

Lawrence Berkeley National Laboratory

LBL Publications

Title

MFIX-Exa: CFD-DEM simulations of thermodynamics and chemical reactions in multiphase flows

Permalink

<https://escholarship.org/uc/item/0519m8nn>

Authors

Porcu, Roberto
Musser, Jordan
Almgren, Ann S
et al.

Publication Date

2023-06-01

DOI

10.1016/j.ces.2023.118614

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

MFIX-Exa: CFD-DEM simulations of thermodynamics and chemical reactions in multiphase flows

Roberto Porcu^{a,c,*}, Jordan Musser^a, Ann S. Almgren^b, John B. Bell^b, William D. Fullmer^{a,c} and Deepak Rangarajan^{a,c}

^aNational Energy Technology Laboratory

^bLawrence Berkeley National Laboratory

^cNETL Support Contractor

ARTICLE INFO

Keywords:

Chemical looping reactor
Fluidized beds
Godunov time-advancing
Embedded boundaries
CO₂ capturing
AMReX
Exascale

Abstract

MFIX-Exa is a CFD-DEM code for the numerical solution of chemically reacting multiphase flows (fluid and solids phases), specifically targeted for flows in complex reactor geometries. The fluid is modeled using a low Mach number formulation with a multicomponent ideal gas equation of state, which is imposed as a constraint of the velocity field. The fluid equations are discretized using an embedded boundary (EB) aware Godunov scheme with an approximate projection. The particles (that constitute the solids phase) are represented by a soft-sphere spring-dashpot model and evolved using a forward Euler method with subcycling. The fluid and particles models are coupled through a volume fraction field in addition to interphase mass, momentum, and energy transfer. The mathematical model and numerical approach are benchmarked against three different verification tests and validated with two separate tests. Also, a scaling analysis is provided. This manuscript represents the current state-of-the-art of MFIX-Exa and describes the major extensions to the previous work presented in Musser et al. [28], including the Godunov time integration algorithm for the fluid phase and the inclusion of thermodynamics and chemistry modeling to both the fluid and solids phases.

1. Introduction

Electricity production and industrial processes are responsible for approximately half of all greenhouse gas emissions in the United States [39], most of which are due to the burning of fossil fuels. Carbon capture and storage (CCS) technologies based on chemical looping reactor (CLR) designs offer a promising approach for decreasing CO₂ emissions from fossil fuel power plants. Large-scale commercial deployment of new and developing CO₂ capture technologies requires the scale up from laboratory designs to industrial systems. This large separation of scales is known to be fraught with issues, particularly for the multiphase reactors at the core of these devices.

Modeling and simulation can help aid in the design and deployment of new CCS technologies, however, the underlying physics of gas-solid multiphase flows is complex. As a result, a hierarchy of modeling approaches exist for particle flow, which range in applicable scales and are inversely proportional to the level of model error and uncertainty [20]. At one end of the spectrum, particle resolved direct numerical simulation (PR-DNS) [38] is accurate and reliable but computationally expensive, limiting the scale to which it can be applied. Conversely, commercial-scale systems can be modeled by multiphase particle-in-cell (PIC) methods [10] in which computational affordability is achieved by sacrificing accuracy through lumping hundreds of thousands of particles into single Lagrangian elements.

In an effort to help bridge this gap with high performance computing (HPC), we have developed MFIX-Exa, a modern, massively parallel CFD-DEM code. As an Exascale Computing Project (ECP) application code, MFIX-Exa was developed to be performant on current and the next generation of U.S. Department of Energy leadership class computing facilities. MFIX-Exa follows a traditional CFD-DEM approach [16] in which the trajectories of all particles are explicitly resolved with a continuous collision model yet the particles are unresolved by the fluid grid, requiring closures for interfacial transfer. We believe this provides an optimal modeling approach, balancing fidelity and cost.

*Corresponding author

✉ roberto.porcu@netl.doe.gov (R. Porcu); jordan.musser@netl.doe.gov (J. Musser); asalmgren@lbl.gov (A.S. Almgren); jbell@lbl.gov (J.B. Bell); william.fullmer@netl.doe.gov (W.D. Fullmer); deepak.rangarajan@netl.doe.gov (D. Rangarajan)
ORCID(s): <https://orcid.org/0000-0002-2225-4154> (R. Porcu); <https://orcid.org/0000-0003-4542-7166> (J. Musser); <https://orcid.org/0000-0003-2103-312X> (A.S. Almgren); <https://orcid.org/0000-0002-5749-334X> (J.B. Bell); <https://orcid.org/0000-0002-0142-7781> (W.D. Fullmer)

In a previous work [26], we laid the ground work for how the MFiX-Exa code was developed within the AMReX [42, 41] framework for cold-flow physics, i.e., only considering mass and momentum transport equations. In this work we present the conservation equations for internal energy, species mass, and chemical reactions as well as two ideal-gas equations of state for open and closed systems. Additionally, the numerical implementation has been improved by extending the time-integration algorithm from Method-of-Lines (MOL) to a Godunov method, providing improved performance while preserving the second-order accuracy in time.

This paper is organized into three major sections. First in Sec. 2, we describe the mathematical model for multiphase flow physics considered. We first introduce the conservation equations for the fluid phase in Sec. 2.1. Then in Sec. 2.2 the two possible equations of state, incompressible or ideal gas, for the fluid phase are discussed. The fluid description is followed by the conservation equations for the particles in Sec. 2.3. The mathematical model is closed in Sec. 2.4 by defining the interphase transfer terms. The mathematical model is followed by a description of the numerical implementation adopted to discretize the fluid phase and particle advance in Sec. 3.1 and Sec. 3.2, respectively. The code is applied in Sec. 4 which is split into verification tests in Sec. 4.1, validation tests in Sec. 4.2, and performance tests in Sec. 4.3. The work is concluded in Sec. 5.

2. Mathematical model

The mathematical model is based on a CFD-DEM approach that couples a continuum representation of the gas phase with a discrete element method (DEM) for representing the dispersed phase. The continuous gas phase is modeled using a multicomponent low Mach-number formulation in an Eulerian frame. The dispersed phase, referred to as particles, is treated in a Lagrangian frame employing equations of classical dynamics with collision forces computed using a soft-sphere model [20, 13]. Unlike particle-resolved direct numerical simulations [38], particles in CFD-DEM are typically smaller than the fluid mesh, so additional models are needed to account for fluid-particle interactions, such as an interfacial drag. In the following sections, the fluid model and equation of state are provided, followed by a description of the particle model. Lastly, interphase and intraphase transfer quantities are discussed.

2.1. Fluid phase equations

The low Mach number (M) formulation assumes that the fluid velocity is much smaller than the speed of sound. In this regime, pressure p_g can be decomposed into a thermodynamic pressure p_{therm} that is only a function of time, and a perturbational pressure field π_g that satisfies $\pi_g/p_{\text{therm}} \sim \mathcal{O}(M^2)$, [see 30]:

$$p_g(\mathbf{x}, t) = p_{\text{therm}}(t) + \pi_g(\mathbf{x}, t). \quad (1)$$

In the low Mach number approximation, π_g does not affect the thermodynamics of the gas.

Conservation of mass (i.e., the continuity equation) for the fluid is

$$\frac{\partial (\varepsilon_g \rho_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u}_g) = \mathcal{G}_{p \rightarrow g}^p, \quad (2)$$

where ρ_g and \mathbf{u}_g are the fluid density and velocity, respectively, and ε_g is the gas phase volume fraction. ε_g is computed from the deposition of particle volume to the grid such that $\varepsilon_g \equiv 1 - \varepsilon_p$, where ε_p is the volume fraction of the solids phase. $\mathcal{G}_{p \rightarrow g}^p$ is a source term that represents the mass transferred from the particles to the gas due to interphase chemical reactions. In this manuscript, the subscript arrow in the interphase transfer terms represents the direction with a sign of the transferred quantities. With this notation, a positive $\mathcal{G}_{p \rightarrow g}^p$ stands for a transfer of mass from the particle phase to the fluid phase, and vice-versa. The right-hand side is zero in the case of flows without chemical reactions or physical processes such as evaporation.

Convective form of linear momentum equation for the fluid is:

$$\frac{\partial (\varepsilon_g \rho_g \mathbf{u}_g)}{\partial t} + (\varepsilon_g \rho_g \mathbf{u}_g) \cdot \nabla \mathbf{u}_g + \nabla \pi_g = \nabla \cdot \boldsymbol{\tau}_g + \varepsilon_g \rho_g \mathbf{g} + \mathcal{F}_{p \rightarrow g}^p + \mathcal{G}_{p \rightarrow g}^p, \quad (3)$$

where $\boldsymbol{\tau}_g$ is the fluid phase stress tensor, and \mathbf{g} is the constant acceleration of gravity. The fluid phase viscous stress tensor is given by

$$\boldsymbol{\tau}_g = \mu_g \left(\nabla \mathbf{u}_g + \nabla \mathbf{u}_g^T \right) - \frac{2}{3} \mu_g (\nabla \cdot \mathbf{u}_g) \mathbf{I},$$

where μ_g is the fluid phase shear viscosity, and \mathbf{I} is the identity tensor. The last two terms on the right hand side include interphase momentum transfer from fluid-particle interfacial drag, $\mathcal{F}_{p \rightarrow g}^\ell$, and from mass exchange with particles, $\mathcal{G}_{p \rightarrow g}^\rho$.

Conservation of species in the fluid phase is given by

$$\frac{\partial (\varepsilon_g \rho_g X_{gk})}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g X_{gk} \mathbf{u}_g) = \nabla \cdot (\varepsilon_g \mathbf{J}_{gk}) + \mathcal{G}_{p \rightarrow gk}^\rho \equiv S_k^\rho, \quad \forall k = 1, \dots, N_g, \quad (4)$$

where $X_{gk} \in [0, 1]$, is the k -th fluid species mass fraction of N_g different fluid species. The species mass fractions are required to satisfy the identity $\sum_k X_{gk} = 1$. The quantity \mathbf{J}_{gk} is the fluid species diffusion flux. Rather than using a more detailed Maxwell-Stefan representation of species diffusion, we use a Fickian approximation [6] for the species flux,

$$\mathbf{J}_{gk} = \rho_g \mathbf{D}_g \nabla X_{gk}, \quad \forall k = 1, \dots, N_g.$$

For simplicity, we have assumed that the diffusion coefficient is a constant for all species, $\mathbf{D}_{gk} = \mathbf{D}_g$. The second source term, $\mathcal{G}_{p \rightarrow gk}^\rho$, represents contributions due to heterogeneous chemical reactions with the particles and satisfies $\sum_k \mathcal{G}_{p \rightarrow gk}^\rho = \mathcal{G}_{p \rightarrow g}^\rho$. The two source terms of (4) are combined into a single quantity S_k^ρ for convenience when the equation of state is introduced in Section 2.2.

In low Mach number models, the conservation of energy is expressed in terms of enthalpy,

$$\frac{\partial (\varepsilon_g \rho_g h_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g h_g \mathbf{u}_g) - \varepsilon_g \frac{dp_{\text{therm}}}{dt} = \sum_{k=1}^{N_g} \nabla \cdot (\varepsilon_g h_{gk} \mathbf{J}_{gk}) - \nabla \cdot (\varepsilon_g \mathbf{q}_g) + \mathcal{F}_{p \rightarrow g}^h + \mathcal{G}_{p \rightarrow g}^h \equiv S^h, \quad (5)$$

where h_g is the specific enthalpy of the fluid, and h_{gk} is the specific enthalpy of the k -th fluid species. p_{therm} is the thermodynamic pressure defined in eq. (1), which is only a function of time. Conductive heat transfer is given by Fourier's law [6]

$$\mathbf{q}_g = -\kappa_g \nabla T_g,$$

where T_g is the fluid temperature, and κ_g is the thermal conductivity coefficient. Given a fixed reference temperature T_{ref} , the specific enthalpy of the fluid species is

$$h_{gk}(T_g) = h_{gk}^{\text{ref}} + \int_{T_{\text{ref}}}^{T_g} c_{p,gk}(T') dT', \quad \forall k = 1, \dots, N_g,$$

where $c_{p,gk}$ (which is a function of T_g) is the specific heat at constant pressure of species k , and h_{gk}^{ref} is the k -th fluid species specific enthalpy of formation at T_{ref} temperature. Then, for an ideal gas, the specific mixture enthalpy h_g of the fluid is

$$h_g(T_g) = \sum_{k=1}^{N_g} X_{gk} h_{gk}(T_g). \quad (6)$$

The additional right hand side terms in eq. (5) correspond to the enthalpy flux due to the species diffusion and interphase enthalpy sources from convective heat transfer, $\mathcal{F}_{p \rightarrow g}^h$, and due to the heterogeneous chemical reactions, $\mathcal{G}_{p \rightarrow g}^h$. Again, all terms on the right-hand side of the conservation of enthalpy equation are combined into a single convenience quantity, S^h .

2.2. Fluid phase equation of state

The system of conservation equations for the fluid phase is closed by prescribing an equation of state from which we can derive a constraint on the fluid velocity field. The constraint forces the evolution of the conservation equations to be consistent with the equation of state and determines the perturbational pressure, π_g . Details of the computation of π_g are discussed in Section 3.1. We consider two equations of state: an *incompressible fluid* and a *multicomponent*

ideal gas. The incompressible fluid formulation is used for non-reacting, cold-flow simulations where the fluid density is constant. In this case, the constraint is given by

$$\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) = 0, \quad (7)$$

which states that the flux of fluid into a region is exactly balanced by the change in fluid volume fraction.

For more general cases, where reactions, heat transfer, and species transport play an essential role, the fluid phase is treated as a multicomponent ideal gas with the equation of state given by

$$p_{\text{therm}} = \rho_g R T_g \sum_{k=1}^{N_g} \frac{X_{gk}}{W_{gk}}, \quad (8)$$

where R is the universal gas constant and W_{gk} is the k -th gas species molecular weight.

When the system is open, it maintains a specified ambient pressure throughout the evolution so that the thermodynamic pressure is constant; i.e., $p_{\text{therm}} = p_{\text{amb}} = \text{const}$. One can show that this constraint implies

$$\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) = \frac{S^h}{\rho_g c_{p,g} T_g} + \frac{1}{\rho_g} \sum_{k=1}^{N_g} \left(\frac{W_g}{W_{gk}} - \frac{h_{gk}}{c_{p,g} T_g} \right) S_k^\rho \equiv S, \quad (9)$$

where W_g and $c_{p,g}$ are the molecular weight and the specific heat constant of the mixture of species for the fluid phase, respectively. S is a term that represents the right-hand side of the constraint equation and will be used later. W_g and $c_{p,g}$ are defined as

$$W_g = \left(\sum_{k=1}^{N_g} \frac{X_{gk}}{W_{gk}} \right)^{-1}, \quad \text{and} \quad c_{p,g}(T_g) = \sum_{k=1}^{N_g} X_{gk} c_{p,gk}(T_g).$$

This constraint ensures that the evolution of the species' masses and enthalpy are constrained so that the thermodynamic pressure remains constant. Details of the derivation of the constraint are shown in Appendix A.

For a closed system, the constraint is more complex because the fluid occupies a fixed volume, and the thermodynamic pressure needs to evolve in time to balance processes such as reactions. The approach here follows the work discussed in Nonaka et al. [30]. In this case, the equation of state takes the form of a constraint on the fluid velocity and an equation for the evolution of the thermodynamic pressure

$$\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) = \delta S - \delta \theta \frac{\bar{S}}{\theta}, \quad \text{and} \quad \frac{D p_{\text{therm}}}{Dt} = \frac{\bar{S}}{\theta}. \quad (10)$$

Here we have introduced the decomposition of a variable $\phi = \bar{\phi} + \delta\phi$ into its mean value $\bar{\phi}$ and its fluctuations $\delta\phi$, and S represents the right hand side of the constraint (9) discussed above. We also define θ as

$$\theta = \varepsilon_g \left(\frac{1}{p_{\text{therm}}} - \frac{1}{\rho_g c_{p,g} T_g} \right).$$

As in the open system case, this constraint analytically enforces that the evolution of the gas density and temperature are consistent with the ideal gas equation of state (8). Details of this derivation can also be found in Appendix A.

