedback, and the dust-to-gas mass ratio for each species is set to unity. We use the "RK2-Implicit" drag integrator, PLM reconstruction for both gas and dust, the HLLC Riemann solver on gas, and a CFL number of 0.4. For comparison, simulations with a uniform resolution of 2048\(^2 \) (matching the finest level) are also conducted.

Figure 11 shows the results of our dusty KHI tests with AMR. The first and third rows show the gas and dust density patterns, while the second and fourth rows show the relative differences from the runs with uniform resolution. We see that the strongly coupled dust with \( T_{s,2} = 10^{-8} \) shares exactly the same density pattern as the gas, as expected. The more marginally coupled dust with \( T_{s,1} = 10^{-2} \), on the other hand, is depleted in the vortex...
centers, as it is relatively slow in response to the rapid vortical motion of filling in the vortex eyes. We also see that the sizes of the vortices are larger when feedback is included, as the inertia from more dust loading would require more space for the KHI patterns to roll over.

We find that there are no distinguishable differences between the AMR and uniform grid runs, with relative differences of at most a few percent at the vortex centers in the first dust species with $T_{s,1} = 10^{-2}$, due in part to the low dust densities in there. The time step of our KHI tests is around $5 \times 10^{-5}$ in code units, which is much larger than the stopping time of the second dust species ($T_{s,2} = 10^{-8}$), making the drag interaction highly stiff. The results again testify that our fully implicit methods are accurate and robust in these extremely stiff regimes with AMR.

4. Summary and Discussion

In this paper, we describe the algorithm and implementation of a multifluid dust module in Athena++, together with a suite of benchmark numerical tests. The dust is treated as an arbitrary
are conducted over normalized dust density explicitly simulating turbulence. This approach has been shown which mimics the response to background turbulence without implemented as a diffusion term in the dust continuity equation, concentration diffusion. Dust concentration diffusion is commonly not to conserve angular momentum in disk problems (Tomimaga et al. 2019). We further derive from a Reynolds averaging procedure the proper terms that should be included in the momentum equation, to ensure not only the proper momentum diffusion flux, but also Galilean invariance. The physically meaningful behavior of dust concentration diffusion including dust feedback is then illustrated by a simple test problem.

First, we have provided a consistent formulation of dust concentration diffusion. Dust concentration diffusion is commonly implemented as a diffusion term in the dust continuity equation, which mimics the response to background turbulence without explicitly simulating turbulence. This approach has been shown not to conserve angular momentum in disk problems (Tomimaga et al. 2019). We further derive from a Reynolds averaging procedure the proper terms that should be included in the momentum equation, to ensure not only the proper momentum diffusion flux, but also Galilean invariance. The physically meaningful behavior of dust concentration diffusion including dust feedback is then illustrated by a simple test problem.

Second, we have developed two fully implicit, second-order, accurate drag integrators, which naturally combine with the existing VL2 and RK2 integrators in Athena+++, to ensure second-order accuracy in time for the composite system, together with momentum conservation to machine precision. The integrators are stable to highly stiff regimes, not only in small dust stopping times, but also in regimes of high dust mass loading. Our code is, to our knowledge, the first to achieve the combination of these features. We have also implemented a number of explicit and semi-implicit drag integrators for nonstiff applications. In addition, we have incorporated frictional heating that can be applied to any of the drag integrators.

The development of the multifluid dust module in Athena++, a higher-order Godunov code, complements the multifluid dust module in the widely used FARGO3D code (Benítez-Llambay et al. 2019), which is Zeus-like (Stone & Norman 1992a, 1992b). We conducted a large suite of code tests demonstrating code performance, many in parallel to those done in Benítez-Llambay et al. (2019). In particular, we studied the SI from linear to nonlinear regimes, and the results are generally in good agreement. We anticipate that the aforementioned new features in our implementation will represent more benefits, in addition to the better shock-capturing capabilities inherent to Godunov codes.

