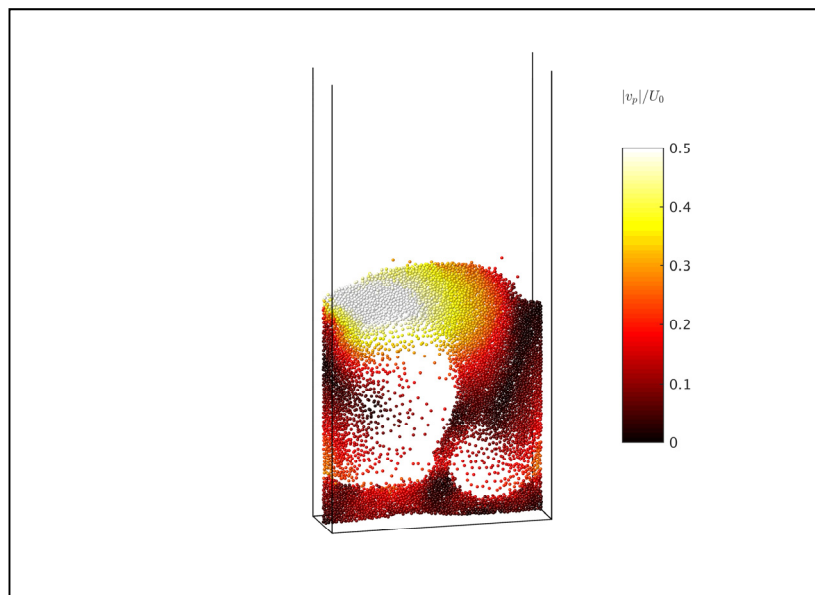




An Investigation into Solution Verification for CFD-DEM

X October 2017



U.S. DEPARTMENT OF
ENERGY



NATIONAL
ENERGY
TECHNOLOGY
LABORATORY

Office of Fossil Energy

NETL-TRS-X-2017

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, 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 therein 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 therein do not necessarily state or reflect those of the United States Government or any agency thereof.

This project was funded by the Department of Energy, National Energy Technology Laboratory, an agency of the United States Government, through a support contract with AECOM. Neither the United States Government nor any agency thereof, nor any of their employees, nor AECOM, nor any of their employees, makes any warranty, expressed 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.

Cover Illustration: Snapshot of particles frozen at an instant in an SSCP-I simulation for a fixed-bed solution verification study.

Suggested Citation: Fullmer, W. D.; Musser, J. *An Investigation into Solution Verification for CFD-DEM*; NETL-TRS-X-2017; NETL Technical Report Series; U.S. Department of Energy, National Energy Technology Laboratory: Morgantown, WV, 2017; p XX.

An electronic version of this report can be found at:

<http://netl.doe.gov/research/on-site-research/publications/featured-technical-reports>

An Investigation into Solution Verification for CFD-DEM

William D. Fullmer¹ and Jordan Musser²

**¹ U.S. Department of Energy, National Energy Technology Laboratory, AECOM, 3610
Collins Ferry Road, Morgantown, WV 26507**

**² U.S. Department of Energy, National Energy Technology Laboratory, 3610 Collins Ferry
Road, Morgantown, WV 26507**

NETL-TRS-X-2017

X October 2017

NETL Contacts:

William Rogers, Principal Investigator

Jonathan Lekse, Technical Portfolio Lead

David Alman, Executive Director, Acting, Research & Innovation Center

This page intentionally left blank.

Table of Contents

ABSTRACT	1
1. INTRODUCTION.....	2
2. METHODS	5
2.1 MODEL	5
2.2 CONVERGENCE.....	7
2.3 EXTRAPOLATION METHODS.....	9
3. CASE 1: BURSTING BUBBLE	11
3.1 SYSTEM AND CONDITIONS.....	11
3.2 GRID CONVERGENCE.....	12
3.3 BASELINE RESULTS.....	14
3.4 DISCRETIZATION SCHEMES	15
3.5 FILTER WIDTH.....	17
3.6 GARG_2012 INTERPOLATION	23
3.7 DIFFERENT PARTICLE CONFIGURATIONS.....	25
3.8 DISCRETIZATION SCHEMES ON PC2	28
3.9 DRAG LAWS ON PC3T.....	30
3.10 SUMMARY OF EXTRAPOLATED DATA	31
4. SSCPI	34
4.1 SYSTEM AND CONDITIONS.....	34
4.2 CONVERGENCE RESULTS	35
4.3 GIDASPOW BLEND DRAG LAW.....	38
4.4 SUMMARY OF EXTRAPOLATED DATA	40
4.5 NUMERICAL UNCERTAINTY	41
5. CONCLUSIONS	44
6. REFERENCES.....	46
APPENDIX.....	A-1