2.3. Particle equations

The positions and velocities of the particles are computed using a DEM approach that models particle collisions via a soft-sphere spring-dashpot model [7]. Particles are assumed to be spherical and with constant diameter;¹ therefore, the p -th particle mass is given by $m_p = \rho_p \mathcal{V}_p$ where ρ_p is the particle density, and $\mathcal{V}_p = \pi d_p^3/6$ is particle volume. The mass balance equation for the p -th particle satisfies²

$$\frac{dm_p}{dt} = G_{g \rightarrow p}^m, \quad (11)$$

¹We do not consider breakage or agglomeration models that would cause particle mass to change without chemical reactions or phase changes.

²The loss of mass could be reflected in a change in diameter rather than a change in density; however, this case introduces additional complexity into the gas phase equation. Specifically, the low Mach number constraints discussed above would need to be modified to take this into account.

where $G_{g \rightarrow p}^m$ is the net mass transfer between the p -th particle and the fluid. $G_{g \rightarrow p}^m$ is positive when the particle receives mass from the fluid phase, and it is negative when there is a transfer of mass from the particles to the fluid phase. This term contributes to the right-hand side of the fluid continuity equation (2), $G_{p \rightarrow g}^o$, so that the net mass in the system is not changed. The connection between these terms is discussed in Section 2.4.1.

The motion of the particle is given by

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p, \quad \text{and} \quad \frac{d(m_p \mathbf{u}_p)}{dt} = m_p \mathbf{g} + \mathbf{F}_{pw} + \sum_{q=1}^{N_{\text{ngb}}} \mathbf{F}_{pq} + \mathbf{F}_{g \rightarrow p}^L + \mathbf{G}_{g \rightarrow p}^L, \quad (12)$$

where \mathbf{x}_p is the particle position and \mathbf{u}_p is the particle velocity. The first three forces on the right-hand side represent gravitational acceleration, collision forces with wall boundaries, and collision forces with the N_{ngb} neighbor particles, respectively. The last two terms account for interphase linear momentum exchange related to drag and buoyancy, $\mathbf{F}_{g \rightarrow p}^L$, and linear momentum transfer from chemical reactions, $\mathbf{G}_{g \rightarrow p}^L$.

The angular momentum equation for each particle is given by

$$\frac{d(I_p \boldsymbol{\omega}_p)}{dt} = \mathbf{T}_{pw} + \sum_{q=1}^{N_{\text{ngb}}} \mathbf{T}_{pq}, \quad (13)$$

where $I_p = m_p d_p^2 / 10$ is the particle moment of inertia. On the right-hand side, the first term represents torque arising from collisions with wall boundaries, and the second term accounts for the sum of the torques produced by collisions with the N_{ngb} neighbor particles.

The solid species mass evolution for each particle is

$$\frac{d(m_p X_{pk})}{dt} = G_{g \rightarrow pk}^m, \quad \forall k = 1, \dots, N_s, \quad (14)$$

where X_{pk} represents the mass fraction of the k -th species in the particle. The terms on the right-hand side account for the change in the k -th species mass fraction due to reactions within a single particle³ and heterogeneous chemical reactions with the gas phase. N_s is the number of species for the dispersed phase.

The evolution of the enthalpy for each particle is given by

$$\frac{d(m_p h_p)}{dt} = F_{g \rightarrow p}^H + G_{g \rightarrow p}^H, \quad (15)$$

where h_p is the particle's specific enthalpy, $F_{g \rightarrow p}^H$ accounts for the interphase convective heat transfer, and the last term represents enthalpy sources from interphase chemical reactions. We view each particle as a mixture of N_s different solid species so that the specific enthalpy, h_p for particle p is

$$h_p(T_p) = \sum_{k=1}^{N_s} X_{pk} h_{pk}(T_p), \quad (16)$$

where T_p is the particle temperature and $h_{pk}(T_p)$ is the specific heat of the k -th solids species, defined by the following equation:

$$h_{pk}(T_p) = h_{pk}^{\text{ref}} + \int_{T_{\text{ref}}}^{T_p} c_{p,pk}(T) dT, \quad \forall k = 1, \dots, N_s. \quad (17)$$

Here, $c_{p,pk}$ is the specific heat coefficient of the k -th solid species at constant pressure, T_{ref} is the reference temperature, and h_{pk}^{ref} is the specific enthalpy of formation of the k -th solid species at T_{ref} temperature.

³We do not consider chemical reactions between two particles.

2.4. Fluid-particle coupling

Unlike particle-resolved direct numerical simulations [38], closure models are needed to account for interphase mass, momentum, and energy transfer. These terms are computed at the p -th particle position using continuous fluid properties mapped to the particle location by an interpolation operator, χ . A given particle property, B_p , is transformed into a continuous field, \mathcal{B} , by the unit normal transfer kernel \mathcal{K} [3, 9]

$$\mathcal{B}(\mathbf{x}, t) = \sum_{p=1}^{N_{\text{txfr}}} B_p(t) \mathcal{K}(\|\mathbf{x} - \mathbf{x}_p\|). \quad (18)$$

Both χ and \mathcal{K} are compact and monotonically decreasing so that only a small compact subset of computational cells near the p -th particle contributes to the interpolated value, and only a few particles N_{txfr} contribute to an Eulerian cell value of a filtered particle property. Details of the numerical implementation are provided in [28]. Subsequent sections present the fluid-particle transfer terms for mass, momentum, and energy, which are computed at a particle's position before being transferred to the fluid.

2.4.1. Species mass

For a system with Q heterogeneous chemical reactions, the interphase species mass transfer rates between the fluid and p -th particle are given by

$$G_{p \rightarrow gk}^m = \sum_{q=1}^Q [\alpha_{gk}]_q W_{gk} \mathcal{R}_q, \quad G_{g \rightarrow pk}^m = \sum_{q=1}^Q [\alpha_{pk}]_q W_{pk} \mathcal{R}_q,$$

where \mathcal{R}_q is the reaction rate and $[\alpha_{gk}]_q$ and $[\alpha_{pk}]_q$ are the signed stoichiometric coefficients of the k -th fluid and particle species for the q -th reaction, respectively. The sign of a stoichiometric coefficient is positive if the species is produced and negative if the species is consumed. The total mass transfer rates due to heterogeneous chemical reactions $G_{g \rightarrow p}^m$ and $G_{p \rightarrow g}^m$, respectively for the p -th particle and the fluid, are

$$G_{p \rightarrow g}^m = \sum_{k=1}^{N_g} G_{p \rightarrow gk}^m, \quad \text{and} \quad G_{g \rightarrow p}^m = \sum_{k=1}^{N_s} G_{g \rightarrow pk}^m,$$

which satisfy $G_{p \rightarrow g}^m + G_{g \rightarrow p}^m = 0$. Also, $G_{g \rightarrow p}^m$ and $G_{g \rightarrow pk}^m$ are source terms in the particle mass (11) and species mass (14) equations. Here, the source terms connected to chemical reactions for the fluid species mass (4) and continuity (2) equations are obtained by volume filtering $G_{p \rightarrow gk}^m$ and summing over all fluid species:

$$\mathcal{G}_{p \rightarrow gk}^\rho = \sum_{p=1}^{N_{\text{txfr}}} G_{p \rightarrow gk}^m \mathcal{K}(\|\mathbf{x} - \mathbf{x}_p\|), \quad \text{and} \quad \mathcal{G}_{p \rightarrow g}^\rho = \sum_{k=1}^{N_g} \mathcal{G}_{p \rightarrow gk}^\rho.$$

2.4.2. Momentum

In this work, only buoyancy and steady drag are considered in the linear momentum conservation equations (3) and (12), due to the high-density ratio of gas and solid particles, in addition to the large Stokes numbers of our target applications. For this reason, the force acting on each p -th particle is

$$\mathbf{F}_{g \rightarrow p}^L = -\chi_p(\nabla p_g) \mathcal{V}_p - f_{\text{dp}}(\mathbf{u}_p - \chi_p(\mathbf{u}_g)), \quad (19)$$

where the first component is related to the buoyancy force, and the second describes the interphase drag acting on the particle. The linearized drag coefficient in (19), f_{dp} , is given by

$$f_{\text{dp}} = \frac{1}{2} C_d \chi_p(\rho_g) \|\mathbf{u}_p - \chi_p(\mathbf{u}_g)\| A_{\text{proj}}. \quad (20)$$

where $A_{\text{proj}} = \pi d_p^2/4$ is the projected area of the p -th particle. Three standard multi-particle drag laws have been implemented to close C_d in Eq. (19): Wen and Yu [40], Ding and Gidaspow [14, 23], and Tang et al. [5, 37]. The drag

coefficient, and the drag force component coming from the solids phase, when transferred to the grid, respectively become

$$\beta = \sum_p f_{dp} \mathcal{K} \left(\|\mathbf{x} - \mathbf{x}_p\| \right), \quad \text{and} \quad \beta \mathbf{u}_s = \sum_p f_{dp} \mathbf{u}_p \mathcal{K} \left(\|\mathbf{x} - \mathbf{x}_p\| \right).$$

Thus, the source term for the fluid momentum equation (3) is given by

$$\mathcal{F}_{p \rightarrow g}^{\ell} = \beta \mathbf{u}_s - \beta \mathbf{u}_g.$$

Finally, we discuss the interphase momentum transfer that occurs when there is a net exchange of mass from one phase to another. We define the fluid to solid phase, $G_{g \rightarrow p}^L$, and the solid to fluid phase, $G_{p \rightarrow g}^{\ell}$, momentum transfer quantities as

$$G_{g \rightarrow p}^L = \max \left\{ 0, G_{g \rightarrow p}^m \right\} \chi_p(\mathbf{u}_g), \quad \text{and} \quad G_{p \rightarrow g}^{\ell} = \sum_p \max \left\{ 0, G_{p \rightarrow g}^m \right\} \mathbf{u}_p \mathcal{K} \left(\|\mathbf{x} - \mathbf{x}_p\| \right).$$

For example, if there is a net mass transfer between the fluid and the p -th particle, the new mass received by the fluid is assumed to be moving with the same velocity as the p -th particle. Conversely, mass transferred from the fluid to the particle is assumed to be moving at the velocity of the fluid.

2.4.3. Energy

The rate of convective heat exchange between the p -th particle and the fluid is given by

$$F_{g \rightarrow p}^H = \gamma_p A_{\text{surf}} \left(\chi_p(T_g) - T_p \right),$$

where $A_{\text{surf}} = \pi d_p^2$ is the particle surface area, and T_g and T_p are the fluid and particle temperatures, respectively. $\gamma_p = \text{Nu}_p \kappa_g / d_p$ is the convective heat transfer coefficient, and κ_g is the fluid thermal conductivity. Two Nusselt number correlations have been implemented to model heat transfer coefficient, Ranz and Marshall [32] and Gunn [19]. When transferred to the grid, the convective heat coefficient, and the convective heat exchange component coming from the solids phase, respectively become

$$\Gamma = \sum_{p=1}^{N_{\text{txfr}}} \gamma_p A_{\text{surf}} \mathcal{K} \left(\|\mathbf{x} - \mathbf{x}_p\| \right), \quad \text{and} \quad \Gamma T_s = \sum_{p=1}^{N_{\text{txfr}}} \gamma_p A_{\text{surf}} T_p \mathcal{K} \left(\|\mathbf{x} - \mathbf{x}_p\| \right).$$

Then, the source term contributing to the fluid enthalpy equation (5) is given by

$$\mathcal{F}_{p \rightarrow g}^h = \Gamma T_s - \Gamma T_g.$$

The model for heat transfer due to heterogeneous chemical reactions follows the approach described in Musser et al. [29]. Given the q -th reaction, the gas phase reactants are transferred to the p -th particle at the gas phase temperature T_g :

$$\left[G_{g \rightarrow p}^H \right]_q = \sum_{k=1}^{N_g} h_{gk} \left(\chi_p(T_g) \min \left\{ 0, \chi_p \left(\left[G_{p \rightarrow gk}^m \right]_q \right) \right\} \right).$$

The above expression relies on the sign of the interphase species mass transfer term to distinguish between reactants and products. Specifically, $\left[G_{p \rightarrow gk}^m \right]_q$ is negative if the k -th gas species is a reactant in the q -th reaction and positive otherwise. On the other hand, gas phase products are transferred to the gas at the particle temperature:

$$\left[G_{p \rightarrow g}^H \right]_q = \sum_{k=1}^{N_g} h_{gk}(T_p) \max \left\{ 0, \left[G_{p \rightarrow gk}^m \right]_q \right\}.$$

The total energy transfer to the gas phase is given by summing $[G_{g \rightarrow p}^H]_q$ and $[G_{p \rightarrow g}^H]_q$ over all the heterogeneous chemical reactions, while the energy transfer to the particle phase is equal but opposite in sign:

$$G_{p \rightarrow g}^H = \sum_{q=1}^Q \left([G_{p \rightarrow g}^H]_q + [G_{g \rightarrow p}^H]_q \right), \quad \text{and} \quad G_{g \rightarrow p}^H = -G_{p \rightarrow g}^H.$$

Finally, the source term for the fluid enthalpy equation (5) is given by the volume filtered equivalent of $G_{p \rightarrow g}^H$:

$$\mathcal{G}_{p \rightarrow g}^h = \sum_{p=1}^{N_{\text{txfr}}} G_{p \rightarrow g}^H \mathcal{K} \left(\left\| \mathbf{x} - \mathbf{x}_p \right\| \right).$$

3. Numerical method

MFIX-Exa is built on the AMReX software framework [41, 42] that provides data structures and iterators defined on a block-structured mesh in a distributed memory environment. From a high level, MFIX-Exa uses an operator splitting approach in which fluid and particle updates are segregated with several coupling steps between them. A breakdown of the overall advance is presented in Algorithm 1. The fluid phase is advanced using a fractional step approach with an approximate projection method. Standard forward Euler time-stepping is used to update the particle phase. Coupling is achieved by a trilinear interpolator transfer kernel. The key numerical methods are given below for particles and fluid cells in a “regular” neighborhood, i.e., away from domain boundaries. As in AMReX, irregular boundaries are modeled in MFIX-Exa with an embedded boundary (EB) approach. The EB intersects the fluid mesh producing cut cells and a level-set function is constructed for particle-wall interactions. Details of the EB-aware Godunov method and EB-aware transfer kernel which must be applied in “irregular” neighborhoods can be found in Appendix B and our previous work [26], respectively.

Algorithm 1 Overview of the time step

- 1: Calculate the fluid volume fraction ε_g^n , i.e., the fraction of each cell occupied by fluid rather than particles;
 - 2: Calculate the interphase transfer terms per particle $G_{g \rightarrow pk}^{m,n}$, $G_g^{m,n}$, $F_{g \rightarrow p}^{H,n}$, $G_{g \rightarrow p}^{H,n}$, $\mathbf{F}_{g \rightarrow p}^{L,n}$, $\mathbf{G}_{g \rightarrow p}^{L,n}$;
 - 3: Calculate the fluid interphase transfer terms $\mathcal{G}_{p \rightarrow gk}^{\rho,n}$, $\mathcal{G}_{p \rightarrow g}^{\rho,n}$, $\mathcal{G}_{p \rightarrow g}^{h,n}$, $\mathcal{G}_{p \rightarrow g}^{\ell,n}$, as well as the drag coefficient β^n , the drag related transfer quantity $(\beta \mathbf{u}_s)^n$, the heat transfer convection coefficient Γ^n , and the quantity $(\Gamma T_s)^n$ related to the convective heat transfer;
 - 4: Update the fluid phase (see Algorithm 2 below);
 - 5: Update the particle phase (see Algorithm 3 below).
-

3.1. Fluid phase update

We break the fluid phase update into two stages – first the update of density, species and enthalpy, followed by the update of velocity. The fluid phase time integration method is outlined below in Algorithm 2. We stress that during the fluid phase update from time t^n to t^{n+1} , particle quantities are frozen at their t^n state.