One of the main advantages of our multifluid dust module is its compatibility with many of the existing functionalities and physics modules in Athena++. In particular, our dust fluid module is compatible with SMR and AMR, curvilinear coordinate systems, including cylindrical and spherical coordinates, shearing box and orbital advection, magnetic fields, diffusion physics (viscosity, thermal conduction, and nonideal MHD), etc. The implementation of the multifluid dust module thus enables a wide range of applications involving dust dynamics, particularly in relation to the study of physics, gas dynamics, and the observational signatures of PPDs and planet formation, which are especially relevant to current and future disk observations by ALMA, the James Webb Space Telescope, the Chinese Space Station Telescope (CSST), the Next Generation Very Large Array (ngVLA) and the Square Kilometer Array (SKA). We will also make this module publicly available in the near future, to benefit the broader astrophysical community.

There is still substantial room for further extensions to the multifluid dust module, including dust coagulation/fragmentation (Drazkowska et al. 2019), coupling with nonequilibrium radiative heating and cooling (Kamp & Dullemond 2004), and self-gravity. Additionally, the drag term is exactly the same as the coupling among charged and neutral species in weakly ionized plasmas (e.g., O’Sullivan & Downes 2006), and the coupling term can be extremely stiff in the strong coupling regime. Thus, our code can also potentially be extended to accurately handle weakly ionized plasmas from multifluid to strong coupling regimes. These directions will be considered in future works.

We thank Pablo Benítez-Llambay, Leonardo Krapp, Hui Li, Shengtai Li, Ruobing Dong, Rixin Li, and Shangfei Liu for helpful discussions. This work is supported by the National Key R&D Program of China, No. 2019YFA0405100, and the China Manned Space Project, with No. CMS-CSST-2021-B09.

Software: astropy (Astropy Collaboration et al. 2013), Athena++ (Stone et al. 2020), Mathematica (Wolfram 1991)

Appendix A

Galilean Invariance in Momentum Diffusion

In this appendix, we prove the Galilean invariance of our dust concentration diffusion formulation. In doing so, it would be much easier to recast the dust momentum equation into a
Euler-like equation. By applying the dust continuity Equation (4), and after some algebra, we arrive at

\[ \frac{\partial}{\partial t} (\rho_d + \nu_{d,\text{diff}}) + \nabla \cdot ((\rho_d + \nu_{d,\text{diff}}) \cdot \nabla)(\rho_d + \nu_{d,\text{diff}}) = \frac{1}{\rho_d} \nabla \cdot (\rho_d \nu_{d,\text{diff}} \nu_{d,\text{diff}}) + \frac{\nu_d - \nu_{d,\text{diff}}}{T_d}. \]  

(A1)

Let us consider a different frame that moves at a constant speed \( v_0 \), where physical quantities are denoted with a prime \( ' \). Obviously, we have \( \rho' = \rho, v' = v - v_0 \), and \( \nu'_{d,\text{diff}} = \nu_{d,\text{diff}} \). To prove that the equation is Galilean invariant, it suffices to show that the equations written in the new frame are exactly the same when expressed with primed quantities.

With Equation (A1), the proof becomes quite straightforward. We first replace \( \rho_d \) with \( \nu'_d + v_0 \), \( v'_d \) with \( v'_d + v_0 \), \( \rho \) with \( \rho' \), and \( \nu_{d,\text{diff}} \) with \( \nu'_{d,\text{diff}} \). We note that the right-hand side remains unchanged. Next, note that \( \partial/\partial t' = \partial/\partial t + v_0 \cdot \nabla \); we can see that the terms proportional to \( v_0 \) all cancel out, thus the form of the equation remains exactly the same as (A1), except that it is expressed in all primed quantities.
Figure 11. Snapshots of the KHI tests with two dust species with AMR. The top (bottom) six panels are the cases without (with) dust feedback at $t = 1.2$ ($t = 1.6$). The abbreviations “nofb” and “fb” are for the cases without and with feedback, respectively. From left to right, the panels are for gas, dust species 1, and dust species 2. The stopping times of the dust are $T_{s,1} = 10^{-2}$ and $T_{s,2} = 10^{-8}$. The first and third rows are the gas and dust densities from the AMR runs with up to three refinement levels. The second and fourth rows are the relative differences between the AMR runs and the uniform grid runs, at a resolution matching the finest AMR refinement level of 3. The edges of the meshblocks in the AMR runs are also indicated by the black solid lines.
In the above discussion, we emphasize that if drop off the time derivative term on \( \rho_d \nabla \), the Euler-like equation would become

\[
\frac{\partial v_d}{\partial t} + (v_d + 2n_d) \cdot \nabla v_d + \frac{v_d}{\rho_d} \nabla \cdot (\rho_d v_d \nabla) = \frac{v_d - v_d}{T_s},
\]

which is very different to Equation (A1). In particular, going through the same procedures, the presence of the third term on the left-hand side makes this equation violate the Galilean invariance.