List of Figures

Figure 1: Sketch of two primary coupling schemes for Eulerian-Lagrangian numerical methods. The red circle indicates the particle scale, the black lines indicate the CFD grid and the blue scale indicate a filtered variable, e.g., V_p .	3
Figure 2: Convergence of the non-dimensionalized pressure temporal residuals for all grids. Horizontal lines show selected tolerances.	8
Figure 3: At left, comparison of the SRQoI for the bursting bubble case with (both) transient and residual tolerances set to 10^{-5} and 10^{-6} . At right, absolute difference between computed solutions.	8
Figure 4: Initial particle configuration for the bursting bubble case showing all particles with centroids within d_p of the centerline $z = L_z/2$.	11
Figure 5: Convergence of the solids volume fraction field (top), gas horizontal (x) velocity field, $Re_u = \rho_p d_p u_g/\mu_g$ (center), and gas vertical (v) velocity field, $Re_v = \rho_p d_p v_g/\mu_g$ (bottom) at the center of the domain, $z^* = L_z^*/2$, with increasing grid refinement.	13
Figure 6: Convergence behavior of the bed pressure drop in the bursting bubble case for uniform (blue) and grid-inverse (red) weightings with a least-squares power law extrapolation.	15
Figure 7: Convergence behavior of the bed pressure drop in the bursting bubble case with the FOU scheme (left) and the MUSCL scheme (right). Both cases show a least-squares power law extrapolation.	16
Figure 8: Direct comparison of results in Figure 6 and Figure 7: Superbee ($\square$), FOU ($\circ$) and MUSCL ($\times$) discretization schemes.	17
Figure 9: Convergence behavior of the bed pressure drop in the bursting bubble case filter widths of $\delta^* = 8$ (left) and $\delta^* = 4$ (right). Red least-squares curves are power law extrapolation and blue curve is mixed-order.	18
Figure 10: Convergence behavior of the bed pressure drop in the bursting bubble case filter widths of $\delta^* = 3$ (left) and $\delta^* = 2$ (right). Red least-squares curves are power law extrapolation and blue curves are mixed-order.	19
Figure 11: Variation in bed pressure drop with filter width, δ^* , on a numerical grid of $N_z = 40$ ($\Delta^* = 0.25$).	20
Figure 12: At left, absolute difference between mean calculated concentration and exact, analytical value. At right, maximum calculated solids concentration within the domain. Both shown as a function of the filter width, δ^* , on a numerical grid of $N_z = 40$ ($\Delta^* = 0.25$).	21
Figure 13: Convergence of the solids volume fraction field (top), gas horizontal (x) velocity field, $Re_u = \rho_p d_p u_g/\mu_g$ (center), and gas vertical (v) velocity field, $Re_v = \rho_p d_p v_g/\mu_g$ (bottom) at the center of the domain, $z^* = L_z^*/2$, with increasing grid refinement.	22
Figure 14: Same data as in Figure 10 here using least-squares power law extrapolation with $W_i = h_i^{-8}$.	23
Figure 15: Convergence behavior of the bed pressure drop in the bursting bubble case with the GARG_2012 interpolation scheme. Red least-squares curves are power law extrapolation and blue curves are mixed-order.	24
Figure 16: Data of Figure 15 compared to data in Figure 11, i.e., variation in bed pressure drop calculated with Garg_2012 interpolation on different numerical grids (red) compared to diffusion interpolation with different filter widths, δ^* , on a single numerical grid of $N_z = 40$	

($\Delta^* = 0.25$) (black). The horizontal lines span from $2\Delta^*$ to $3\Delta^*$ with the connecting curve at $2.5\Delta^*$	25
Figure 17: Initial particle configuration of original particle configuration (PC1) on left, a second replicate PC2 in center and a third replicate PC3 at right, which is also considered in a thermal condition (as shown), PC3t. For PC3t colors indicate particle velocity magnitude normalized by U_0 varying from 0 (black) to 0.1 (white). All particles shown have centroids within d_p of the centerline, $z=L_z/2$	26
Figure 18: Convergence behavior of the bed pressure drop in PC2 (left) and PC3 (right) showing least-squares power law extrapolations (red). On the right, the dashed line indicates the thermal case, PC3t.	27
Figure 19: Comparison of convergence behavior of the bed pressure drop in PC1, PC2 and PC3 and their respective least-squares power law extrapolations (red).	28
Figure 20: Convergence behavior of the bed pressure drop in the bursting bubble case with the FOU scheme (left) and the MUSCL scheme (right). Both compare to a least-squares power law extrapolation (red) with the coarsest grid neglected in the FOU case. Also shown for FOU are power law fit with inverse Fitted to FOU data, the coarsest grid has been neglected law with $W_i = h_i^{-2}$ (dashed) weighting and the mixed-order method (blue).	29
Figure 21: Direct comparison of results in Figure 18 (left) and Figure 20 Superbee ($\square$), FOU ($\circ$) and MUSCL ($\times$) discretization schemes on replicate PC2.	29
Figure 22: Convergence behavior of the bed pressure drop in PC3t using the GIDASPOW_BLEND (left), WEN_YU (center), and KOCH_HILL (right) drag laws showing least-squares power law extrapolations (red).	30
Figure 23: Convergence behavior of the bed pressure drop in PC2 (left) and PC3 (right) showing least-squares power law extrapolations (red).	31
Figure 24: On left, slice of PC5, particle configuration of SSCP-I simulation at time $t = 5$ s, showing particles with centroids within d_p of the centerline, $z=L_z/2$, colored by normalized velocity magnitude. At right, convergence behavior of the bed pressure drop compared to least-squares extrapolations of power law (red) and mixed-order (blue) regression types. .	35
Figure 25: Same as Figure 24 frozen at $t = 6$ s, i.e., PC6.	36
Figure 26: Same as Figure 24 frozen at $t = 7$ s, i.e., PC7.	36
Figure 27: Same as Figure 24 frozen at $t = 8$ s, i.e., PC8.	37
Figure 28: Same as Figure 24 frozen at $t = 9$ s, i.e., PC9.	37
Figure 29: Same as Figure 24 frozen at $t = 10$ s, i.e., PC10.	38
Figure 30: Convergence behavior of the bed pressure drop for PC5 (left) and PC6 (right) using the GIDASPOW_BLEND drag law compared to least-squares extrapolations of power law (red) and mixed-order (blue) regression types.	39
Figure 31: Same as Figure 30 for PC7 (left) and PC8 (right).	39
Figure 32: Same as Figure 30 for PC9 (left) and PC10 (right).	40
Figure 33: Comparison of the fixed-bed pressure drops with the BVK (red) and GIDASPOW_BLEND (blue) drag laws on the $N_z = 6$ grid with estimated numerical uncertainties. The results from the six fixed-bed PCs are superimposed over the complete transient results of the bed pressure drop predicted from the original simulation.	43