The construction of the advective terms for fluid species, $A_{Xk}^{n+1/2}$, enthalpy, $A_h^{n+1/2}$, and velocity, $\mathbf{A}_u^{n+1/2}$, follows a Godunov approach. These terms are computed as:

$$A_{Xk}^{n+1/2} = \nabla \cdot \left(\hat{\varepsilon}_g \hat{\rho}_g^{n+1/2} \hat{X}_{gk}^{n+1/2, \text{MAC}} \mathbf{u}_g^{n+1/2, \text{MAC}} \right), \quad A_h^{n+1/2} = \nabla \cdot \left(\hat{\varepsilon}_g \hat{\rho}_g^{n+1/2} \hat{h}_g^{n+1/2} \mathbf{u}_g^{n+1/2, \text{MAC}} \right),$$

and $\mathbf{A}_u^{n+1/2} = \left(\hat{\varepsilon}_g \hat{\mathbf{u}}_g^n \right) \cdot \nabla \mathbf{u}_g^{n+1/2, \text{MAC}},$

where, by the hat notation we indicate fluid variables computed at the faces of the grid cells. In the Godunov approach, the fluid quantities are predicted from cell centroids at time t^n to face centroids at $t^{n+1/2}$, and fluxes are constructed using these states and an intermediate velocity field, $\mathbf{u}_g^{n+1/2, \text{MAC}}$. This differs from the Method-of-Lines approach described in [28], and it is described in detail in Appendix B.

Algorithm 2 Update of the fluid phase

1: Update fluid mass density, ρ_g , per eq. (2),

$$\rho_g^{n+1} = \rho_g^n + \frac{\Delta t}{\varepsilon_g^n} \left(-\sum_k A_{Xk}^{n+1/2} + \mathcal{G}_{p \rightarrow g}^{\rho,n} \right).$$

2: Update species mass fractions, X_{gk} , per Eq. (4). By including the diffusive term implicitly, solve

$$\left(1 - L_X^D\right) X_{gk}^{n+1,*} = \frac{\rho_g^n}{\rho_g^{n+1}} X_{gk}^n + \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1}} \left(-A_{Xk}^{n+1/2} + \mathcal{G}_{p \rightarrow gk}^{\rho,n} \right), \quad \forall k = 1, \dots, N_g,$$

where

$$L_X^D \left[X_{gk}^{n+1,*} \right] = \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1}} \nabla \cdot \left(\hat{\varepsilon}_g^n \mathcal{J}_{gk}^{n+1,*} \right), \quad \text{and} \quad \mathcal{J}_{gk}^{n+1,*} = \hat{\rho}_g^{n+1} \hat{D}_g \nabla X_{gk}^{n+1,*}.$$

Finally, enforce the solution to belong to the closed interval $[0, 1]$ by setting $X_{gk}^{n+1,*} = \max(0, \min(1, X_{gk}^{n+1,*}))$ and update the species mass fraction through normalization:

$$\text{define } X_{g,\text{sum}}^* := \sum_{j=1}^{N_g} X_{gj}^{n+1,*}, \quad \text{then } X_{gk}^{n+1} = \frac{X_{gk}^{n+1,*}}{X_{g,\text{sum}}^*}, \quad \forall k = 1, \dots, N_g.$$

3: Update fluid specific enthalpy, h_g , and temperature, T_g , in three stages. First we define an explicit auxiliary enthalpy h_g^{**} as

$$h_g^{**} = \frac{\rho_g^n}{\rho_g^{n+1}} h_g^n + \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1}} \left(\frac{\varepsilon_g^n \bar{S}^n}{\bar{\theta}^n} - A_h^{n+1/2} + \nabla \cdot \left(\hat{\varepsilon}_g^n \sum_k \hat{h}_{gk}^n \mathcal{J}_{gk}^{n+1,*} \right) + \mathcal{G}_{p \rightarrow g}^{h,n} \right),$$

where \bar{S}^n and $\bar{\theta}^n$ are defined in Appendix A. Then, the diffusive term is included implicitly and $T_g^{n+1,*}$ is solved iteratively using Newton's method. Specifically, set $T_g^{n+1,0} = T_g^n$, and obtain the updated temperature $T_g^{n+1,k+1}$ by solving

$$\left(c_{p,g}(T_g^{n+1,k}) - L_h^D \right) T_g^{n+1,k+1} = c_{p,g}(T_g^{n+1,k}) T_g^{n+1,k} - h_g(T_g^{n+1,k}) + h_g^{**},$$

where

$$L_h^D \left[T_g^{n+1,k+1} \right] = \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1}} \nabla \cdot \left(\hat{\varepsilon}_g^n \hat{\kappa}_g \nabla T_g^{n+1,k+1} \right).$$

Update the specific enthalpy $h_g^{n+1,*} = h_g(T_g^{n+1,*})$ per eq. (6) with the intermediate temperature. Then, include the interphase convective heat transfer term implicitly and apply Newton's method again to get T_g^{n+1} . Specifically, set $T_g^0 = T_g^{n+1,*}$ and iteratively solve

$$\left(c_{p,g}(T_g^{n+1,k}) + \frac{\Delta t \Gamma^n}{\varepsilon_g^n \rho_g^{n+1}} \right) T_g^{n+1,k+1} = c_{p,g}(T_g^{n+1,k}) T_g^{n+1,k} - h_g(T_g^{n+1,k}) + h_g^{n+1,*} + \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1}} (\Gamma T_s)^n.$$

Finally, update the specific enthalpy again with the final temperature T_g^{n+1} .

4: Update momentum to get an approximated velocity field \mathbf{u}_g^{**} per eq. (3),

$$\left(1 - L_u^D\right) \mathbf{u}_g^{n+1,*} = \mathbf{u}_g^n + \Delta t \mathbf{g} - \frac{\Delta t \mathbf{A}_u^{n+1/2}}{\varepsilon_g^n} + \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1/2}} \left(-\varepsilon_g^n \mathbf{G}_p^{n-1/2} + \mathcal{G}_{p \rightarrow g}^{\ell,n}\right),$$

where $\rho_g^{n+1/2} = 0.5 \cdot (\rho_g^n + \rho_g^{n+1})$, and

$$L_u^D \left[\mathbf{u}_g^{n+1,*}\right] = \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1}} \nabla \cdot \boldsymbol{\tau}_g^{n+1,*}.$$

Finally, obtain a provisional velocity field $\mathbf{u}_g^{n+1,*}$ at t^{n+1} , by implicitly including the interphase momentum transfer from fluid-particle interfacial drag:

$$\left(1 + \frac{\Delta t \beta^n}{\varepsilon_g^n \rho_g^{n+1/2}}\right) \mathbf{u}_g^{n+1,*} = \mathbf{u}_g^{**} + \frac{\Delta t}{\varepsilon_g^n \rho_g^{n+1/2}} (\beta \mathbf{u}_s)^n.$$

5: Project the provisional velocity field to enforce the divergence constraint by solving the following equation in $\mathbf{G}_p^{n+1/2}$

$$\nabla \cdot \left(\frac{\Delta t \varepsilon_g^n}{\rho_g^{n+1/2}} \mathbf{G}_p^{n+1/2}\right) = \nabla \cdot \left(\varepsilon_g^n \mathbf{u}_g^{n+1,*} + \frac{\varepsilon_g^n \Delta t}{\rho_g^{n+1/2}} \mathbf{G}_p^{n-1/2}\right) - S^{n+1}.$$

Here, S^{n+1} is the right-hand-side of the constraint defined in Eq. (9). Then, define the new-time velocity field as

$$\mathbf{u}_g^{n+1} = \mathbf{u}_g^{n+1,*} - \frac{\Delta t}{\rho_g^{n+1/2}} \mathbf{G}_p^{n+1/2}.$$

3.2. Solids phase update

Many of the details of the particle update have been previously published in [26] which are now extended to include species and enthalpy. Therefore, only an overview of the advancement algorithm is presented here. Particle quantities are updated only after the advance of the fluid phase is complete. For each fluid time step, Δt , the particles are advanced multiple substeps at $\Delta t_{\text{DEM}} < \Delta t$. Δt_{DEM} is initialized as a fraction of the collisional time scale τ_{coll} , [e.g., see 15, 17] and adjusted to give an integer number of particle substeps for each fluid step. Substepping uses a simple first-order Euler method. Typically, Δt_{DEM} is much smaller than Δt in low Mach number flows (see [28]).

The particle velocity update includes particle-wall interactions, particle-particle collisions, and interphase transfer terms using fluid quantities interpolated from the grid to the particle positions. Mass transfer drives the update of particle species mass fractions due to heterogeneous chemical reactions. Once the species mass fractions have been updated, a normalization algorithm is applied to guarantee that $\sum_k X_{pk} = 1$ when exiting the particle update. Finally, the particle specific enthalpies are updated to account for heat transfer due to convective processes and interphase chemical reactions. Then, particle temperatures are updated by implicitly solving eq. (16) using a Newton-Raphson algorithm, as described in Algorithm 3.

3.3. Eulerian-Lagrangian coupling

In CFD-DEM multiphase flow applications, some physical quantities are transferred from the fluid phase to the solids phase and vice-versa. The conversion of these quantities between the (fluid) Eulerian grid and the (solid) Lagrangian points is computed through specific transfer kernels. As discussed in Section 2.4, in MFIX-Exa, we define an interpolation kernel, χ_p , for transferring fluid phase quantities from a compact stencil, S_p , made of 2^3 cells in the Eulerian grid, to the p -th particle position, see Section 3.3 for example. On the other hand, we define a deposition

Algorithm 3 Update of the particle phase

1: Update particle mass, m_p , per eq. (11)

$$m_p^{n+1} = m_p^n + \Delta t_{\text{DEM}} G_{g \rightarrow p}^{m,n}.$$

2: Update species mass fractions, X_{pk} , per eq. (14),

$$X_{pk}^{n+1,*} = \frac{m_p^n}{m_p^{n+1}} X_{pk}^n + \frac{\Delta t_{\text{DEM}}}{m_p^{n+1}} \left(G_{g \rightarrow pk}^{m,n} \right), \quad \forall k = 1, \dots, N_s.$$

Then, by computing $X_{pk}^{n+1,*} = \max(0, \min(1, X_{pk}^{n+1,*}))$, enforce $X_{pk}^{n+1,*}$ to belong to the closed interval $[0, 1]$, and update species mass fractions through normalization of $X_{pk}^{n+1,*}$:

$$\text{define } X_{p,\text{sum}}^* := \sum_{j=1}^{N_s} X_{pj}^{n+1,*}, \quad \text{then } X_{pk}^{n+1} = \frac{X_{pk}^{n+1,*}}{X_{p,\text{sum}}^*}, \quad \forall k = 1, \dots, N_s,$$

3: Update particle specific enthalpy, h_p , per eq. (15),

$$h_p^{n+1} = \frac{m_p^n}{m_p^{n+1}} h_p^n + \frac{\Delta t_{\text{DEM}}}{m_p^{n+1}} \left(F_{g \rightarrow p}^{H,n} + G_{g \rightarrow p}^{H,n} \right),$$

then compute the particle mixture temperature T_p by solving the implicit eq. (16) using Newton's method. First define $T_p^{n+1,0} = T_p^n$, then iteratively solve

$$c_{p,p}(T_p^{n+1,k}) T_p^{n+1,k+1} = c_{p,p}(T_p^{n+1,k}) T_p^{n+1,k} + \left(h_p^{n+1} - h_p(T_p^{n+1,k}) \right),$$

4: Update particle velocity \mathbf{u}_p , per eq. (12),

$$\mathbf{u}_p^{n+1} = \frac{m_p^n}{m_p^{n+1}} \mathbf{u}_p^n + \frac{\Delta t_{\text{DEM}}}{m_p^{n+1}} \left(m_p^n \mathbf{g} + \mathbf{F}_{pw}^n + \sum_{q=1}^{N_{\text{ngb}}} \mathbf{F}_{pq}^n + \mathbf{F}_{g \rightarrow p}^{L,n} + \mathbf{G}_{g \rightarrow p}^{L,n} \right),$$

and then particle position $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t_{\text{DEM}} \mathbf{u}_p^{n+1}$.

5: Update particle angular velocity $\boldsymbol{\omega}_p$, per eq. (13),

$$\boldsymbol{\omega}_p^{n+1} = \frac{I_p^n}{I_p^{n+1}} \boldsymbol{\omega}_p^n + \frac{\Delta t_{\text{DEM}}}{I_p^{n+1}} \left(\mathbf{T}_{pw}^n + \sum_{q=1}^{N_{\text{ngb}}} \mathbf{T}_{pq}^n \right).$$

kernel, \mathcal{K} , for transferring particle phase quantities from their position to the centroids of the fluid phase cells they occupy, see [28].

Considering the p -th particle position $\mathbf{x}_p = (x_p, y_p, z_p)$, the upper indices (i, j, k) of the S_p stencil are:

$$i = \left\lfloor \frac{x_p - D_{low,x}}{\Delta x} + 0.5 \right\rfloor, \quad j = \left\lfloor \frac{y_p - D_{low,y}}{\Delta y} + 0.5 \right\rfloor, \quad \text{and} \quad k = \left\lfloor \frac{z_p - D_{low,z}}{\Delta z} + 0.5 \right\rfloor, \quad (21)$$

where Δx , Δy , and Δz are the grid spacing in each of the three dimensions, and $D_{low,x}$, $D_{low,y}$, and $D_{low,z}$ are the physical locations of the lower side of the domain.

When all the cells for the interpolation or deposition are regular, i.e., they do not contain any embedded boundary cells, we use standard trilinear kernels for transferring the different physical quantities. On the other hand, close to the

$$S_p = \begin{cases} 0 : (i-1, j-1, k-1) \\ 1 : (i, j-1, k-1) \\ 2 : (i, j, k-1) \\ 3 : (i-1, j, k-1) \\ 4 : (i-1, j-1, k) \\ 5 : (i, j-1, k) \\ 6 : (i, j, k) \\ 7 : (i-1, j, k) \end{cases}$$

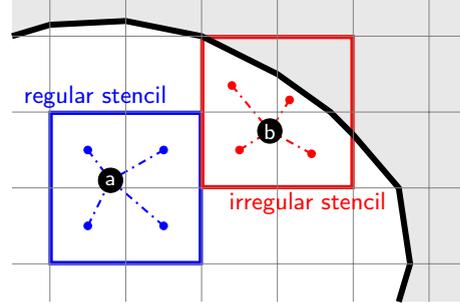


Figure 1: On the left: cell indexing (from 0 to 7) for the 2^3 -cells compact stencil used for the interpolation kernel, χ_p , that maps fluid (Eulerian) quantities to solid (Lagrangian) positions. Indexes i , j , and k are defined as in (21). On the right: a bi-dimensional representation of a regular stencil (in blue) and an irregular stencil (in red), including cut-cells. The black circles (a) and (b) represent particles, while dot-dashed lines represent the distances of the particles from the centroids of the cells. The solid black line represents the embedded boundaries.

embedded boundaries, the approach is more complex when any of the cells are cut or covered, and we use nonlinear transfer kernels. We refer to [28] for a detailed description of these different situations.

4. Results and discussion

This section presents and discusses some numerical results obtained with MFIX-Exa. The problems addressed in the following subsections represent benchmarks that assess the performance, the correctness of our model, and the accuracy of our implementation.

The results are reported here in different categories. First, we present three *verification* tests: (i) transient heat conduction in packed beds, (ii) water droplet evaporating in a humidified air flow, and (iii) water vapor condensing on a water droplet. Then, we discuss two *validation* tests: (i) a cooling fluidized bed and a (ii) CO₂ capture in a fluidized bed. Finally, we assess MFIX-Exa's performance with a weak scaling analysis and by progressively adding physical and numerical complexity at a single scale.