### Appendix B

**Design of Second-order Fully Implicit Drag Integrators**

The “VL2-implicit” integrator is designed from the following format:

\[
M^{(n+1)} - M^{(n)} = hf(M^{(n+\frac{1}{2})}, W^{(n+\frac{1}{2})})
\]

\[
= hf\left[M^{(n+\frac{1}{2})} - \frac{h}{2}f(M^{(n+1)}, W^{(n+\frac{1}{2})}, W^{(n+\frac{1}{2})})\right]
\]

\[
= hf(M^{(n+\frac{1}{2})}, W^{(n+\frac{1}{2})}) - \frac{h^2}{2} \frac{\partial f}{\partial M}(M^{(n+1)} - M^{(n)})
\]

\[
= hf(M^{(n+\frac{1}{2})}, W^{(n+\frac{1}{2})}) + \frac{h^2}{2} \frac{\partial f}{\partial M}(M^{(n+1)} - M^{(n)})
\]

\[
- \frac{h^2}{2} \frac{\partial f}{\partial M}(M^{(n+1)} - M^{(n)}),
\]

where

\[
M^{(n+1)} = M^{(n)} + hf(M^{(n)}, W^{(n)})
\]

\[
M^{(n+1)} = M^{(n)} + hf(M^{(n+\frac{1}{2})}, W^{(n+\frac{1}{2})}),
\]

This leads to the integration schemes (25) and (26). The quantities at the \( n + \frac{1}{2} \) time step (denoted by a prime in Section 2.3.2) are obtained from the first stage of the algorithm, and any first-order implicit integration suffices (we use the backward Euler method).

The “RK2-Implicit” integrator is designed from the following format:

\[
M^{(n+1)} - M^{(n)} = \frac{h}{2}(M^{(n+1)}, W^{(n+1)}) + \frac{h}{2}f(M^{(n)}, W^{(n)})
\]

\[
= \frac{h}{2}f(M^{(n+1)}, W^{(n+1)}) + \frac{h}{2}[f(M^{(n+1)}, W^{(n+1)}, W^{(n+1)})]
\]

\[
= \frac{h}{2}f(M^{(n+1)}, W^{(n+1)}) + \frac{h}{2}f(M^{(n+1)}, W^{(n+1)}, W^{(n+1)})
\]

\[
= \frac{h}{2}f(M^{(n+1)}, W^{(n+1)}) + \frac{h}{2}\left[1 - \frac{\partial f}{\partial M}\right]f(M^{(n+1)}, W^{(n+1)})
\]

\[
+ \frac{h}{2}\left[1 - \frac{\partial f}{\partial M}\right]\frac{\partial f}{\partial M}(M^{(n+1)} - M^{(n)})
\]

\[
\Rightarrow M^{(n+1)} = M^{(n)} + \frac{h}{2}\left[1 - \frac{\partial f}{\partial M}\right]f(M^{(n+1)}, W^{(n+1)})
\]

\[
C.1. Second-order Explicit Methods

Here, we document the two explicit integrators, following the VL2 and RK2 integrators in Athena++, termed “VL2-Explicit” and “RK2-Explicit” in this paper. Note that the explicit integrators usually require the time step \( h < T_s \) for all dust species. The momentum update is as follows.

\[
M' = M^{(n)} + hf(M^{(n)}, W^{(n)}),
\]

\[
M^{(n+1)} = \frac{1}{2}(M^{(n)} + M') + \frac{1}{2}hf(M', W').
\]

\[
C.2. Second-order Semi-implicit Methods

Here, we present two semi-implicit methods, which are more robust than the explicit methods.