List of Tables

Table 1: Number of grid cells in the z-dimension of grids considered for bursting bubble cases.	12
Table 2: Summary of coefficients from least-squares extrapolation using power-law regression.	
.....	32
Table 3: Summary of coefficients from least-squares extrapolation using mixed-order regression.	
.....	33
Table 4: Summary of coefficients from least-squares extrapolation using power-law regression.	
.....	40
Table 5: Summary of coefficients from least-squares extrapolation using mixed-order regression.	
.....	41
Table 6: Estimated numerical uncertainties for PC5 – PC10 on grid $N_z = 6$.	42

Acronyms and Abbreviations

Term	Description
ASME	American Society of Mechanical Engineers
BC	Boundary condition
BVK	Beetstra, van der Hoef and Kuipers (drag law)
CFD	Computational fluid dynamics
DEM	Discrete element model
DNS	Direct numerical simulation
DOE	U.S. Department of Energy
DPVM	Divided particle volume method
FOU	First-order upwind
GCI	Grid convergence index
FWHM	Full width (at) half maximum (height)
LES	Large Eddy Simulation
MFIX	Multiphase Flow with Interface Exchanges
MUSCL	Monotonic Upwind Scheme for Conservation Laws
NETL	National Energy Technology Laboratory
PC	Particle configuration
QoI	Quantity of Interest
RANS	Reynolds-Averaged Navier-Stokes
RDE	Relative discretization error
SGS	Sub-grid scale
SRQ	System response quantity
SSCPI	Small Scale Challenge Problem-I
TFM	Two-fluid model
VVVQ	Verification, validation and uncertainty quantification

Symbols

Term	Description
Latin	
c, c_1, c_2	Constants (in extrapolation)
d_p	Particle diameter
DP^*_{1-2}	Non-dimensional pressure drop between elevations of $y = y_1$ and $y = y_2$
e	Restitution coefficient
g	Gravity
h_k	Grid size k normalized by the finest grid, $= \mathbb{H}_k/\mathbb{H}_1$
L_x, L_y, L_z	Domain length in x-, y-, and z-directions
n_g	Number of (different) grids
N_x, N_y, N_z	Number of grid points (cells) in x-, y-, and z-directions
N_{xyz}	Total number of grid points (cells) in domain, $= N_x \times N_y \times N_z$ for rectangular case
p	Order of convergence (order of accuracy); gas-phase pressure
$r_{i,i+1}$	Grid refinement factor, $= h_{i+1}/h_i$
R_f	Data range parameter
Re_p	Particle Reynolds number, $= \rho_p d_p (1-\phi) \Delta U / \mu_g$
Re_w	Phasic velocity nondimensionalized into Reynolds number, $= \rho_p d_p w / \mu_g$ (as opposed to $w^* = w/U$ non-dimensionalization)
S	Regression function to be minimized
t	Time
u, v, w	x-, y-, z-directional components of a velocity vector
U_0	Inlet superficial gas velocity
$\mathbf{U}, \mathbf{v}$	Velocity vectors
V_p	Particle volume, $= \pi/6 (d_p)^3$ for spherical particles
V_{tot}	Total domain volume, $= L_x \times L_y \times L_z$ for rectangular
w_i, W_i	Weighting factors, $w_i = W_i / \text{sum}(W_i)$
Greek	
α	Particle data (to be filtered onto continuum grid)
Δ^*	Grid size
δ^*	Filter width; Richardson-extrapolated error
ε	Numerical error/uncertainty

Symbols (cont.)

Term	Description
μ	Viscosity (gas), kinetic friction coefficient (particle)
ρ	Density
ν_f	Diffusion coefficient (filter width)
τ	Artificial time in diffusion equation
ϕ	Solids volume fraction, solids concentration
Sub- and Super-script	
*	Dimensionless
0	Approximate (extrapolated) exact solution, i.e., grid-less solution $k=0$
f	Filter or SRQol
i, k	Grid number, $i, k = 1$ set as the finest grid
p	Particle
x, y, z	Spatial coordinate dimension
W	Wall

Acknowledgments

This work was completed as part of National Energy Technology Laboratory (NETL) research for the U.S. Department of Energy's (DOE) Advanced Reaction Systems (ARS) Program. This technical effort was performed in support of the National Energy Technology Laboratory's ongoing research under the RES contract DE-FE0004000.

ABSTRACT

This report presents the study of the convergence behavior of the computational fluid dynamics-discrete element method (CFD-DEM) method, specifically National Energy Technology Laboratory's (NETL) open source MFiX code (MFiX-DEM) with a diffusion based particle-to-continuum filtering scheme. In particular, this study focused on determining if the numerical method had a solution in the high-resolution limit where the grid size is smaller than the particle size. To address this uncertainty, fixed particle beds of two primary configurations were studied: i) fictitious beds where the particles are seeded with a random particle generator, and ii) instantaneous snapshots from a transient simulation of an experimentally relevant problem. Both problems considered a uniform inlet boundary and a pressure outflow. The CFD grid was refined from a few particle diameters down to $1/6^{\text{th}}$ of a particle diameter. The pressure drop between two vertical elevations, averaged across the bed cross-section was considered as the system response quantity of interest. A least-squares regression method was used to extrapolate the grid-dependent results to an approximate "grid-free" solution in the limit of infinite resolution. The results show that the diffusion based scheme does yield a converging solution. However, the convergence is more complicated than encountered in simpler, single-phase flow problems showing strong oscillations and, at times, oscillations superimposed on top of globally non-monotonic behavior. The challenging convergence behavior highlights the importance of using at least four grid resolutions in solution verification problems so that (over-determined) regression-based extrapolation methods may be applied to approximate the grid-free solution. The grid-free solution is very important in solution verification and VVUQ exercise in general as the difference between it and the reference solution largely determines the numerical uncertainty. By testing different randomized particle configurations of the same general problem (for the fictitious case) or different instances of freezing a transient simulation, the numerical uncertainties appeared to be on the same order of magnitude as ensemble or time averaging uncertainties. By testing different drag laws, almost all cases studied show that model form uncertainty in this one, very important closure relation was larger than the numerical uncertainty, at least with a reasonable CFD grid, roughly five particle diameters. In this study, the diffusion width (filtering length scale) was mostly set at a constant of six particle diameters. A few exploratory tests were performed to show that similar convergence behavior was observed for diffusion widths greater than approximately two particle diameters. However, this subject was not investigated in great detail because determining an appropriate filter size is really a validation question which must be determined by comparison to experimental or highly accurate numerical data. Future studies are being considered targeting solution verification of transient simulations as well as validation of the filter size with direct numerical simulation data.