4.1. Verification

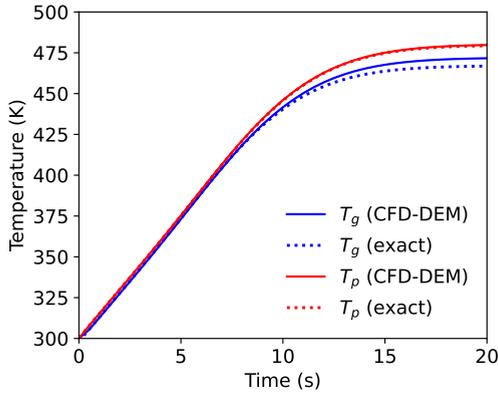
4.1.1. Thermal step: transient heat conduction in packed beds

By assuming a purely 1-D system, Schumann [36] derived an analytical solution to continuum equations for transient heat conduction in packed beds. Standing for nearly one hundred years, this benchmark solution has been used many times (see references in [34]). Recently, the Schumann solution was revisited and expanded by Salehi et al. [34] who used Laplace transforms to derive a transient solution including a uniform particle volumetric heat source.

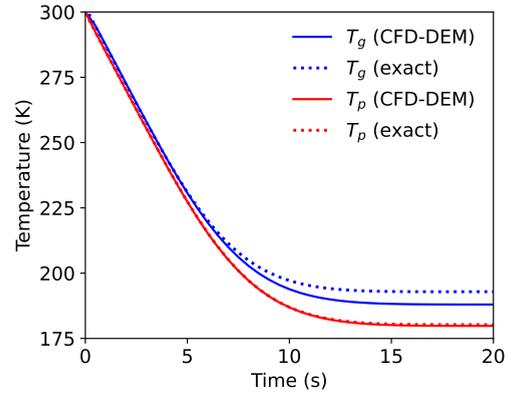
The same problem setup and flow conditions which were originally used to benchmark a CFD-DEM code [34] are applied here. The particle size, density, and specific heat are set to $d_p = 2.2$ cm, $\rho_p = 1000$ kg/m³, and $c_{p,p} = 5$ J/(kg K). The fluid is assumed to be incompressible with the following properties: density $\rho_g = 1.16$ kg/m³, viscosity $\mu_g = 1.8 \times 10^{-5}$ Pa s, specific heat $c_{p,p} = 1007$ J/(kg K) and heat conductivity $\kappa_g = 0.026$ W/(m K). The gas and particles are coupled through the Gunn [19] interfacial heat transfer correlation.

The domain is $8d_p$ in width and depth and $32d_p$ tall in the flow direction which is bounded by mass inflow and pressure outflow boundary conditions. The lateral dimensions are periodic. A uniform $\Delta x = d_p$ fluid mesh is applied with one particle per cell. The positions of the particles are held fixed throughout the simulation, by decoupling the DEM momentum update. The inlet gas velocity is $u_{in} = 0.1$ m/s. Due to the uniform conditions, the initial gas velocity is $u_0 = u_{in}/\phi_0$ where $\phi_0 = \pi/6$. The initial temperature is $T_0 = 300$ K for both the fluid and the particles. The inlet temperature is $T_{in} = 330$ K, giving rise to a thermal step which propagates throughout the domain.

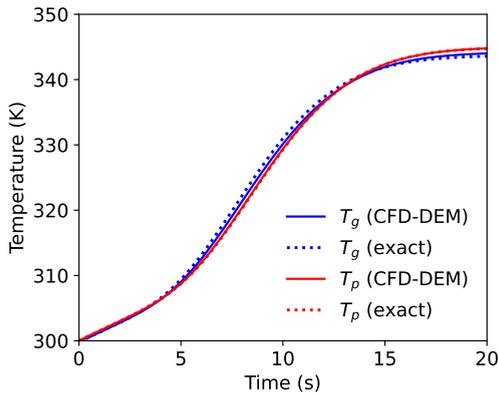
Four enthalpy sources are considered: a strong source $q = +0.5$ J/(kg s), a strong sink $q = -0.5$ J/(kg s), a weak source $q = +0.05$ J/(kg s), and a weak sink $q = -0.05$ J/(kg s). The temperature evolution is monitored just below the center of the domain in the streamwise direction (precisely at the center of the 16th row of the fluid mesh) and is averaged over the spanwise directions. The fluid and particle responses are compared to the exact solutions for



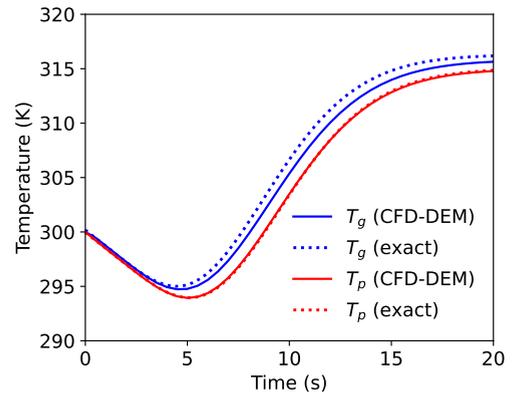
(a) Strong enthalpy source: $q = +0.5 \text{ J}/(\text{kg K})$



(b) Strong enthalpy sink: $q = -0.5 \text{ J}/(\text{kg K})$



(c) Weak enthalpy source: $q = +0.05 \text{ J}/(\text{kg K})$



(d) Weak enthalpy sink: $q = -0.05 \text{ J}/(\text{kg K})$

Figure 2: Analytical (exact) and numerical (CFD-DEM) profiles of the fluid phase temperature T_g , and the solids phase temperature T_p , for four test cases of transient heat conduction in packed beds with different enthalpy sources applied to the particles: (a) strong enthalpy source, (b) strong enthalpy sink, (c) weak enthalpy source, (d) weak enthalpy sink.

the four conditions in Fig. 2. The predicted solutions match the analytical solutions well in all four regimes. Similar results (not shown) are also found at various elevations throughout the bed. Close examination shows that, compared to the high level of agreement with the particle temperature, there is some disagreement in the gas-phase temperature. The CFD-DEM results show a slight bias in all cases, predicting a smaller temperature difference, $|T_p - T_g|$, than the analytical solutions. This bias suggests that the discrepancy arises from the interfacial transfer as the particles are the enthalpy source (or sink). It is important to note that the exact solution is an analytical solution to continuum equations, which is not an exact representation of the CFD-DEM model. Specifically, there is no transfer kernel. The CFD-DEM code of Salehi et al. [34] showed a similar bias, and the authors reached the same conclusion.

4.1.2. Water droplet evaporation

In this benchmark, we follow [29] and study a water droplet evaporating in a humidified airflow. We consider the water droplet, modeled as a particle, to be suspended in a relatively large cubic domain such that $\varepsilon_g \approx 1$. Also, we assume liquid water, $\text{H}_2\text{O}(l)$, to evaporate from the droplet into water vapor, $\text{H}_2\text{O}(g)$.

In the simulation, the droplet position is fixed inside a humidified air stream. The initial droplet temperature is equal to the airflow temperature, and different initial temperatures are chosen for each test. The rate of reaction in units of molar mass per time (mol/s) is

$$\mathcal{R} = \max \left\{ 0, A_p \Gamma_{\text{H}_2\text{O}} (C_p - C_g) \right\},$$

	Air	Vapor	Liquid
a_0 (mol kg ⁻¹)	3.568	4.198×10^1	7.256×10^1
a_1 (mol kg ⁻¹ K ⁻¹)	-6.787×10^{-4}	-2.036×10^{-2}	-6.625×10^{-1}
a_2 (mol kg ⁻¹ K ⁻²)	1.554×10^{-6}	6.52×10^{-5}	2.562×10^{-3}
a_3 (mol kg ⁻¹ K ⁻³)	-3.299×10^{-12}	5.488×10^{-8}	-4.366×10^{-6}
a_4 (mol kg ⁻¹ K ⁻⁴)	-4.664×10^{-13}	1.772×10^{-11}	2.782×10^{-9}
a_5 (mol kg ⁻¹ K)	-1.062×10^3	-3.029×10^5	-4.189×10^4

Table 1

Species-specific coefficients we used for computing specific heat, c_p , and specific enthalpy, h , for the fluid and solid phases according to the "NASA Polynomials model", see [8].

where A_p is the droplet surface area, and $\Gamma_{\text{H}_2\text{O}}$ is the mass transfer coefficient, and C_p and C_g , are the molar concentrations of liquid water and water vapor at the droplet's surface, respectively C_p and C_g , defined as

$$C_p = \frac{P_{\text{sat}}(T_p)}{RT_g}, \quad \text{and} \quad C_g = \frac{Y_{g,\text{H}_2\text{O}(g)} p_g}{RT_g},$$

where p_g is the gas pressure, $Y_{g,\text{H}_2\text{O}(g)}$ is water vapor mole fraction, $P_{\text{sat}}(T_p)$ is the saturation pressure of water vapor calculated at the droplet's temperature from [33], and R is the gas constant. The mass transfer coefficient is calculated from

$$\Gamma_{\text{H}_2\text{O}} = \frac{N_{\text{Sh}} D_{\text{H}_2\text{O,Air}}}{d_p},$$

where N_{Sh} is the particle Sherwood number, $D_{\text{H}_2\text{O,Air}}$ is the diffusion coefficient of water vapor in air [27], and d_p is the droplet diameter. Following Musser et al. [29], the Sherwood number correlation of Ranz and Marshall [32] is applied here.

The domain is a cube with 1 cm long edges and is meshed with $5 \times 5 \times 5$ cubic cells. The inlet gas velocity is 3.0 m/s, and atmospheric pressure enforced at the outlet. The remaining faces are periodic. The initial pressure is also set to be equal to the atmospheric pressure. The density and droplet diameter are 958.6 kg/m^3 and $1 \times 10^{-3} \text{ m}$, respectively. The specific heat and heat of formation for the three species are defined according to [8] and are computed as follows

$$c_p(T) = a_0 \cdot R + a_1 \cdot R \cdot T + a_2 \cdot R \cdot T^2 + a_3 \cdot R \cdot T^3 + a_4 \cdot R \cdot T^4,$$

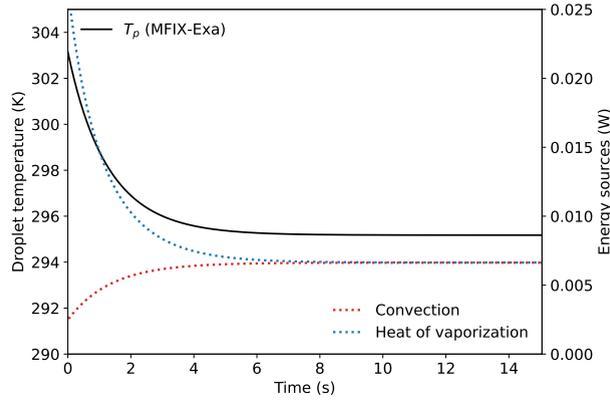
$$h(T) = a_0 \cdot R \cdot T + \frac{a_1 \cdot R \cdot T^2}{2} + \frac{a_2 \cdot R \cdot T^3}{3} + \frac{a_3 \cdot R \cdot T^4}{4} + \frac{a_4 \cdot R \cdot T^5}{5} + a_5 \cdot R,$$

where the species-specific coefficients a_i , with $i = 0, \dots, 5$, are listed in Section 4.1.2.

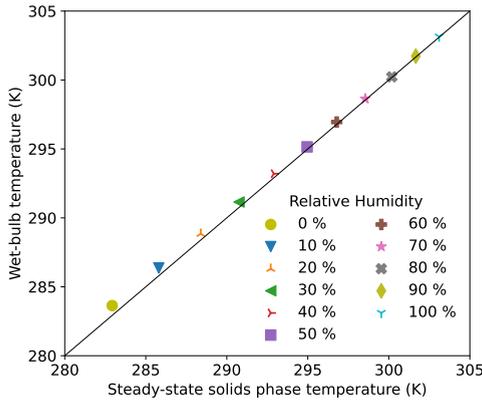
For each test, we ran 15 s of physical time, which is sufficient to reach a steady-state [29]. In Fig. 3a we plot the droplet temperature and enthalpy sources (convection and heat of vaporization) over time. We observe that the enthalpy sources converge to the same value after a few seconds, and the droplet temperature decreases until it reaches a steady state when the rate of enthalpy loss due to the reaction equals the rate of enthalpy gain due to convective heat transfer. Because the steady droplet temperature is a function of the air temperature and relative humidity [29], we ran eleven combinations of the gas stream temperature and relative humidity for testing the model. The steady-state droplet temperature is analyzed in Fig. 3b via parity plots comparing the calculated result against the theoretical wet-bulb temperature [35]. The largest difference, less than 1 K representing a relative error of just 0.25%, is observed for the 0% humidity case. Consistent with [29], the error decreases with increasing relative humidity.

4.1.3. Water vapor condensation

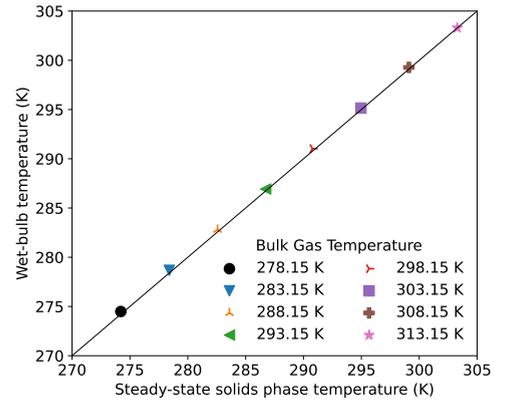
Now, we consider the reverse case in which water vapor condenses out of the free stream onto the droplet. Again, the droplet is modeled by a single particle suspended in a relatively large cubic domain of humidified air such that



(a) Convection and heat of vaporization drive the droplet temperature over time.



(b) Wet-bulb and droplet steady-state temperatures increase with relative humidity.



(c) Wet-bulb and droplet steady-state temperatures increase with bulk-gas temperature.

Figure 3: Results of the droplet evaporation test. In order: (a) plot of the droplet temperature over time; (b) plot of the wet-bulb temperature given a steady-state solids phase temperature for different values of relative humidity; (c) plot of the wet-bulb temperature given a steady-state solids phase temperature for different values of bulk gas temperature.