Trapezoid (Crank–Nicholson method): the trapezoid method is derived from

\[
M^{(n+1)} = M^{(n)} + \frac{1}{2}[hf(M^{(n)}, W^{(n)}) + hf(M^{(n+1)}, W^{(n)})],
\]

\[
M^{(n+1)} = M^{(n)} + hf(M^{(n+1)}, W^{(n+1)}),
\]

\[
M^{(n+1)} = M^{(n)} + \left(1 - \frac{h}{2}\right)f(M^{(n+1)}, W^{(n+1)}),
\]

The first stage of “Trapezoid” is updated by the backward Euler method with \( h \), so as to be compatible with “RK2-Explicit.”

Trapezoid backward differentiation formula 2 (TrBDF2): In the TrBDF2, the momentum at the middle stage \( n + \frac{1}{2} \) is calculated by the “Trapezoid” method, with the time step \( \frac{h}{2} \), so as to be compatible with “VL2-Explicit.” Then \( M^{(n+1)} \) is updated by the backward differentiation formula 2 (BDF2)
method at the stage $n+1$:

\[
M^{(n+\frac{1}{2})} = M^{(n)} + \left(1 - \frac{h}{4} \frac{\partial f}{\partial M}{(n)} \right)^{-1} \frac{h}{2} f(M^{(n)}, W^{(n)}),
\]

\[
M^{(n+1)} = \frac{4}{3} M^{(n+\frac{1}{2})} - \frac{1}{3} M^{(n)}
+ \frac{1}{3} \left(1 - \frac{h}{4} \frac{\partial f}{\partial M}{(n)} \right)^{-1} \frac{h}{2} f(M^{(n)}, V^{(n)}).
\] (C4)

**Appendix D**

**Performance**

The two higher-order fully implicit drag integrators, VL2-Implicit and RK2-Implicit, involve solving the inverse of an $(N_d + 1) \times (N_d + 1)$ matrix. The cost of matrix inversion is $O((N_d + 1)^3)$. Moreover, it also takes $O((N_d + 1)^3)$ to handle matrix multiplications, such as in computing $A$ in Equations (26) and (28). Such operations would make the calculations increasingly expensive as $N_d$ increases, and could become a bottleneck at sufficiently large $N_d$.

Here, we measure code performance as a function of $N_d$ from the NSH equilibrium test in the shearing box. The test is run on an Intel Xeon Gold 6132 CPU with 28 cores. We use all the cores with 28 meshblocks, so that we occupy the entire CPU (and hence its cache), to mimic more realistic situations in large-scale simulations (note that communications in Athena++ are mostly hidden, thanks to its use of tasklist and its performance being more sensitive to cache use). We use the HLLE Riemann solver for gas and PLM reconstruction for both gas and dust. We measure the performance in terms of the time spent by an individual core to update a single cell. The results for the different drag integrators are shown in Figure 12, as a function of the total number of species $(N_d + 1)$, which are further compared to results with the gas drag turned off.

For explicit and semi-implicit integrators, we see that the drag integrators add very limited computational cost relative to the no-drag case. In particular, the semi-implicit integrators that involve two matrix inversion operations remain computationally efficient, thanks to the fast analytical solver (Krapp & Benítez-Llambay 2020) that reduces the cost to $O(N_d + 1)$. The total cost increases linearly with $N_d$ for $N_d \lesssim 12$, but gets slightly nonlinear at larger $N_d$. We speculate that this is likely due to heavier memory use, which reduces cache performance.

For the two fully implicit solvers, we manage to improve the performance by using the fast matrix inversion at the first integration stage, yet the more complex matrix computation and inversion at the second stage substantially increases the computational cost. This cost increases nonlinearly with $N_d$. It is relatively negligible for $N_d \lesssim 5$, and remains minor compared to the rest of the dust integrator for $N_d \lesssim 10$, but becomes rather significant for larger $N_d$.