1. INTRODUCTION

Convergence tests have often been conducted rather casually, in which a codified numerical method is applied to two or three grid discretizations and the results overlaid on top of one another to qualitatively determine if the solution is converged or not. Converged solutions are then frequently treated as being free of numerical error, or grid independent. Solution verification is the formal study of numerical error within the still emerging field of verification, validation and uncertainty quantification (VVUQ) in which the error in a numerical solution of a particular problem is quantified, at least approximately. The most general procedure is similar to casual convergence tests: the problem is computed at several (two at the very least) grid resolutions and a system response quantity (SRQ) or Quantity of Interest (QoI), i.e., the code output the user is interested in (SRQoI is used here to cover all bases) to approximate an exact or grid-free solution, i.e., what the SRQoI would be in the limit of infinite grid resolution, through Richard extrapolation and the error is then calculated based on the difference between the (approximate) exact solution and the solution on a given grid. Other methods exist, e.g., method refinement, error transport equation and adjoint methods among others, (Oberkampf and Roy, 2010); however, grid-refinement/Richard extrapolation based methods remain the most popular in scientific computing, which is used in this work.

Solution verification has been primarily developed for and largely applied to single-phase flows and, more specifically, steady Reynolds Averaged Navier Stokes (RANS)-type models. Even in single-phase flows solution verification based on grid refinement breaks down in the case of Large-Eddy Simulations (LES). In LES the sub-grid scale (SGS) viscosity typically depends on the grid size, so, changing the grid essentially changes the model (Celik et al., 2005). Further, in the limit of infinite resolution, the SGS viscosity should approach zero (or in practical limits the SGS viscosity should become substantially less than the material viscosity), in which case a DNS model is recovered. The validity of estimating error in a LES from a level of grid refinement which has transitioned to a DNS is uncertain. A similar issue arises in the case of the computational fluid dynamics-discrete element method (CFD-DEM) implemented in NETL's open source MFIX code. The CFD-DEM method couples a traditional RANS-type CFD model for the gas-phase (including volume fraction and interfacial transfer terms like one phase of a two-fluid model (TFM)) with Lagrangian method for the solids where position and velocity of each particle are solved using Newtons' laws of motion, and, in the case of MFIX-DEM, a soft-sphere collision model for contacting particles (Garg et al., 2012a; 2012b). Filtering or interpolating schemes must be applied to transfer particle information, e.g., drag, to the continuum grid and vice versa.

As with LES, there are essentially two "philosophical" approaches to the transfer function: grid-coupled or grid-independent. In practice, CFD-DEM models typically use grid-coupled methods and the Garg scheme (Garg et al., 2007) employed in MFIX-DEM is the most commonly applied method among the user base. The grid-coupled approach also seems to fit with the idea that, like LES, CFD-DEM is a coarse-grained version of DEM-direct numerical simulation (DNS) (Fullmer and Hrenya, 2017). However, the grid-coupled approach presents a unique challenge to grid-refinement based solution verification methods which is not encountered in LES. Shown on the top of Figure 1, if the grid is refined below the scale of the particle, the interpolated values no longer occupy a region of space as large as the representative particle itself. In the case of solids volume fraction, some cells may have a very large value or even exceed unity. This sets a vague lower bound on how fine a grid-coupled interpolation-based CFD-DEM method can be refined that is somewhere in the vicinity of $\Delta^* \sim 1$, where $\Delta^* = (V_{tot}/N_{xyz})^{1/3}/d_p$ is the dimensionless grid

size, V_{tot} is the volume of the computational domain, N_{xyz} are the total number of grid points (cells) in the domain, and d_p is the particle diameter. Superficially, setting limitations on the level of refinement for a solution verification study is dissatisfying at best and, more critically, could be seriously problematic if the “asymptotic region” cannot be reached for a particular solution outside of this limitation. The concern is compounded by the fact that many studies have shown that the CFD-grid must be refined to a level of $\Delta^* \sim 2$ in order to achieve grid-insensitive solutions, see, for example, (Capecelatro et al., 2015; Liu et al., 2016).

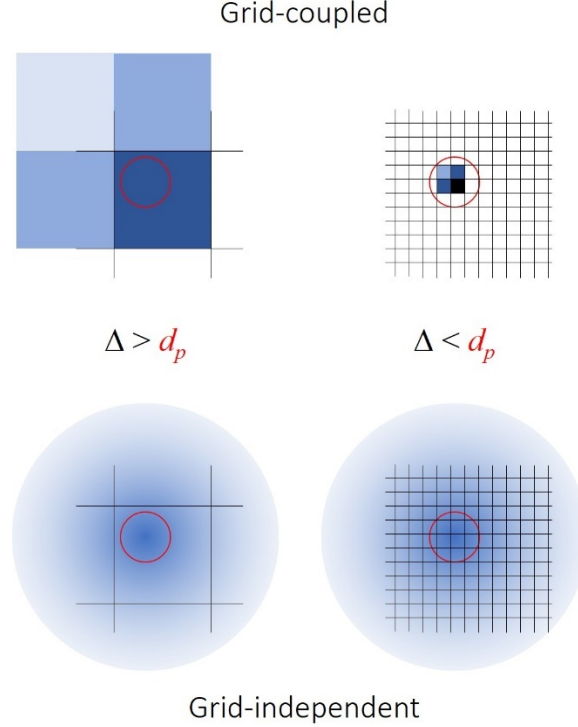


Figure 1: Sketch of two primary coupling schemes for Eulerian-Lagrangian numerical methods. The red circle indicates the particle scale, the black lines indicate the CFD grid and the blue scale indicates a filtered variable, e.g., V_p .