$\varepsilon_g \approx 1$. Furthermore, we assume that water vapor in the fluid condenses onto the droplet. By precluding evaporation and any other chemical reactions, mass transfer between the fluid and droplet are equal and opposite.

$$\underbrace{G_{g \rightarrow p, H_2O(l)}^m > 0}_{\text{production of liquid water}}, \quad \underbrace{G_{p \rightarrow g, H_2O(g)}^m < 0}_{\text{consumption of water vapor}}, \quad G_{g \rightarrow p, H_2O(l)}^m = -G_{p \rightarrow g, H_2O(g)}^m, \quad (22)$$

The rate of reaction in units of molar mass per time (mol/s) is

$$\mathcal{R} = \max \left\{ 0, A_p \Gamma_{H_2O} C_{gm, H_2O} \right\}.$$

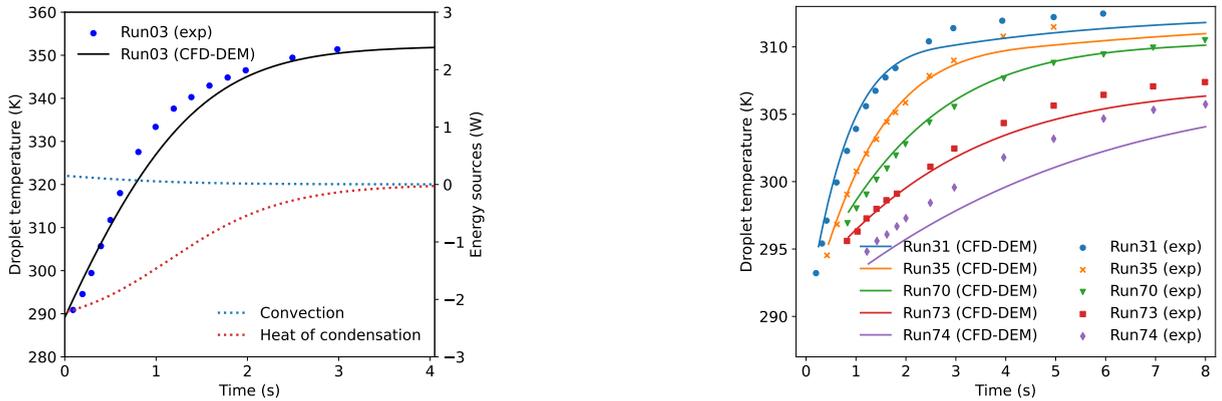
where C_{gm, H_2O} is the water vapor concentration difference, defined as

$$C_{gm, H_2O(g)} = \frac{\rho_g \left(X_{g, H_2O(g)} - X_{m, H_2O(g)} \right)}{W_{g, H_2O(g)}},$$

initial values	Run03	Run31	Run35	Run70	Run73	Run74
d_p (m)	0.0029	0.0013	0.0018	0.0025	0.0032	0.0040
T_p (K)	289.15	289.95	289.65	291.95	292.35	290.35
T_g (K)	353.5	312.5	312.5	312.5	309.6	309.6
u_g (m/s)	1.91	3.49	3.49	3.51	4.97	4.97
$X_{\text{H}_2\text{O}(g)}$	0.3555	0.0382	0.0382	0.0382	0.0317	0.0317
X_{air}	0.6445	0.9618	0.9618	0.9618	0.9683	0.9683

Table 2

Initial and boundary conditions for the vapor condensation simulations taken from [22]. In all the tests, the gas pressure is assumed to be equal to the atmospheric pressure 1.01325×10^5 Pa.



(a) Convection and heat of vaporization drive the droplet temperature over time for Run03.

(b) Droplet temperature evolution for different runs, from Run31 to Run74.

Figure 4: Experimental (exp) and numerical (CFD-DEM) results of the vapor condensation test. In order: (a) plot of the droplet temperature over time for Run03; (b) plot of the droplet temperature over time for Run31, Run35, Run70, Run73, and Run74.

where $X_{m,\text{H}_2\text{O}(g)}$ is the mass fraction of saturated water vapor at the particles surface

$$X_{m,\text{H}_2\text{O}(g)} = \frac{W_{\text{H}_2\text{O}(g)} N_{\text{H}_2\text{O}(g)}}{W_{\text{H}_2\text{O}(g)} N_{\text{H}_2\text{O}(g)} + W_{\text{air}} (1 - N_{\text{H}_2\text{O}(g)})},$$

and $N_{\text{H}_2\text{O}(g)} = P_{\text{sat}}(T_p)/p_g$ is the mole fraction of saturated water vapor at particle surface. Here, p_g is the gas pressure, and $P_{\text{sat}}(T_p)$ is the saturation pressure of water vapor calculated at the droplet temperature from [33].

Six different conditions are considered, labeled as “Run03”, “Run31”, “Run35”, “Run70”, “Run73”, and “Run74”. In each case, the initial conditions are described in Section 4.1.3 and are analogous to the experiments discussed in [22]. Fig. 4a shows the experimental and numerical values for the droplet temperature over time for Run03, together with the heat sources that drive the droplet temperature, i.e., convective heat transfer and heat of condensation. The heat of condensation is the source with the more significant role in the droplet temperature rise. The remaining cases are compared to the corresponding experimental data in Fig. 4b. In some cases we observe non-negligible differences in the computed temperature values with respect to the data, particularly in Run74. Generally, the predictions improve as the condensation source driving the temperature rise increases. As discussed by [29], these discrepancies may be due an overestimation of the mass transfer rates, inaccurate gas temperature values in the experiment, and the fact that the air/steam mixture is assumed to be saturated.

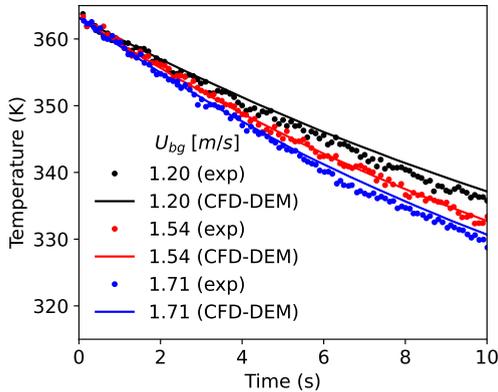
4.2. Validation

4.2.1. Cooling fluidized bed

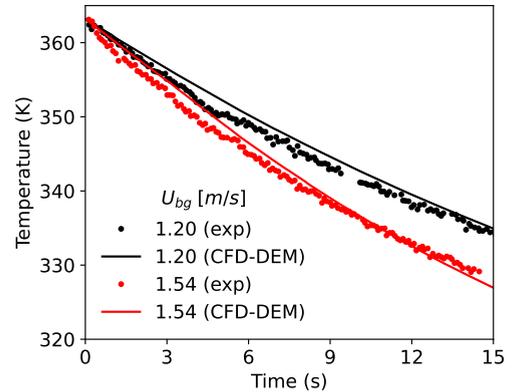
The bench-scale fluidized bed of Patil et al. [31] is a test with a simple rectangular geometry having width, depth and height equal to 80, 15, and 250 mm, respectively. The particles are 1.0 mm glass beads of density $\rho_p = 2500 \text{ kg/m}^3$. The specific heat is taken to be $c_{p,p} = 840 \text{ J/K/kg}$. Two bed masses were investigated, $M = 75 \text{ g}$ and 125 g , which correspond to approximately $N_p = 57300$ and 95500 particles, assumed here to be perfectly monodispersed.

The domain is discretized by a uniform 2.0 mm fluid mesh. Along the depth of the bed, one-quarter of the first and last cells are cut by the front and back walls (using embedded boundaries). The experimental device included a central, round jet of diameter $D_j = 13 \text{ mm}$. The central jet was not operated during the relevant experiments; however, this defect in the uniform inlet could affect the flow and should be modeled. An inactive jet region is modeled as an 8×4 parts of fluid cells centered in the inflow plane with zero velocity. This approach gives an effective area approximately 4% smaller than the actual circular jet present in the experimental device. The percentage of the remaining inlet area with respect to the total cross-sectional area, $a_{in} = 1 - A_j/L_x L_z$, is approximately 90%.

The fluidizing gas is nitrogen modeled as an ideal gas with the following properties: molecular weight $W_g = 28.014 \text{ g/mol}$, density $\rho_g = 1.5 \text{ kg/m}^3$, viscosity $\mu_g = 1.75 \times 10^{-5} \text{ Pa s}$, specific heat $c_{p,p} = 1041 \text{ J/(kg K)}$ and thermal conductivity $\kappa_g = 0.0254 \text{ W/(m K)}$. The bed is operated at three “background” velocities, $U_{bg} = 1.2 \text{ m/s}$, 1.54 m/s , and 1.71 m/s where U_{bg} refers to a superficial velocity in the absence of the jet, i.e., the inlet velocity is prescribed by $u_{in} = U_{bg}/a_{in} = 1.343 \text{ m/s}$, 1.724 m/s , and 1.914 m/s . (Note that the larger bed mass did not operate at the highest flow condition.) The inlet gas temperature is $T_{g,in} = 293.15 \text{ K}$. The initial gas temperature in the bed, $T_{g,0}$, and the (constant) embedded boundary temperature, T_w , are also set to 293.15 K .



(a) “Patil light”: bed mass is equal to 75 g.



(b) “Patil heavy”: bed mass is equal to 125 g.

Figure 5: Experimental (exp) and numerical (CFD-DEM) results of cooling fluidized beds with the same settings as in Patil et al. [31]. two different bed masses and three different “background” velocities are considered: (a) the bed has mass 75 g and it is operated at 1.2 m/s, 1.54 m/s, and 1.71 m/s background velocities; (b) the bed has mass 125 g and it is operated at 1.2 m/s, and 1.54 m/s background velocities.

In the experiments, the particles are heated in an oven, discharged into the bed, and immediately fluidized by the colder nitrogen gas. Numerically, this is modeled simply with a 2^3 particle packing per fluid cell with initial bed heights of $h_0 = 54 \text{ mm}$ ($M = 75 \text{ g}$) and $h_0 = 88 \text{ mm}$ ($M = 125 \text{ g}$). The particles are initialized at $T_{p,0} = 363.15 \text{ K}$. The fluid and the particles are coupled with the Gidaspow [23] correlation for momentum and the Gunn [19] correlation for energy. The particles cool as the bed is fluidized, as measured by infrared (IR) imaging. The bed-averaged mean particle temperature is the most important quantity of interest and perhaps the most reliable measurement. The MFIX-Exa simulations are compared to the data in Fig. 5. Generally, we see very good agreement between simulation and experiment at all flow rates and at both bed loadings.

4.2.2. CO₂ capturing fluidized bed

The validation dataset of Li et al. [24] (also see the thesis of Janssen [21]) is quite similar to the cooling fluidized bed of Patil et al. [31]. In fact, the same bench-scale system is used, a simple rectangular bed 80 mm wide, 15 mm

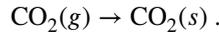
	20% CO ₂	40% CO ₂	60% CO ₂	80% CO ₂	100% CO ₂
Y_{CO_2}	0.20	0.40	0.60	0.80	1.00
q_e (g/g)	0.10	0.13	0.15	0.16	0.17

Table 3

Equilibrium adsorption, q_e , of CO₂ onto $d_p = 1.9$ mm zeolite 13X particles as a function of CO₂ mole fraction, Y_{CO_2} , in a N₂-CO₂ mixture at atmospheric pressure [21].

deep, and 250 mm long. The unused jet is not mentioned in the reference work [24] and, therefore, not modeled in this test. The most important change is the material: rather than glass beads, the bed is now filled with zeolite particles that are highly porous and can adsorb gaseous CO₂. The zeolite particles are modeled with the following properties: diameter $d_p = 1.9$ mm, density $\rho_p = 1100$ kg/m³, and specific heat $c_{p,p} = 760$ J/K/kg. The bed mass is $M = 40$ g, approximately $N_p = 10125$ particles. Because the particle diameter has effectively doubled, the fluid mesh is coarsened by a factor of two relative to the cooling fluidized bed of Patil et al. [31].

The fluidizing gas is a mixture of nitrogen and carbon dioxide. The species dependent properties are $W_{g,\text{N}_2} = 28.013$ g/mol, $W_{g,\text{CO}_2} = 44.01$ g/mol, $c_{p,\text{N}_2} = 1041$ J/(kg K), and $c_{p,\text{CO}_2} = 851$ J/(kg K). The mixture is modeled as an ideal gas with density $\rho_g = 1.13$ kg/m³, viscosity $\mu_g = 1.782 \times 10^{-5}$ Pa s, heat conductivity $\kappa_g = 0.0259$ W/(m K) and species diffusivity $D_g = 16 \times 10^{-6}$ m²/s. The single, trivial chemical reaction is:



The reverse reaction, i.e., desorption where CO₂ is released from the zeolite pores back into the gas, is not considered. The “solid” CO₂(s), i.e., the carbon dioxide trapped in the pores, is given the same properties as the zeolite so that the particle properties are independent of the amount of adsorbed CO₂. In this case, the heat of formation of CO₂(s) is actually the heat of adsorption, $\Delta H_{\text{ads}} = 30$ kJ/mol. Although there is significant room for improvement, we chose to apply the n^{th} -order reaction model developed and originally applied by [24]:

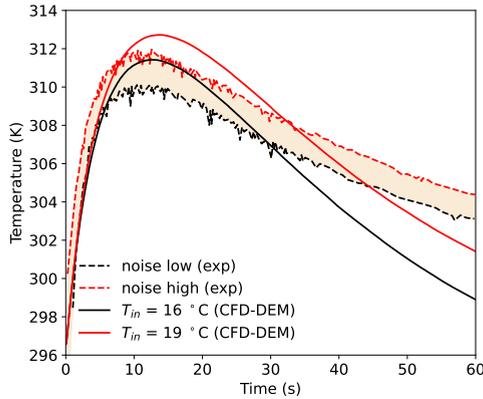
$$\frac{dq}{dt} = k (q_e - q)^n, \quad \text{and} \quad k = k_0 e^{-E_a/RT}, \quad (23)$$

where q is the amount of CO₂ adsorbed onto the particle (in g/g), q_e is the equilibrium adsorption, i.e., the maximum amount of CO₂ that a particle can adsorb at a given pressure and temperature. The order of the reaction, $n = 2.47$, was determined from a best-fit of thermogravimetric analysis (TGA) data. The pre-exponent of the Arrhenius equation, $k_0 = 1.5 \times 10^{-5}$ s⁻¹, was also data-fitted. An activation energy of $E_a = -29.38$ kJ/mol is taken from the open literature [12] and $R = 8.3145$ J/(mol K) is the universal gas constant. The equilibrium adsorption is also determined from TGA data by Janssen [21] and provided in Section 4.2.2. These constant values were used directly in the CFD-DEM simulations of Li et al. [24] which coincides with an assumption that “there is no CO₂ supply limitation due to fast mixing in the relatively small bed and constant CO₂ partial pressure on the particles’ surface.” While this may be a valid assumption, it presents a difficulty when fluidizing the bed with pure N₂ (discussed below), i.e., once q_e is set, the reaction proceeds whether or not there is any CO₂ in the bed or not. Therefore, we use a very simple fit of the data of Section 4.2.2 for the equilibrium adsorption in the reaction model: $q_e = 0.17 Y_{\text{CO}_2}^{0.3}$. It should be strongly stressed that this fit is not general and does not have appropriate limiting behavior with increasing CO₂ partial pressure. However, it is accurate within the small window of applicability of this experiment and does have the correct limiting behavior we are interested in, namely

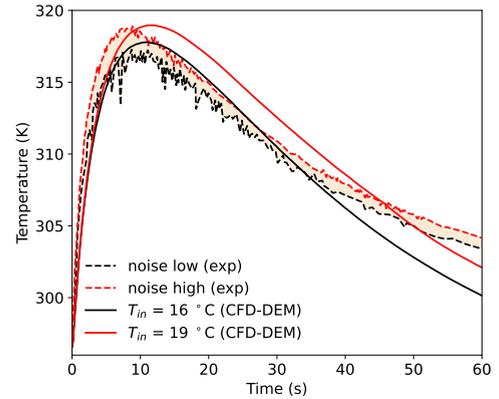
$$\lim_{Y_{\text{CO}_2} \rightarrow 0} q_e = 0.$$

The bed wall temperature is set to a constant $T_w = 300$ K. The initial fluid and particle temperature is set to $T_{g,0} = T_{p,0} = 296.4$ K [24]. The inlet velocity is $u_{g,\text{in}} = 1.2$ m/s uniform over inflow plane. The particles are initialized in a uniform eight particles per cell lattice spanning the width and depth of the bed and an initial static bed height of $h_0 = 92$ mm. This packed, uniform configuration is in stark contrast to the experiment which was fluidized by pure N₂ and then switched to a N₂-CO₂ mixture. To achieve this fluidized initial state, the bed is fluidized from the

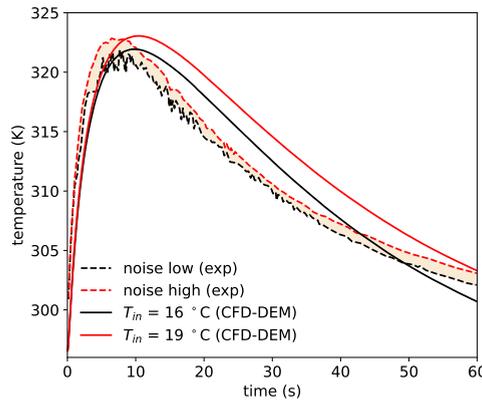
true initial state for a period of $t_0 = 5$ s. Then, a rapid transient from $t = t_0$ to $t = t_0 + 0.1$ occurs in which a linear ramp up in Y_{CO_2} and down in Y_{N_2} occurs. To avoid changing the initial (i.e., at time t_0) bed temperature, the inlet gas temperature is $T_{g,in} = T_{g,0}$ during the startup period. A similar 0.1 s sharp linear ramp in temperature is also triggered at $t = t_0$ in which $T_{g,in}$ shifts from the initial particle temperature to the desired inlet temperature. We note that there is some uncertainty in the value of $T_{g,in}(t > t_0)$ which has a non-negligible influence on the quantity of interest. In their own CFD-DEM simulations, Li et al. [24] use a value of $T_{g,in} = 289.15$ K, however the experimental description gives the range of this value as 16 °C to 19 °C. We capture the variation due to this uncertain parameter by running a pair of simulations for each condition, one using $T_{g,in} = 289.15$ K and the other with $T_{g,in} = 292.15$ K.