**Appendix E**

**Solutions to the Collision Tests**

The mutual drags between the gas and $n$ dust species can be written in the matrix form:

\[
\frac{\partial M}{\partial t} = AM = \begin{bmatrix}
-n \epsilon_i \alpha_i & \alpha_1 & \alpha_2 & \cdots & \alpha_n \\
\epsilon_1 \alpha_1 & -\alpha_1 & 0 & \cdots & 0 \\
\epsilon_2 \alpha_2 & 0 & -\alpha_2 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\epsilon_n \alpha_n & 0 & 0 & \cdots & -\alpha_n 
\end{bmatrix} \begin{bmatrix}
M_g \\
M_{d,1} \\
M_{d,2} \\
\vdots \\
M_{d,n}
\end{bmatrix}
\] (E1)

The analytic solution of the momentum vector must take the form $M(t) = \sum_i e_i \exp(\lambda_i t)$, where $e_i$ are the coefficients for each momentum component determined by the initial condition and $\lambda_i$ are the eigenvalues. The key is to solve the eigensystem $AM = \lambda M$, yielding eigenvalues $\lambda_i$ and eigenvectors $M_i$. The coefficients $e_i$ are obtained by decomposing the initial condition into the eigenvectors.

Because of momentum conservation, i.e., the drag force acted on the gas equals the sum of the drag forces acted on dust, the matrix $A$ has an eigenvalue $\lambda_0 = 0$, corresponding to bulk motion, and hence the coefficient $c_0 = v_{\text{COM}}$ is the velocity of the center of mass (COM). When multiple dust species share the same stopping time, the eigensystem can be greatly simplified (see Table 1 of Benítez-Llambay et al. 2019), but this is no longer true in the general case. We use Mathematica (Wolfram 1991) to calculate the eigenvalues and the rest of the coefficients in Table 1.

**Appendix F**

**Additional Numerical Tests**

In this appendix, we present additional code tests that largely reproduce the dusty sound wave test and the dusty shock test in Benítez-Llambay et al. (2019) to demonstrate our code performance.

**F.1. Dusty Sound Wave**

To demonstrate that our multifluid dust module achieves second-order accuracy when combined with the hydrodynamic solver, we conduct the dusty sound wave test by following Section 3.2 of Benítez-Llambay et al. (2019) exactly, originally proposed by Laibe & Price (2011, 2012). We use PLM spatial reconstruction and an isothermal equation of state with the

![Figure 12](image-url)
HLLE gas Riemann solver, and consider both the “VL2-Implicit” and “RK2-Implicit” drag integrators. The tests are conducted in 1D, starting from a resolution of $N = 64$ cells, and we double the resolution each time until we reach a resolution of $N = 512$ cells. We have conducted simulations for both single-species and multispecies cases, and found excellent agreement with the analytical theory. For brevity, we show in Figure 13 the time evolution of the normalized dust and gas velocities for the single-species dust case, showing that our numerical solution perfectly matches the analytical solution. Moreover, we measure the rms of the L1 norms $\left(\sum_n \left| U_n - U_{n,\text{ana}} \right| / N \right)^{1/2}$ (Equation 21 in Stone et al. 2020) after one wave period, similar to the approach in the linear wave test in Athena++ (Stone et al. 2020), where $U_n$ and $U_{n,\text{ana}}$ are the numerical and analytical solutions of the $n$th variable. The variables include gas density and velocity, and dust density and velocity, all in normalized units. We see in the right panel of Figure 13 that our code clearly achieves second-order convergence.

F.2. Dusty Shock

Being a Godunov code, Athena++ has excellent shock-capturing properties, which we demonstrate using the generalized dusty shock test presented in Section 3.3 of Benítez-Llambay et al. (2019), which is generalized from Lehmann & Wardle (2018). We follow the same procedures and adopt identical parameters as in Benítez-Llambay et al. (2019) to conduct two simulations with one and three dust species (two and four species in total) on 400 grid points. PLM spatial reconstruction and the “VL2-Implicit” drag integrator are used for these tests. The results are shown in Figure 14, which is to be compared with Figure 5 of Benítez-Llambay et al. (2019), side by side. Note that the shocks are in a steady state, thus we focus on the overall shock profile, rather than the specific shock locations. Because of the dust drag, the dust profile near the shock is connected by a precursor that is accurately reproduced, similar to that in FARGO3D. On the other hand, Athena++ captures the discontinuity in the gas within the neighboring cells, as opposed to ∼four cells in FARGO3D.
Figure 14. Normalized velocities (top) and densities (bottom) of the dusty wave tests, with one (left) and three (right) dust species, to be directly compared with Figure 5 of Benítez-Llambay et al. (2019). The gas profile is shown in red, while the other colors correspond to the dust species, with the insets showing the zoomed-in profiles across the shock. The analytical solutions are shown with the solid lines, while the numerical results are shown with the filled circles.