For the troubling resolution limitations of grid-coupled interpolation, we conclude that the grid-independent filtering method should be the preferred method for transferring particle and continuum information. As shown on the bottom of Figure 1, a grid-independent filtering method would give the same spatial distribution of weights regardless of the underlying CFD-grid. While this easily circumvents the resolution limitation challenge of grid-coupled interpolation schemes, some challenges arise with grid-independent filtering methods. From a “philosophical” view, one can no longer think of DNS as being the limiting case of CFD-DEM, completely breaking the similarity with single-phase LES-to-DNS models. Although this might break the multiphase numerical methods hierarchy for some, note that, unlike LES-to-DNS, the actual CFD-DEM model itself would have to be changed at some point to achieve a DNS type method. Specifically, terms like interfacial drag do not vanish (or become much less than other terms) in the limit of infinite resolution. From a more practical standpoint, grid-independent methods present a numerical challenge when the CFD grid becomes much smaller than the filter size. The lower right sketch in Figure 1 shows that the number of cells used in transferring information (interpolation stencil) between the grid and particle increases as the grid length scale is refined below the diameter of a

particle. Such an approach quickly becomes computationally intractable as the interpolation stencil grows. Furthermore, if distant cells reside in the memory space of a different processor than the particle (for distributed memory parallelism), the MFiX code fails as only two ghost layers surround each domain partition. (This could be remedied by increasing the ghost cell layers, however this also becomes computationally intractable for CFD grid sizes significantly smaller than the filter size.) Fortunately, grid-independent filtering can be achieved by a diffusion method rather than an explicit interpolation stencil based on the distance between the CFD grid and the particle centroid. The diffusion method of grid (size)-independent filtering is applied in this work, see Section 2.

The question of “*What filter size should be used?*” also arises. This is fundamentally a validation question, not a verification question (and will be discussed in Section 3.5). Further, the same question arises (or at least should arise) for grid-coupled interpolation schemes, but is impossible to isolate an answer to this question when the model and the numerical error both depend on grid. Whatever the filter size (maybe the selected value gives good agreement with DNS, maybe there are analytical reasons to choose a certain value, perhaps it is an arbitrary and poor selection, etc.), the verification question exists: *What would the given model predict in the mesh free limit, and what is the resulting error of a simulated result on a given grid?* For CFD-DEM, this is an open question and, as the title of the work suggests, it is largely unknown whether a grid-independent solution can even be achieved in the fine grid limit. In this work, we address this question in a preliminary manner by considering fixed-bed conditions, i.e., particles are not allowed to change position or velocity in time so that a steady-state solution is sought for the fluid at several different grid resolutions. Such a problem is similar to a porous media model, however the general CFD-DEM method implemented in MFiX is used throughout so that the results may be considered as instantaneous snapshots of grid-convergence of a transient CFD-DEM simulation.

It is valuable to note that one may question why CFD-DEM on a grid below the particle scale would even be considered at all. After all, DNS (for lower particle Reynolds numbers, Re_p) may be attained at resolutions of $\Delta^* \sim 1/6$. The simple answer is that all scientific computing methods should be subject to solution verification to quantify the numerical error present in a given solution, especially when that solution is used in decision making, safety analysis or may have any other real-world implications. Even if more practical solution verification of CFD-DEM use relatively coarse grids (e.g., $\Delta^* = 2, 2.6$ and 3.4 , are practical grid sizes and generally accepted as grid-insensitive resolutions within the community yet satisfy American Society of Mechanical Engineers’ (ASME) (Celik et al., 2008) suggested minimum $r = \Delta_{k+1}/\Delta_k \geq 1.3$, where k is the grid number), it is worth studying how well the error calculated from solutions on relatively coarse grids corresponds to similar calculations with relatively fine grids. Finally, it is worthwhile in general to study the applicability of MFiX CFD-DEM to cases with grid resolution at or smaller than the particle diameter as such instances can occur in practical applications, e.g., when high resolution of a gas-phase wall boundary layer is required (Capecelatro et al., 2016), in filtration and other fluidic devices where the flow area is smaller than the particle size (Baran et al., 2016).

2. METHODS

2.1 MODEL

The results reported in this report were generated with the CFD-DEM model available in NETL's open source MFiX code (<https://mfix.netl.doe.gov/>) version 2016.1. The theory and numerics of the CFD-DEM method have been documented elsewhere (Garg et al., 2012a; 2012b) and are not repeated here for the sake of brevity. Instead, we focus on: *i*) a few, slight modifications to the 2016.1 version that were necessary for this work, and *ii*) the diffusion method for particle-to-grid interpolation which has not been previously documented for the MFiX code.