(a) CO₂ molar fraction equal to 20%.



(b) CO₂ molar fraction equal to 40%.



(c) CO₂ molar fraction equal to 60%.

Figure 6: Experimental (exp) and numerical (CFD-DEM) results of the CO₂-capturing fluidized bed tests. The computed results are compared against the Li et al. [24] dataset for three different values of CO₂ mole fractions: (a) 20%, (b) 40%, and (c) 60%.

As with the cooling bed, the bed averaged particle temperature is the primary quantity of interest in the adsorbing bed. The particle temperature is again measured experimentally with IR imaging. There is significantly more noise in the time series data and each condition was repeated three times [24, 21], further increasing the spread of the data. In an effort to capture the spread of the experimental data, a pair of curves were digitized for each figure representing an envelope capturing the upper and lower bounds, irrespective of the three ensembles. The simulation results are also plotted as a pair of curves, owing to the upper and lower inlet gas temperatures prescribed. Therefore while the experimental and computational results appear similar qualitatively, it is important to note that the former represents an envelope of aleatory uncertainty while the latter gives the system response to a single (although important) epistemic input uncertainty. The MFIX-Exa simulations are compared to the experimental data in Fig. 6. As might well be

expected due to the increased physical complexity of the problem, the agreement is noticeably worse than the non-reacting case. However, the predictions are still quite good. The model is able to capture the steep rise in temperature as the particles heat up by capturing CO₂ from the fluidizing gas. The maximum temperature is also predicted well, as is the time that it is reached, corresponding to the saturation of the zeolite particles with adsorbed CO₂. Where we see the most disagreement is in the cooling phase of the simulation. Because the cooling in the adsorbing bed appears to be predicted slightly worse than the cooling-only bed (non-reacting), this points to an approximation of the kinetic model of adsorption. Specifically, the equilibrium adsorption is, in general, a decreasing function of temperature. Indeed, a standard method to regenerate saturated CO₂ adsorbers is temperature swing cycle whereby the zeolite particles are introduced to a high temperature environment to release the CO₂ before repeating the process. By neglecting this secondary reaction mechanism, the model is unable to capture the reduced capacity at the elevated temperature which is recovered during the cooling phase.

4.3. Performance

In our previous work [28], a weak scaling analysis was performed using an idealized, geometry-free, infinitely extensible problem setup. The system was periodic in two directions with mass inflow and pressure outflow boundary conditions in the third dimension. The “base” problem was decomposed into $2 \times 2 \times 2$ grids. Each grid contained 64^3 fluid cells having an edge 0.5 mm long. Each cell contained eight particles of diameter $d_p = 2 \times 10^{-4}$ m and density $\rho_p = 4250.0 \text{ kg m}^{-3}$. The problem was weakly scaled by a factor of 2 first applied along the flow dimension, then along each of the periodic dimensions, and finally repeated until the desired scale was reached. The same problem is considered here, first by adding new physics, one at a time, to the original incompressible fluid system and then repeating the scaling analysis.

Problem type	Physics
0	base problem
1	add solving for fluid density
2	set ideal gas constraint
3	add solving for enthalpy
4	add solving for species
5	add “null” reactions
6	set “hot-null” reactions

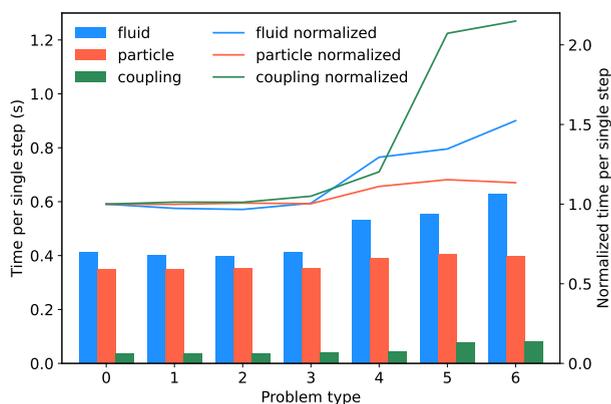
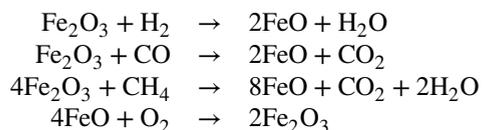


Figure 7: Timings for fluid update, particles update, and interphase coupling for different problem types with increasing computational costs.

To progressively include the different parts of the physical model, we consider the moderate-sized four-cubed system, i.e., four grids in each dimension. The problem is evolved for ten fluid steps, and the particle spring constant is set such that there are twenty DEM subcycles per fluid step. All simulations are carried out on the OLCF Summit supercomputer. Each case is repeated three times, and the lowest total time per step is recorded. First, the original case is re-run to provide a new baseline (“problem type 0”). The fluid is incompressible and uniform in density, with $\rho_g = 0.256695 \text{ kg m}^{-3}$. Then, keeping the same problem definition, density advection is enabled (“problem type 1”). In this case, density remains uniform and constant; however, the simulation incurs the same computational expense as if the incompressible fluid had a varying density field. Next, the incompressible constraint is dropped for the open system ideal gas equation of state (“problem type 2”). For simplicity, the ideal gas is given the molecular weight of pure nitrogen. The cold-flow simulation becomes hot in the third step: fluid and particles are set to a temperature of 1100 K, and enthalpy becomes an advected scalar (“problem type 3”). Next, both phases are decomposed into constituent species (“problem type 4”). Seven fluid species, H₂, O₂, N₂, H₂O, CO, CO₂ and CH₄, and three solid species, Fe₂O₃, FeO, and inert, are considered, each with a pair of seven-coefficient NASA polynomials [8] for specific heat capacity

evaluation. Finally, four heterogeneous chemical reactions are considered



representing a reduction of hematite by hydrogen, carbon monoxide, and methane, and the oxidation of wustite in a lumped model of an ilmenite oxygen carrier. As an intermediate step, the reactions are included, but each is given a zero rate (“problem type 5”). Finally, the kinetic rates of [1] are included (“problem type 6”) for the reduction and oxidation reactions. Although encapsulated in a trivial problem setup, this final configuration includes most the physics needed to represent a CO₂ capturing CLR.

The six levels of additional physics above the incompressible baseline are listed in Fig. 7, along with the corresponding timing results, which show the average cost per fluid step for the fluid advance, the particle subcycling, and the interphase coupling. All three components increase the computational cost as increasingly complex physical models are added to the system. In particular, there is a visible increase in the fluid phase update cost when species mass fractions and reactions with algebraic rates are added to the system. The additional computational cost is due to the significant increase in the number of operations for the multi-component fluid. Additionally, there is a considerable jump in the coupling cost when heterogeneous reactions are included, even with null rates. Although the 2x jump for interphase coupling cost is significant, it remains a small contribution to the total time. We believe the approximately 34% increase in overall time to solution moving from an incompressible system to a hot, reacting system is remarkable.

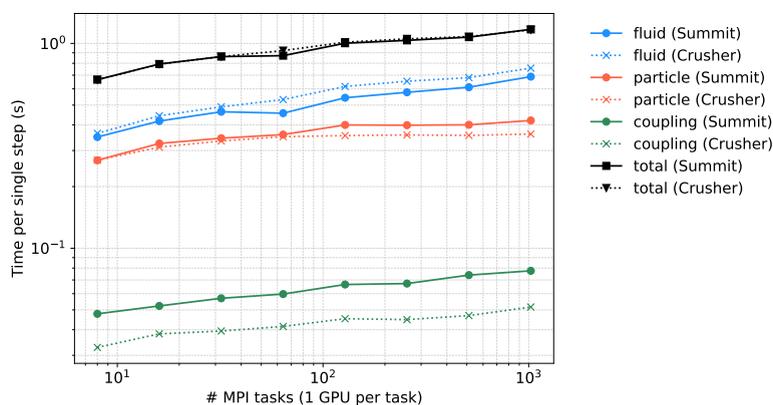


Figure 8: Weak scaling analysis for the fluid update, particle update, interphase coupling, and the total updating step

Using the complete physics system, i.e., “problem type 6,” we carried out a weak scaling study from 8 to 1024 grids. Each grid is mapped to one compute rank. On Summit, each rank is a CPU-GPU pair running on an Intel Power9 and Nvidia P100 GPU architecture. The study was also carried out on the Crusher Test and Development System. Though smaller in scale, Crusher contains the same next-generation architecture that will soon be available on Frontier, slated to be the US DOE’s first exascale capable machine. Each rank is mapped to a CPU-GCD pair on Crusher using AMD EPYC 7A53 CPUs and AMD MI250X, where each MI250x contains two Graphics Compute Dies (GCDs). The average time per step observed on both machines is provided in Fig. 8. The results show that MFIX-Exa maintains good performance and scalability on current and emerging leadership-class facilities.

5. Conclusions

This work describes the latest advances in MFIX-Exa for numerical simulation of chemically reacting multiphase flows. MFIX-Exa’s new features include thermodynamics and chemistry modeling through the addition of conservation equations for chemical species and energy and the inclusion of chemical reactions. Another improvement we recently made in MFIX-Exa is the development of an EB-aware Godunov approach for the time integration of the fluid phase. The Godunov algorithm preserves the second-order accuracy in time while reducing the computational cost. Also,

we have added the possibility of using the ideal gas equation of state for the fluid phase in either open or closed domains. The model results were benchmarked against well-known problems such as wet-bulb tests and packed and fluidized beds. In particular, we ran three verification and two validation tests. The numerical simulations agreed with the experimental data and analytical solutions, consistently obtaining satisfactory results. Finally the performance of the new code was benchmarked by progressively adding physical models finding an acceptable increase in time to solution between cold and fully reacting flows. A weak scaling analysis showed that that code performs well on the current leadership-class computer (Summit) and will continue to perform well on the next generation of leadership class computers (Frontier). As we transition to a clean energy economy, mitigating the impact of fossil fuels on climate is critical and technologies based on chemical looping reactors (CLRs) offer a promising strategies to reduce CO₂ emissions. We have developed MFix-Exa to support to the research and development of such technologies by enabling simulations of industrial-scale CLRs on exascale supercomputers.

CRediT authorship contribution statement

Roberto Porcu: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Visualization. **Jordan Musser:** Conceptualization, Methodology, Software, Formal analysis, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Ann S. Almgren:** Methodology, Software, Formal analysis, Writing - Original Draft, Writing - Review & Editing, Supervision. **John B. Bell:** Methodology, Formal analysis, Writing - Review & Editing, Supervision. **William D. Fullmer:** Validation, Formal analysis, Writing - Review & Editing, Visualization. **Deepak Rangarajan:** Methodology, Software, Writing - Original Draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Disclaimer

This project was funded by the U.S. Department of Energy, National Energy Technology Laboratory, in part, through a site support contract. Neither the United States Government nor any agency thereof, nor any of their employees, nor the support contractor, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Appendices

Appendix A Ideal gas constraint

We want to constrain the evolution of the thermodynamic equations so that the solution is consistent with the multicomponent ideal gas equation of state,

$$p_{\text{therm}} = \rho_g RT_g \sum_{k=1}^{N_g} \frac{X_{gk}}{W_{gk}}. \quad (24)$$

Low Mach number asymptotics [25] show that p_{therm} is only a function of time; it is constant in space. Rather than trying to numerically enforce the constraint directly, we will derive a constraint on the velocity field that forces the

evolution of the thermodynamic variables to satisfy Eq. (8). To derive this constraint we first take the Lagrangian derivative of the ideal gas equation of state (8) to obtain

$$\frac{Dp_{\text{therm}}}{Dt} = \frac{p_{\text{therm}}}{\rho_g} \frac{D\rho_g}{Dt} + \frac{p_{\text{therm}}}{T_g} \frac{DT_g}{Dt} + p_{\text{therm}} W_g \sum_{k=1}^{N_g} \frac{1}{W_{gk}} \frac{DX_{gk}}{Dt}. \quad (25)$$

Differentiating the definition of the mixture enthalpy Eq. (6) gives

$$c_{p,g} \frac{DT_g}{Dt} = \frac{Dh_g}{Dt} - \sum_{k=1}^{N_g} h_{gk} \frac{DX_{gk}}{Dt}.$$

Substituting this result into Eq. (25), multiplying by ε_g , and dividing by p_{therm} , we obtain

$$\frac{\varepsilon_g}{p_{\text{therm}}} \frac{Dp_{\text{therm}}}{Dt} = \frac{\varepsilon_g}{\rho_g} \frac{D\rho_g}{Dt} + \frac{\varepsilon_g}{c_{p,g} T_g} \left(\frac{Dh_g}{Dt} - \sum_{k=1}^{N_g} h_{gk} \frac{DX_{gk}}{Dt} \right) + \varepsilon_g W_g \sum_{k=1}^{N_g} \frac{1}{W_{gk}} \frac{DX_{gk}}{Dt}. \quad (26)$$

From the conservation equations (2), (4), (5) we can show that

$$\begin{aligned} \frac{\varepsilon_g}{\rho_g} \frac{D\rho_g}{Dt} &= \frac{G_{p \rightarrow g}^\rho}{\rho_g} - \left(\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) \right), \\ \frac{DX_{gk}}{Dt} &= \frac{1}{\rho_g} \left[S_k^\rho - X_{gk} \left(\frac{\partial (\varepsilon_g \rho_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u}_g) \right) \right] = \frac{S_k^\rho}{\rho_g} - \frac{X_{gk} S^\rho}{\rho_g}, \\ \frac{Dh_g}{Dt} &= \frac{1}{\rho_g} \left[\varepsilon_g \frac{Dp_{\text{therm}}}{Dt} + S^h - h_g \left(\frac{\partial (\varepsilon_g \rho_g)}{\partial t} + \nabla \cdot (\varepsilon_g \rho_g \mathbf{u}_g) \right) \right] = \frac{\varepsilon_g}{\rho_g} \frac{Dp_{\text{therm}}}{Dt} + \frac{S^h}{\rho_g} - \frac{h_g S^\rho}{\rho_g}. \end{aligned}$$

Substituting these results into Eq. (26) we get

$$\left(\frac{\varepsilon_g}{p_{\text{therm}}} - \frac{\varepsilon_g}{c_{p,g} \rho_g T_g} \right) \frac{Dp_{\text{therm}}}{Dt} + \left(\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) \right) = \frac{S^h}{\rho_g c_{p,g} T_g} + \sum_{k=1}^{N_g} \left(\frac{W_g}{W_{gk}} - \frac{h_{gk}}{c_{p,g} T_g} \right) \frac{S_k^\rho}{\rho_g} \equiv S,$$

where we have used the identities

$$W_g \sum_{k=1}^{N_g} \frac{X_{gk}}{W_{gk}} S^\rho = S^\rho, \quad \text{and} \quad \sum_{k=1}^{N_g} h_{gk} X_{gk} S^\rho = h_g S^\rho,$$

to simplify the right hand side.

For an open system case, $p_{\text{therm}} = p_{\text{amb}}$, where p_{amb} is a constant ambient pressure. Then, the constraint becomes:

$$\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) = S,$$

In this case the net flow into and out of the domain balances expansion and contraction of the fluid so that the pressure remains constant.