ORCID iDs
Pinghui Huang  https://orcid.org/0000-0002-7575-3176
Xue-Ning Bai  https://orcid.org/0000-0001-6906-9549

References
Andrews, S. M. 2020, ARA&A, 58, 483
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Bae, J., & Zhu, Z. 2018a, ApJ, 859, 118
Bae, J., & Zhu, Z. 2018b, ApJ, 859, 119
Bai, X.-N., & Stone, J. M. 2010a, ApJ, 722, 1437
Bai, X.-N., & Stone, J. M. 2010b, ApJL, 722, L220
Bai, X.-N., & Stone, J. M. 2010c, ApJS, 190, 297
Baruteau, C., & Zhu, Z. 2016, MNRAS, 458, 3927
Benítez-Llambay, P., Krupp, L., & Pessah, M. E. 2019, ApJS, 241, 25
Benítez-Llambay, P., & Masset, F. S. 2016, ApJS, 223, 11
Birnstiel, T., Dullemond, C. P., & Brauer, F. 2010, A&A, 513, A79
Carballido, A., Cuzzi, J. N., & Hogan, R. C. 2010, MNRAS, 405, 2339
Carrera, D., Johansen, A., & Davies, M. B. 2015, A&A, 579, A43
Cui, C., & Bai, X.-N. 2021, MNRAS, 507, 1106
Cuzzi, J. N., Dobrovolskis, A. R., & Champney, J. M. 1993, Icar, 106, 102
de Val-Borro, M., Artymowicz, P., D'Angelo, G., & Peplinski, A. 2007, A&A, 471, 1043
de Val-Borro, M., Edgar, R. G., Artymowicz, P., et al. 2006, MNRAS, 370, 529
Dong, R., Li, S., Chiang, E., & Li, H. 2017, ApJ, 843, 127
Dong, R., Li, S., Chiang, E., & Li, H. 2018, ApJ, 866, 110
Dong, R., Rafikov, R. R., & Stone, J. M. 2011a, ApJ, 741, 57
Dong, R., Rafikov, R. R., Stone, J. M., & Petrovich, C. 2011b, ApJ, 741, 56
Drazkowska, J., Li, S., Birnstiel, T., Stummeler, S. M., & Li, H. 2019, ApJ, 885, 91
Flock, M., Ruge, J. P., Dziurczewich, N., et al. 2015, A&A, 574, A68
Fu, W., Li, H., Lubow, S., Li, S., & Liang, E. 2014, ApJL, 795, L39
Fung, J., & Muley, D. 2019, ApJS, 244, 42
Garaud, P., Barrière-Fouchet, L., & Lin, D. N. C. 2004, ApJ, 603, 292
Goldreich, P., & Lynden-Bell, D. 1965, MNRAS, 130, 125
Goodman, J., & Pindor, B. 2000, Icar, 148, 537
Hanasz, M., Kowalik, K., Wójtowski, D., & Pawlasek, R. 2010a, EAS, 42, 275
Hanasz, M., Kowalik, K., Wójtowski, D., Pawlasek, R., & Kornet, K. 2010b, EAS, 42, 281
Hawley, J. F., Gummie, C. F., & Balbus, S. A. 1995, ApJ, 440, 742
Hopkins, P. F., & Squire, J. 2018, MNRAS, 479, 4681
Hsieh, H.-F., & Lin, M.-K. 2020, MNRAS, 497, 2425
Hu, Z., & Bai, X.-N. 2021, MNRAS, 503, 162
Huang, P., Li, H., Isella, A., et al. 2020, ApJ, 893, 89
Johansen, A., Klahr, H., & Henning, T. 2011, A&A, 529, A62