The modifications and alterations to the 2016.1 code base used in this study included:

- The most important modification is a simple one character insertion. This study simulated a steady-state problem, which is not available in the DEM side of the code. Furthermore, the diffusion step was only taken at the start of a fluid timestep, so if the code were to be run in a steady-state mode, the diffusion step would need to be placed inside of the iteration step. It is much easier to simply comment out the CALL DES_TIME_MARCH, effectively eliminating any DEM calculations. In the 2016.1 version of the code, this call occurred on line 160 of time_step.f which was simply commented out.
- Since the steady-state problem is run in a transient mode, a new residual (or error) occurs which is due to the transient term. In theory, the solution will reach steady-state when $\partial u_g / \partial t = \partial v_g / \partial t = \partial w_g / \partial t = \partial p_g / \partial t = 0$. In practice, however, a solution is observed to be converged when these transient residuals reach a certain tolerance or threshold level. What this tolerance should be is unknown *a priori* and is investigated in Section 2.2. It should be noted that the full transient term is taken as a residual, i.e., including the time-step in the denominator, since the difference between new and old time values may become very small if the time step is adapted to be very small. The terms are also non-dimensionalized based on particle diameter, d_p , and the distributor inlet velocity, U_0 , i.e., $u^* = u/U_0$, $x^* = x/d_p$, $t^* = td_p/U_0$, and $p^* = (p - P_0)/\rho_g U_0^2$, where the value of P_0 is irrelevant as pressure differences are only considered in this work. The user defined hooks used to monitor the transient residuals are provided in the Appendix, Section A.1.
- During the transient simulation, MFiX will increase the time step if a certain set of criteria are satisfied indicating good convergence behavior, i.e., specified iterative residuals are being met quickly. In this case, increasing the time step is favorable to reaching a state of steady-state convergence (transient residual tolerances being satisfied). Therefore, the number of steps taken before performing a time step increase check is reduced. This value is controlled by the step_min parameter in the function ADJUST_DT which was reduced to a value of 2. In the 2016.1 version the step_min parameter is located on line 852 of the iterate.f subroutine.
- There is a known limitation in MFiX-DEM when the CFD grid becomes smaller than the particle diameter, which really begins at a grid resolution of about 3 particle diameters if the default DES_GRIDSEARCH parameters are used. The issue is this: when periodic boundaries are applied, if the DES grid (which is used for neighbor detection) is larger than the fluid grid, ghost particles are copied to the boundary layer which can be outside of the ghost fluid cell layer. This can be easily circumvented by forcing the DES grid to the same size as the CFD grid, which is applied in this work. This solution may be fine in practice

for modest grid sizes, however, when the grids are refined below the particle size, close neighbors (and even collisions if the grid is small enough) will be missed. This is not an issue in the current study, since the dynamics of the DEM model have been completely removed. Therefore, the internal code checks on the DES grid size are modified so that the code does not exist when these checks fail, although the error message was still reported. In the 2016.1 version of MFiX, three `ABORT=.TRUE.` assignments were changed to `ABORT=.FALSE.` on lines 148, 157, and 168, in the `check_geometry.f` subroutine.

- Finally, two coding errors were uncovered during this study in the `comp_mean_fields1.f` and `diffuse_mean_fields.f` subroutines. The details of the errors are not reported as the bug fixes were patched into the developmental version of the code and subsequently released under version 2017.1. It is recommended to use this version or later for repeating this study or undertaking a similar one.

Section 1 discussed that there are essentially two particle-to-grid interpolation methods: grid-coupled and grid-independent, both of which are available in MFiX. GARG_2012 interpolation (Garg et al., 2007) is a grid-coupled scheme, was the first method available in MFiX, and still receives the most widespread usage among the user base today. More recently, the Square Divided Particle Volume Method (DPVM) was implemented; in theory, this method is grid-independent. Practical limitations prevent this method from being applied to very fine grids; however, this could be circumvented by increasing particle neighborhoods to more than the nearest 27 cells around particle centroids. Unfortunately, this is not a practical solution as discussed previously in Section 1. Another way to arrive at a grid-independent Gaussian filter is to solve a diffusion equation (Capecelatro and Desjardins, 2013; Sun and Xiao, 2015),

$$\frac{\partial \alpha}{\partial \tau} = \nu_f \nabla^2 \alpha, \quad (1)$$

where α is the data associated with the particle to be filtered, e.g., volume (concentration), drag force, etc., τ is an artificial time and ν_f is a diffusion coefficient. The diffusion coefficient and the (artificial) time for which the equation is solved, τ_f , are related to the width of a Gaussian filter of the transfer process,

$$\nu_f \tau_f = \frac{\max(\delta_f^2 - \Delta^2, 0)}{16 \ln 2}, \quad (2)$$

where δ_f is the Gaussian filter full width at half maximum (FWHM) and Δ is the CFD grid size. Note that for a Gaussian the FWHM is related to the standard deviation by a factor of $2\sqrt{2 \ln 2}$. In MFiX, the start and end artificial times are set to 0 and 1, which is solved in five steps. The filter width is specified user input. In this study, the baseline case considered $\delta_f = 6d_p$ ($\delta_f^* = 6$). It was determined (in some cases at least) that an accurate solution of the linear solver for this equation was necessary to meet certain volume fraction tolerances within the code. As shown in the baseline example input deck in Section A.2 of the Appendix, the tolerance of the linear solver is set to 10^{-10} and the maximum number of linear solver iterations is set to 100.

2.2 CONVERGENCE

There are three errors associated with the numerical solution of this type problem: 1) since the steady-state problem is solved as a transient, there is a transient residual which should (hopefully) approach zero as the simulation progresses; 2) the discretized equations are solved in a linearized, semi-implicit (SIMPLE-type) fashion which has an iterative residual associated with the coupled, nonlinear terms; and 3) at any given iteration within a timestep, the matrix equations are solved with an iterative method which as a linear equation solver tolerance from the computed solution not exactly satisfying the original matrix equation. This study used a small linear equation solver tolerance, 10^{-8} for all equations except the diffusion equation which is even lower at 10^{-10} . These values are at least two orders of magnitude smaller than the transient and iterative residual tolerances, even more in the diffusion case, and is therefore assumed to contribute negligible error to the solutions.