For a closed system, the fluid is confined, so there is no net flow through the domain boundary. Thermodynamic processes must be balanced by a change in the (spatially constant) thermodynamic pressure. To determine the thermodynamic pressure evolution we need to look at the constraint more carefully. The constraint has the form

$$\theta \frac{Dp_{\text{therm}}}{Dt} + \left(\frac{\partial \varepsilon_g}{\partial t} + \nabla \cdot (\varepsilon_g \mathbf{u}_g) \right) = S, \quad \text{where} \quad \theta = \varepsilon_g \left(\frac{1}{p_{\text{therm}}} - \frac{1}{c_{p,g} \rho_g T_g} \right).$$

where we have replaced the Lagrangian derivative to reflect that p_{therm} is only a function of time. We also observe that

$$\int_{\Omega} \left(\frac{\partial \epsilon_g}{\partial t} + \nabla \cdot (\epsilon_g \mathbf{u}_g) \right) dV = \int_{\partial\Omega} \epsilon_g \mathbf{u}_g \cdot \mathbf{n} dA = 0.$$

Here, the integral of the first term is zero reflecting the assumption that the particles volumes and hence the net fluid volume remain constant.

When there are thermodynamic processes, S will not necessarily integrate to zero so its effect cannot be captured by

$$\frac{\partial \epsilon_g}{\partial t} + \nabla \cdot (\epsilon_g \mathbf{u}_g) = S. \quad (27)$$

In fact, when S does not integrate to zero, there is no \mathbf{u}_g that satisfies Eq. (27). On the other hand, p_{therm} is only a function of t . This induces a decomposition of the dynamics. We break S and θ into their mean values (averages over the domain) and fluctuations (local values minus the mean) to obtain

$$\left(\bar{\theta} + \delta\theta \right) \frac{Dp_{\text{therm}}}{Dt} + \frac{\partial \epsilon_g}{\partial t} + \nabla \cdot (\epsilon_g \mathbf{u}_g) = \bar{S} + \delta S.$$

The mean component of the dynamics then defines the evolution of p_{therm} which, in the low Mach number formulation, is only a function of time and can be reduced to

$$\bar{\theta} \frac{dp_{\text{therm}}}{dt} = \bar{S}, \quad (28)$$

and the velocity constraint, as anticipated in Eq. (10), becomes

$$\frac{\partial \epsilon_g}{\partial t} + \nabla \cdot (\epsilon_g \mathbf{u}_g) = \delta S - \delta\theta \frac{\bar{S}}{\bar{\theta}}.$$

Since the right hand side of this equation integrates to zero, the constraint can be enforced.

Appendix B Godunov Methodology with Embedded Boundaries

B.1 Embedded Boundary Method for Complex Geometry

An embedded boundary formulation is used to represent the non-rectangular bounding geometry in the domain. The irregular boundary is defined by intersecting an analytically-specified boundary with a uniform Cartesian grid, with irregularly shaped cells, or ‘‘cut-cells’’, appearing only adjacent to the boundary. The EB information is precomputed and stored in a distributed database at the beginning of the calculation.

Following standard notation, we define each grid cell (i, j, k) to be either *covered*, *cut*, or *regular*. We define the geometric volume fraction, Λ , (not to be confused with the phasic volume fraction, ϵ_g) of each cell to be the fraction of that rectangular cell volume that is inside the fluid/particle region: *covered* cells have $\Lambda = 0$, *regular* cells have $\Lambda = 1$, and for cut cells $0 < \Lambda < 1$. Area fractions, a , are stored on each cell face, again with values in $[0, 1]$ representing the fraction of the face not covered. Finally, the location of the cell centroid (which for regular cells is identical to the cell center), and the locations of the face centroids are stored; these are scaled by the cell widths so have values in $[-0.5, 0.5]$ for each coordinate direction. Additionally, there is connectivity information between neighboring cells. For additional details on the embedded boundary implementation, we refer to the AMReX documentation https://amrex-codes.github.io/amrex/docs_html/.

The construction of the advective terms in the fluid phase update,

$$\begin{aligned} \mathbf{A}_{\rho}^{n+1/2} &= \nabla \cdot \left(\hat{\epsilon}_g \hat{\rho}_g^{n+1/2} \mathbf{u}_g^{n+1/2, \text{MAC}} \right), \\ \mathbf{A}_{Xk}^{n+1/2} &= \nabla \cdot \left(\hat{\epsilon}_g \hat{\rho}_g^{n+1/2} \hat{X}_{gk}^{n+1/2} \mathbf{u}_g^{n+1/2, \text{MAC}} \right), \\ \mathbf{A}_h^{n+1/2} &= \nabla \cdot \left(\hat{\epsilon}_g \hat{\rho}_g^{n+1/2} \hat{h}_g^{n+1/2} \mathbf{u}_g^{n+1/2, \text{MAC}} \right), \text{ and} \\ \mathbf{A}_{\mathbf{u}}^{n+1/2} &= \nabla \cdot \left(\hat{\epsilon}_g \hat{\rho}_g^{n+1/2} \mathbf{u}_g^{n+1/2, \text{MAC}} \otimes \mathbf{u}_g^{n+1/2} \right), \end{aligned}$$

follows a Godunov approach wherein quantities are predicted from cell centroids at time t^n to face centroids at $t^{n+1/2}$ and fluxes are constructed using these states and an intermediate velocity field, $\mathbf{u}_g^{n+1/2,MAC}$. This differs from the Method-of-Lines approach described in [28] and is described in detail below. We note that while the MOL approach required a CFL of less than 0.5, the Godunov approach allows a CFL up to 1.0.

The Godunov methodology for incompressible Navier-Stokes flow in the absence of cut cells is well-established; see, e.g. [2]. Here we focus on the differences that result from the presence of the embedded boundaries and the fact that particles displace the fluid resulting in non-constant fluid volume fraction.

B.2 Creating $\mathbf{u}_g^{n+1/2,MAC}$

We start by extrapolating the normal velocities from cell centroids to the centroids of cell faces using a second-order Taylor series expansion in space and time. For each face with a non-zero area fraction, we extrapolate the normal velocity component from the centroids of the cells on either side to the face centroid, creating left (L) and right (R) states. Using an x -face as an example, for face $(i + 1/2, j, k)$ we define

$$\hat{u}_{i+1/2,j,k}^{L,n+1/2} = \hat{u}_{i+1/2,j,k}^L + \frac{dt}{2} (-(\widehat{vu}_y)_{i,j,k} - (\widehat{wu}_z)_{i,j,k} + f_{x,i,j,k}^n), \quad (29)$$

extrapolated from (i, j, k) , where

$$\hat{u}_{i+1/2,j,k}^L = u_{i,j,k}^n + \left(\delta_x - \frac{dt}{2} u_{i,j,k}^n \right) u_{i,j,k}^x + \delta_y u_{i,j,k}^y + \delta_z u_{i,j,k}^z, \quad (30)$$

and

$$\hat{u}_{i+1/2,j,k}^{R,n+1/2} = \hat{u}_{i+1/2,j,k}^R + \frac{dt}{2} (-(\widehat{vu}_y)_{i+1,j,k} - (\widehat{wu}_z)_{i+1,j,k} + f_{x,i+1,j,k}^n), \quad (31)$$

extrapolated from $(i + 1, j, k)$, where

$$\hat{u}_{i+1/2,j,k}^R = u_{i+1,j,k}^n + \left(\delta_x - \frac{dt}{2} u_{i+1,j,k}^n \right) u_{i+1,j,k}^x + \delta_y u_{i+1,j,k}^y + \delta_z u_{i+1,j,k}^z. \quad (32)$$

Here the slopes (u^x, u^y, u^z) are calculated using a least-squares fit to available data and δ_x, δ_y and δ_z are the components of the distance vector from the cell centroid to the face centroid of the x -face at $(i - 1/2, j, k)$. These slopes are limited with a Barth-Jespersen type of limiter [4] that enforces no new maxima or minima when the state is predicted to the face centroids. If sufficient data is available for cells with unit volume fraction, this computation instead uses a standard second- or fourth-order slope calculation with limiting as described in [11]. f is the sum of external forces, discussed later.

The transverse derivative terms $(\widehat{vu}_y$ and \widehat{wu}_z in this case) are evaluated by first extrapolating all velocity components to the face centroids of the transverse faces from the cell centers on either side, then choosing between these states using the upwinding procedure defined below. In particular, in the y direction we define $\hat{U}_{i,j+1/2,k}^F$ and $\hat{U}_{i,j+1/2,k}^T$ analogously to how we defined $\hat{u}_{i+1/2,j,k}^R$ and $\hat{u}_{i+1/2,j,k}^L$, but here on the y -faces and including all three velocity components. Values are similarly traced from (i, j, k) and $(i, j, k + 1)$ to the $(i, j, k + 1/2)$ faces to define $\hat{U}_{i,j,k+1/2}^D$ and $\hat{U}_{i,j,k+1/2}^U$, respectively.

In this upwinding procedure we first define a normal advective velocity on the face (suppressing the $(i, j + 1/2, k)$ spatial indices on front and back states here and in the next equation):

$$\hat{v}_{i,j+1/2,k}^{adv} = \begin{cases} \hat{v}^F, & \text{if } \hat{v}^F > 0, \hat{v}^F + \hat{v}^B > 0, \\ 0, & \text{if } \hat{v}^F \leq 0, \hat{v}^B \geq 0 \text{ or } \hat{v}^F + \hat{v}^B = 0, \\ \hat{v}^B, & \text{if } \hat{v}^B < 0, \hat{v}^F + \hat{v}^B < 0. \end{cases}$$

We now upwind \hat{U} based on $\hat{v}_{i,j+1/2,k}^{adv}$:

$$\hat{U}_{i,j+1/2,k} = \begin{cases} \hat{U}^F, & \text{if } \hat{v}_{i,j+1/2,k}^{adv} > 0, \\ 1/2(\hat{U}^F + \hat{U}^B), & \text{if } \hat{v}_{i,j+1/2,k}^{adv} = 0, \\ \hat{U}^B, & \text{if } \hat{v}_{i,j+1/2,k}^{adv} < 0. \end{cases}$$

After constructing $\widehat{U}_{i,j-\frac{1}{2},k}$, $\widehat{U}_{i,j,k+\frac{1}{2}}$ and $\widehat{U}_{i,j,k-\frac{1}{2}}$ in a similar manner, we use these upwind values to form the transverse derivatives in Eqs. (29) and (31):

$$(\widehat{u}_y)_{i,j,k} = \frac{1}{2dy} (\widehat{v}_{i,j+\frac{1}{2},k}^{adv} + \widehat{v}_{i,j-\frac{1}{2},k}^{adv}) (\widehat{u}_{i,j+\frac{1}{2},k} - \widehat{u}_{i,j-\frac{1}{2},k}), \quad (33)$$

$$(\widehat{u}_z)_{i,j,k} = \frac{1}{2dz} (\widehat{w}_{i,j,k+\frac{1}{2}}^{adv} + \widehat{w}_{i,j,k-\frac{1}{2}}^{adv}) (\widehat{u}_{i,j,k+\frac{1}{2}} - \widehat{u}_{i,j,k-\frac{1}{2}}). \quad (34)$$

Here $\widehat{v}_{i,j+\frac{1}{2},k}^{adv}$ is the value of \widehat{v}^{adv} bilinearly interpolated to the center of the face at $(i, j + \frac{1}{2}, k)$ from the four nearest y-face centroids; $\widehat{u}_{i,j,k+\frac{1}{2}}$ is the value of \widehat{u} bilinearly interpolated from the same four nearest y-face centroids.

An important note here is that if any of the four faces that contribute to the transverse derivatives for a particular cell have zero area, all of the transverse *and* forcing terms are identically set to 0. For example, when constructing $\widehat{u}_{i+\frac{1}{2},j,k}^{L,n+\frac{1}{2}}$, if any of the areas $a_{i,j+\frac{1}{2},k}$, $a_{i,j-\frac{1}{2},k}$, $a_{i,j,k-\frac{1}{2}}$ or $a_{i,j,k+\frac{1}{2}}$ are zero, then we simply define

$$\widehat{u}_{i+\frac{1}{2},j,k}^{L,n+\frac{1}{2}} = \widehat{u}_{i+\frac{1}{2},j,k}^L. \quad (35)$$

The normal velocity at each face is then determined by an upwinding procedure based on the states predicted from the cell centers on either side. The procedure is similar to that described above, i.e., (suppressing the $(i + \frac{1}{2}, j, k)$ indices). We note that the upwinding procedure on faces with area fractions less than 1 is identical to that on faces with area fractions equal to 1. We do not compute values on covered faces (i.e., faces with zero area fraction).

$$\widehat{u}_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} = \begin{cases} \tilde{u}^{L,n+\frac{1}{2}}, & \text{if } \tilde{u}^{L,n+\frac{1}{2}} > 0 \text{ and } \tilde{u}^{L,n+\frac{1}{2}} + \tilde{u}^{R,n+\frac{1}{2}} > 0, \\ 0, & \text{if } \tilde{u}^{L,n+\frac{1}{2}} \leq 0, \tilde{u}^{R,n+\frac{1}{2}} \geq 0 \text{ or } \tilde{u}^{L,n+\frac{1}{2}} + \tilde{u}^{R,n+\frac{1}{2}} = 0, \\ \tilde{u}^{R,n+\frac{1}{2}}, & \text{if } \tilde{u}^{R,n+\frac{1}{2}} < 0 \text{ and } \tilde{u}^{L,n+\frac{1}{2}} + \tilde{u}^{R,n+\frac{1}{2}} < 0. \end{cases} \quad (36)$$

We follow a similar procedure to construct $\widehat{v}_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}}$ and $\widehat{w}_{i,j,k+\frac{1}{2}}^{n+\frac{1}{2}}$. We refer to this unique value of normal velocity on each face as $U^{MAC,*}$.

B.3 MAC Projection

To enforce the divergence constraint, we apply a discrete projection by solving the following elliptic equation

$$D^{MAC} \left(\frac{\epsilon_g^n}{\rho_g^{n+\frac{1}{2}}} \mathbf{G}^{MAC} \phi^{MAC} \right) = D^{MAC} \left(\frac{2\epsilon_g^{n+\frac{1}{2}}}{dt} U^{MAC,*} \right), \quad (37)$$

for ϕ^{MAC} . Here, D^{MAC} represents the divergence at cell centroids of area-weighted velocities defined at face centroids and \mathbf{G}^{MAC} represents the gradient at face centroids of data at cell centers. The density $\rho_g^{n+\frac{1}{2}}$ and volume fraction $\epsilon_g^{n+\frac{1}{2}}$ are interpolated separately from cell centroids to face centroids. The solution, ϕ^{MAC} , which is defined at cell centers, is then used to compute

$$U^{MAC} = U^{MAC,*} - \frac{dt}{2\rho_g^{n+\frac{1}{2}}} \mathbf{G}^{MAC} \phi^{MAC}. \quad (38)$$

This projection is done using AMReX EB-aware `MacProjector` class, whose method `MacProjector::project`, given a vector field V and a specified RHS , returns a vector field that satisfies $\nabla \cdot V = RHS$.

B.3.1 Velocity forces

The force used for extrapolation is defined by,

$$f_{x,i,j,k}^n = \frac{1}{\rho_g^{n+\frac{1}{2}}} \left(-(G_x p_g)_{i,j,k}^{n-\frac{1}{2}} + \mu \Delta^h u_{i,j,k}^n + g_x + \frac{\epsilon_g^n}{\beta^n} u_{i,j,k}^n \right). \quad (39)$$

Here, Δ^h is a standard seven-point cell-centered approximation to the Laplacian and $G = (G_x, G_y, G_z)$ is a discretization of the gradient operator which defines a cell-centered gradient from a node-based pressure field.