As mentioned previously, the transient residual is based on the non-dimensionalized temporal gradients of the four gas-phase primary variables. The largest single value of each within the domain (L_∞ metric) is verified and then the maximum of the four terms is determined, i.e.,

$$\max\left[\left(\frac{\partial u_g^*}{\partial t^*}\right)_\infty, \left(\frac{\partial v_g^*}{\partial t^*}\right)_\infty, \left(\frac{\partial w_g^*}{\partial t^*}\right)_\infty, \left(\frac{\partial p_g^*}{\partial t^*}\right)_\infty\right], \quad (3)$$

Finally, the average of the current and previous transient residuals in Equation 3 must satisfy the specified tolerance for the simulation to be considered converged to steady-state. The iterative residuals are calculated using the standard MFiX method with for the maximum continuity + momentum residual, i.e., TOL_RESID, normalized based on the dominant term in the continuity equation, i.e., NORM_G = 0.

The tolerance limits of the other two residuals should be set to a level low enough not to affect the results of the solution, but also as large as possible to achieve a solution in a reasonable amount of time. The values used in this study were arrived at after a rather lengthy trial and error phase and only the final (useful) results of this study are shown here. It was determined that iterative and transient tolerances of approximately 10^{-5} were sufficient to achieve converged solutions. Figure 2 shows the convergence behavior of the pressure transient residuals for iterative tolerances of 10^{-5} and 10^{-6} on the left and right, respectively, for the bursting bubble case described in Section 3.1. Some cases fall below the 10^{-5} tolerance level indicating in these cases one of the velocity residuals was slower to converge.

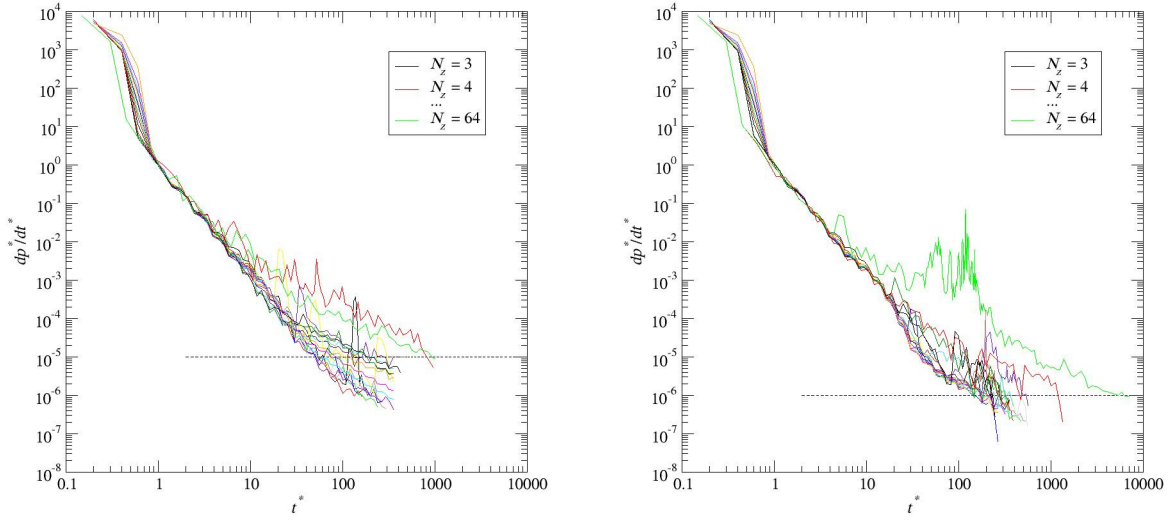


Figure 2: Convergence of the non-dimensionalized pressure temporal residuals for all grids. Horizontal lines show selected tolerances.

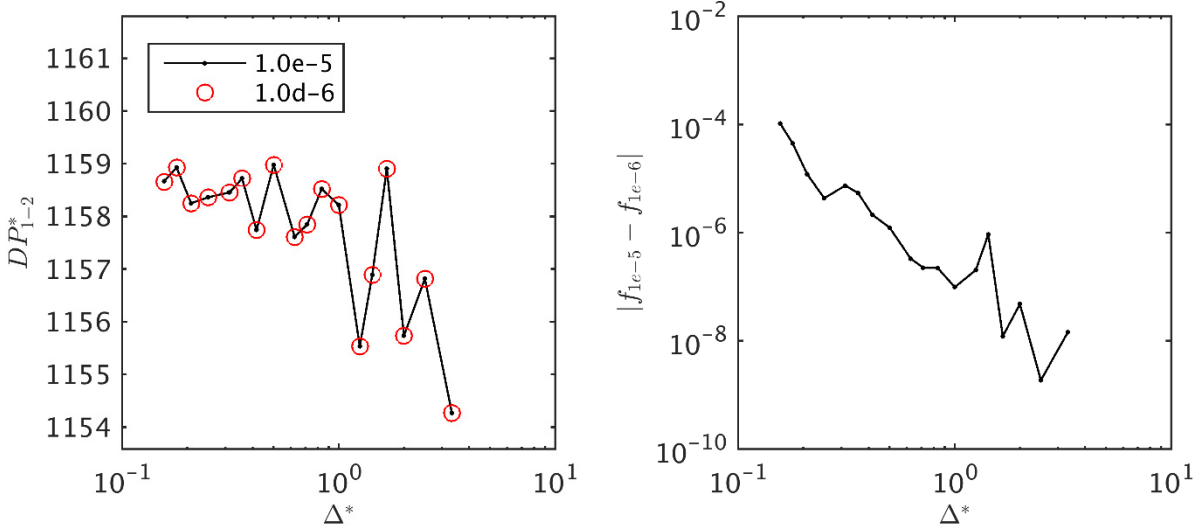


Figure 3: At left, comparison of the SRQoI for the bursting bubble case with (both) transient and residual tolerances set to 10^{-5} and 10^{-6} . At right, absolute difference between computed solutions.