B.4 Creating $s^{n+1/2}$

Let the scalar s represent any advected quantities as well as all three velocity components. We now extrapolate s from cell centroids to face centroids as described in Sec. B.2. For example, on face $(i + 1/2, j, k)$ we define

$$\tilde{s}_{i+1/2,j,k}^{L,n+1/2} = \hat{s}_{i+1/2,j,k}^L + \frac{dt}{2} (-(\widehat{vS}_y)_{i,j,k} - (\widehat{wS}_z)_{i,j,k} + f_{x,i,j,k}^n), \quad (40)$$

extrapolated from (i, j, k) , where

$$\hat{s}_{i+1/2,j,k}^L = s_{i,j,k}^n + \left(\delta_x - \frac{dt}{2} u_{i,j,k}^n \right) s_{i,j,k}^x + \delta_y s_{i,j,k}^y + \delta_z s_{i,j,k}^z, \quad (41)$$

and

$$\tilde{s}_{i+1/2,j,k}^{R,n+1/2} = \hat{s}_{i+1/2,j,k}^R + \frac{dt}{2} (-(\widehat{vS}_y)_{i+1,j,k} - (\widehat{wS}_z)_{i+1,j,k} + f_{x,i+1,j,k}^n), \quad (42)$$

extrapolated from $(i + 1, j, k)$, where

$$\hat{u}_{i+1/2,j,k}^R = u_{i+1,j,k}^n + \left(\delta_x - \frac{dt}{2} u_{i,j,k}^n \right) s_{i+1,j,k}^x + \delta_y s_{i+1,j,k}^y + \delta_z s_{i+1,j,k}^z. \quad (43)$$

Here again the slopes (s^x, s^y, s^z) are calculated using a least-squares fit to available data and δ_x, δ_y and δ_z are the components of the distance vector from the cell centroid to the face centroid of the x -face at $(i - 1/2, j, k)$. The transverse terms are computed in the same fashion described previously. Now, however, we use the component of \mathbf{U}^{MAC} normal to the face in question in upwinding process.

We note again that if any of the four faces that contribute to the transverse derivatives for a particular cell have zero area, all of the transverse *and* forcing terms are identically set to 0. For example, when constructing $\tilde{s}_{i+1/2,j,k}^{L,n+1/2}$, if any of the areas $a_{i,j+1/2,k}, a_{i,j-1/2,k}, a_{i,j,k-1/2}$ or $a_{i,j,k+1/2}$ are zero, then we simply define

$$\tilde{s}_{i+1/2,j,k}^{L,n+1/2} = \hat{s}_{i+1/2,j,k}^L. \quad (44)$$

We upwind $\tilde{s}_{i+1/2,j,k}^{L,n+1/2}$ and $\tilde{s}_{i+1/2,j,k}^{R,n+1/2}$ using the normal component of \mathbf{U}^{MAC} to define $s_{i+1/2,j,k}^{n+1/2}$. Again, suppressing the subscripts, we define

$$s^{n+1/2} = \begin{cases} \tilde{s}^{L,n+1/2}, & \text{if } u^{MAC} > 0, \\ \frac{1}{2}(\tilde{s}^{L,n+1/2} + \tilde{s}^{R,n+1/2}), & \text{if } u^{MAC} = 0, \\ \tilde{s}^{R,n+1/2}, & \text{if } u^{MAC} < 0. \end{cases} \quad (45)$$

B.4.1 Computing the fluxes

We define the density update as

$$\begin{aligned} A_\rho^{n+1/2} = & \left((a_{i+1/2,j,k} u_{i+1/2,j,k}^{MAC} \hat{\epsilon}_{g,i+1/2,j,k} s_{i+1/2,j,k}^{n+1/2} - a_{i-1/2,j,k} u_{i-1/2,j,k}^{MAC} \hat{\epsilon}_{g,i-1/2,j,k} s_{i-1/2,j,k}^{n+1/2}) \Delta_y \Delta_z \right. \\ & + (a_{i,j+1/2,k} v_{i,j+1/2,k}^{MAC} \hat{\epsilon}_{g,i,j+1/2,k} s_{i,j+1/2,k}^{n+1/2} - a_{i,j-1/2,k} v_{i,j-1/2,k}^{MAC} \hat{\epsilon}_{g,i,j-1/2,k} s_{i,j-1/2,k}^{n+1/2}) \Delta_x \Delta_z \\ & \left. + (a_{i,j,k+1/2} w_{i,j,k+1/2}^{MAC} \hat{\epsilon}_{g,i,j,k+1/2} s_{i,j,k+1/2}^{n+1/2} - a_{i,j,k-1/2} w_{i,j,k-1/2}^{MAC} \hat{\epsilon}_{g,i,j,k-1/2} s_{i,j,k-1/2}^{n+1/2}) \Delta_x \Delta_y \right) \\ & / (\Lambda_{i,j,k} \Delta_x \Delta_y \Delta_z), \end{aligned} \quad (46)$$

with analogous constructions for the other scalar updates, $A_{Xk}^{n+1/2}$ and $A_h^{n+1/2}$. Similarly, we define the velocity update as

$$\begin{aligned} \mathbf{A}_u^{n+1/2} = & \left((a_{i+1/2,j,k} u_{i+1/2,j,k}^{MAC} \hat{\epsilon}_{g,i+1/2,j,k} \mathbf{u}_{g,i+1/2,j,k}^{n+1/2} - a_{i-1/2,j,k} u_{i-1/2,j,k}^{MAC} \hat{\epsilon}_{g,i-1/2,j,k} \mathbf{u}_{g,i-1/2,j,k}^{n+1/2}) \Delta_y \Delta_z \right. \\ & + (a_{i,j+1/2,k} v_{i,j+1/2,k}^{MAC} \hat{\epsilon}_{g,i,j+1/2,k} \mathbf{u}_{g,i,j+1/2,k}^{n+1/2} - a_{i,j-1/2,k} v_{i,j-1/2,k}^{MAC} \hat{\epsilon}_{g,i,j-1/2,k} \mathbf{u}_{g,i,j-1/2,k}^{n+1/2}) \Delta_x \Delta_z \\ & \left. + (a_{i,j,k+1/2} w_{i,j,k+1/2}^{MAC} \hat{\epsilon}_{g,i,j,k+1/2} \mathbf{u}_{g,i,j,k+1/2}^{n+1/2} - a_{i,j,k-1/2} w_{i,j,k-1/2}^{MAC} \hat{\epsilon}_{g,i,j,k-1/2} \mathbf{u}_{g,i,j,k-1/2}^{n+1/2}) \Delta_x \Delta_y \right) \\ & / (\Lambda_{i,j,k} \Delta_x \Delta_y \Delta_z). \end{aligned} \quad (47)$$

In all cases $\hat{\varepsilon}_g$ is the value of ε_g interpolated from cell centroids to face centroids. Δ_x , Δ_y and Δ_z are the cell sizes in the three directions.

As noted in [26], a “redistribution” operation is required to address the classic “small cell problem.” Specifically, without redistribution the advective CFL constraint would be dictated by the volume of the smallest cells which would otherwise have a minimal contribution to the global solution. In our previous work, we used a “flux redistribution.” Here, we now use the recently developed weighted state redistribution which is described in detail in [18].

References

- [1] Abad, A., Adanez, J., Cuadrat, A., Garcia-Labiano, F., Gayan, P., Luis, F., 2011. Kinetics of redox reactions of ilmenite for chemical-looping combustion. *Chemical Engineering Science* 66, 689–702.
- [2] Almgren, A.S., Bell, J.B., Colella, P., Howell, L.H., Welcome, M.L., 1998. A conservative adaptive projection method for the variable density incompressible navier–stokes equations. *Journal of Computational Physics* 142, 1–46.
- [3] Anderson, T.B., Jackson, R., 1967. A fluid mechanical description of fluidized beds. *Industrial & Engineering Chemistry Fundamentals* 6, 527–534.
- [4] Barth, T., Jespersen, D., 1989. The design and application of upwind schemes on unstructured meshes, in: 27th Aerospace sciences meeting, Reno, NV USA. doi:10.2172/1630414.
- [5] Beetstra, R., van der Hoef, M.A., Kuipers, J.A.M., 2007. Drag force of intermediate Reynolds number flow past mono- and bidisperse arrays of spheres. *AIChE Journal* 53, 489–501.
- [6] Bird, R.B., Stewart, W.E., Lightfoot, E.N., 2006. *Transport Phenomena* (Second ed.). John Wiley & Sons, New York.
- [7] Boyalakuntla, D.S., 2003. Simulation of Granular and Gas-Solid Flows using Discrete Element Model (DEM) Implemented in MFIX. Ph.D. thesis. Carnegie Mellon University. Pittsburgh, PA.
- [8] Burcat, A., Ruscic, B., et al., 2005. Third millenium ideal gas and condensed phase thermochemical database for combustion (with update from active thermochemical tables). Technical Report. Argonne National Lab.(ANL), Argonne, IL (United States).
- [9] Capecelatro, J., Desjardins, O., 2013. An euler-lagrange strategy for simulating particle-laden flows. *Journal of Computational Physics* 238, 1–31.
- [10] Clarke, M.A., Musser, J.M., 2020. The mfix particle-in-cell method (mfix-pic) theory guide URL: <https://www.osti.gov/biblio/1630426>, doi:10.2172/1630414.
- [11] Colella, P., 1985. A direct eulerian muscl scheme for gas dynamics. *SIAM Journal on Computing* 6, 104.
- [12] Dantas, T., Luna, F., Silva Jr, I., Torres, A., De Azevedo, D., Rodrigues, A., Moreira, R., 2011. Modeling of the fixed-bed adsorption of carbon dioxide and a carbon dioxide-nitrogen mixture on zeolite 13x. *Brazilian Journal of Chemical Engineering* 28, 533–544.
- [13] Deen, N., Annaland, M.V.S., Van der Hoef, M.A., Kuipers, J., 2007. Review of discrete particle modeling of fluidized beds. *Chemical Engineering Science* 62, 28–44.
- [14] Ding, J., Gidaspo, D., 1990. A bubbling fluidization model using kinetic theory of granular flow. *AIChE Journal* 36, 523–538.
- [15] Fullmer, W.D., Almgren, A.S., Rosso, M., Blaschke, J., Musser, J., 2019. Benchmarking of a preliminary mfix-exa code. arXiv preprint arXiv:1909.02067.
- [16] Fullmer, W.D., Hrenya, C.M., 2017. The clustering instability in rapid granular and gas-solid flows. *Annual Review of Fluid Mechanics* 49, 485–510.
- [17] Garg, R., Galvin, J., Li, T., Pannala, S., 2012. Open-source MFIX-DEM software for gas-solids flows: Part I-Verification studies. *Powder Technology* 220, 122–137.
- [18] Giuliani, A., Almgren, A., Bell, J., Berger, M., de Frahan, M.H., Rangarajan, D., 2022. Weighted state redistribution algorithms for embedded boundary methods. submitted for publication.
- [19] Gunn, D., 1978. Transfer of heat or mass to particles in fixed and fluidised beds. *International Journal of Heat and Mass Transfer* 21, 467–476.
- [20] van der Hoef, M., van Sint Annaland, M., Deen, N., Kuipers, J., 2008. Numerical simulation of dense gas-solid fluidized beds: A multiscale modeling strategy. *Annual Review of Fluid Mechanics* 40, 47–70.
- [21] Janssen, T.C.E., 2015. An investigation on heat transfer phenomena in fluidized bed reactors an experimental and modelling approach. Ph.D. thesis. Eindhoven University of Technology.
- [22] Kulic, E., 1976. An experimental and theoretical study of simultaneous heat and mass transfer applied to steam dousing. Ph.D. thesis. University of Waterloo. Waterloo, Ontario, Canada.
- [23] Lathouwers, D., Bellan, J., 2001. Modeling of dense gas–solid reactive mixtures applied to biomass pyrolysis in a fluidized bed. *International Journal of Multiphase Flow* 27, 2155–2187.
- [24] Li, Z., Janssen, T.C.E., Buist, K.A., Deen, N.G., van Sint Annaland, M., Kuipers, J.A.M., 2017. Experimental and simulation study of heat transfer in fluidized beds with heat production. *Chemical Engineering Journal* 317, 242–257.
- [25] Majda, A., Sethian, J.A., 1985. The derivation and numerical solution of the equations for zero Mach number combustion. *Combust. Sci. Technol.* 42, 185–205.
- [26] MFIX-Exa, 2020. MFIX-Exa Documentation. URL: <https://amrex-codes.github.io/MFIX-Exa/>.
- [27] Mills, A.F., 1999. Basic heat and mass transfer. Pearson College Division.
- [28] Musser, J., Almgren, A.S., Fullmer, W.D., Antepara, O., Bell, J.B., Blaschke, J., Gott, K., Myers, A., Porcu, R., Rangarajan, D., Rosso, M., Zhang, W., Syamlal, M., 2021. Mfix-exa: A path toward exascale cfd-dem simulations. *The International Journal of High Performance Computing Applications* doi:10.1177/10943420211009293.
- [29] Musser, J., Syamlal, M., Shahnam, M., Huckaby, D.E., 2015. Constitutive equation for heat transfer caused by mass transfer. *Chemical Engineering Science* 123, 436–443.

- [30] Nonaka, A., Day, M.S., Bell, J.B., 2018. A conservative, thermodynamically consistent numerical approach for low mach number combustion. part i: Single-level integration. *Combustion Theory and Modelling* 22, 156–184. doi:10.1080/13647830.2017.1390610.
- [31] Patil, A.V., Peters, E.A.J.F., Sutkar, V.S., Deen, N.G., Kuipers, J.A.M., 2015. A study of heat transfer in fluidized beds using an integrated dia/piv/ir technique. *Chemical Engineering Journal* 259, 90–106.
- [32] Ranz, W.E., Marshall, W.R., 1952. Evaporation from drops, part 1 & 2. *Chemical Engineering Progress* 48, 141–146; 173–180.
- [33] Reynolds, W.C., 1979. Thermodynamic properties in SI: graphs, tables, and computational equations for forty substances. Department of Mechanical Engineering, Stanford University.
- [34] Salehi, M.S., Askarishahi, M., Radl, S., 2017. Analytical solution for thermal transport in packed beds with volumetric heat source. *Chemical Engineering Journal* 316, 131–136.
- [35] Sauer, H.J., Howell, R.H., Coad, W.J., American Society Of Heating, R., Engineers, A.C., 2001. Principles of heating, ventilating, and air conditioning. American Society Of Heating, Refrigerating And Air-Conditioning Engineers.
- [36] Schumann, T.E.W., 1929. Heat transfer: a liquid flowing through a porous prism. *Journal of the Franklin Institute* 208, 405–416.
- [37] Tang, Y., Peters, E.A.J.F., Kuipers, J.A.M., Kriebitzsch, S.H.L., van der Hoef, M.A., 2015. A new drag correlation from fully resolved simulations of flow past monodisperse static arrays of spheres. *AIChE Journal* 61, 688–698.
- [38] Tenneti, S., Subramaniam, S., 2014. Particle-resolved direct numerical simulation for gas-solid flow model development. *Annual Review of Fluid Mechanics* 46, 199–230.
- [39] United States National Aeronautics and Space Administration, 2019. Overview of greenhouse gases. URL: <https://www.epa.gov/ghgemissions/overview-greenhouse-gases>.
- [40] Wen, C.Y., Yu, Y.H., 1966. Mechanics of fluidization. *Chemical Engineering Progress Symposium* 62, 100–111.
- [41] Zhang, W., Almgren, A., Beckner, V., Bell, J., Blaschke, J., Chan, C., Day, M., Friesen, B., Gott, K., Graves, D., Katz, M., Myers, A., Nguyen, T., Nonaka, A., Rosso, M., Williams, S., Zingale, M., 2019. AMReX: a framework for block-structured adaptive mesh refinement. *Journal of Open Source Software* 4, 1370.
- [42] Zhang, W., Myers, A., Gott, K., Almgren, A., Bell, J., to appear. Amrex: Block-structured adaptive mesh refinement for multiphysics applications. *IJHPCA* .