In Figure 3 the SRQoI for the bursting bubble case for the two tolerance levels, 10^{-5} and 10^{-6} , is compared as a function of grid resolution showing essentially identical results. The absolute difference of the two sets of solutions is provided on the right of Figure 3 which shows that there is a difference in the two solutions produced from the different tolerances levels; however, it is rather small, especially considering the magnitude of the SRQoI. The L_∞ -norm of this error, defined as

$$e_{\infty} = \frac{\max |f_k^{(1e-5)} - f_k^{(1e-6)}|}{\max |f_k^{(1e-5)}|}, \quad (4)$$

is just 9.0×10^{-8} and the L_2 -norm of this error, defined as

$$e_2 = \frac{\|f^{(1e-5)} - f^{(1e-6)}\|^2}{\|f^{(1e-5)}\|^2}, \quad (5)$$

is just 2.3×10^{-8} . Hence, tolerance limits of 10^{-5} for both iterative and transient residuals are used throughout and the remaining residual errors from these sources (and the linear solver residual errors) are assumed to be negligible.

2.3 EXTRAPOLATION METHODS

As mentioned in Section 1, grid-refinement based methods were used to quantify the error in numerical solutions and, more importantly for this study, determine if there even exists a reasonable grid-free limit to the solutions. There are several different approaches to Richard extrapolation such as Roach's grid convergence index (GCI) (Roache 1994; 1998), Roy's mixed order method (Roy, 2003), power law and approximate error spline methods (Celik and Li, 2005), etc. An important variation, which may be applied within the context of several of the approaches mentioned above, is when the system is over-determined, i.e., when there are more solutions (grids) than needed to calculate the unknowns of the function used for extrapolation. Of course, one may simply take the SRQoIs from the three (or however many are required) finest grids—which amounts to distributing an importance weighting uniformly among the highest resolution results—to perform an extrapolation and neglect the coarser solutions. This seems to be a justifiable approach and it appears to work quite well in some instances, e.g., see the curve in Figure 11 of Roy (2003) which is extrapolated from the three finest solutions yet compares quite favorably over the entire region studied. Another approach is to include all of the SRQoIs and find the unknowns of the over-determined extrapolation as the solution to a regression (Eça and Hoestra, 2014) or optimization (Rider et al., 2016) problem. Such methods also have the favorable property that oscillatory or noisy convergence behavior is fit much more reasonably. A quick glance at Figure 3 (left) shows that noisy, oscillatory convergence will indeed be an issue here.

This study primarily used Richardson-type extrapolation based on power laws,

$$\delta_{RE} = f_i - f_0 = c h_i^p, \quad (6)$$

and mixed-order (first and second),

$$\delta_{RE} = f_i - f_0 = c_1 h_i + c_2 h_i^2, \quad (7)$$

error estimators. Both Eqs. (6) and (7) contain three unknowns (f_0 and c and p or c_1 and c_2). Therefore, if three different grid levels are used to compute the solutions, Eqs. (6) and (7) have a unique solution. If there are more than three grid level solutions (as there are here), Eqs. (6) and (7) are overdetermined and can only be solved approximately, i.e., in an optimization framework. Here, least-squares regression was used, i.e., seeking the minimum of the sum of the error squared, given by

$$S_{RE}(f_0, c, p) = \sqrt{\sum_{i=1}^{n_g} w_i \left(f_i - (f_0 + c h_i^p) \right)^2}, \quad (8)$$

for the power law estimator and

$$S_{RE}(f_0, c_1, c_2) = \sqrt{\sum_{i=1}^{n_g} w_i \left(f_i - (f_0 + c_1 h_i + c_2 h_i^2) \right)^2}, \quad (9)$$

for the mixed-order estimator. The minimum of S_{RE} is determined by setting the partial derivatives with respect to each unknown equal to zero. For the power-law fit this results in three equations which must be solved iteratively and for the mixed-order fit a 3x3 matrix equation which can be solved directly. The corresponding equations for each can be found in Appendix B of Eça and Hoekstra (2014).

The new variable w_i appearing in Equations 8 and 9 is a weighting function. This is not an additional unknown, but a parameter that should be applied by the user before regression. If there is no reason to weight some solutions over others, then a uniform weighting function should be specified: $W_i = 1$ where $w_i = W_i / \text{sum}(W_i)$, i.e., $w_i = 1/n_g$ where n_g is the number of grids. In CFD, the results obtained on fine grids are more accurate than those on coarse grids (which is justifiable assuming the numerical scheme is consistent), and so weightings based on the grid spacing are also common and used here, primarily $W_i = 1/h_i$.

3. CASE 1: BURSTING BUBBLE

3.1 SYSTEM AND CONDITIONS

The first case studied is a simple, hypothetical problem which (loosely) resembles a dilute bubble about to burst at the surface of a dense bed. The problem is completely fictional and generated with a slightly modified particle input generator. The source code for the particle generator is provided in the Appendix in Section A.3. (Note that this source code is for the thermal case discussed in Section 3.7. To get static cases, simply change the write statement near the end to print 0.0d0 in place of the particle velocities.) The geometry of the domain is 30 particle diameters wide ($L_x^* = 30$), 200 particle diameters tall ($L_y^* = 200$), and 10 particle diameters deep ($L_z^* = 10$). The particles only occupy the lower quarter of the domain ($y^* \leq 50$) and are separated into three regions: a dilute “freeboard” region above surface, a “dense bed” region below the surface, and a single dilute “bubble” inside the bed near the surface. The regions are defined only in the xy -plane and (statistically) uniform in the z -dimension. The bubble diameter is 10 particle diameters and is centered at $(x^*, y^*) = (15, 25)$. The bed surface is defined by $y^* = 30 + 5 \exp[-(x^* - 15)^2/50]$. The dilute regions have a specified concentration of 1% and the dense region has a concentration of 30%. Based on the specified volume and concentration, the problem contains just 5,181 particles in total. The particles are distributed randomly and no particle is in contact with any other particle. In the baseline case, all particles have null velocity, i.e., static bed. A slice of the initial particle configuration is presented in Figure 4 for all particles within one diameter of the z -centerline (plane).

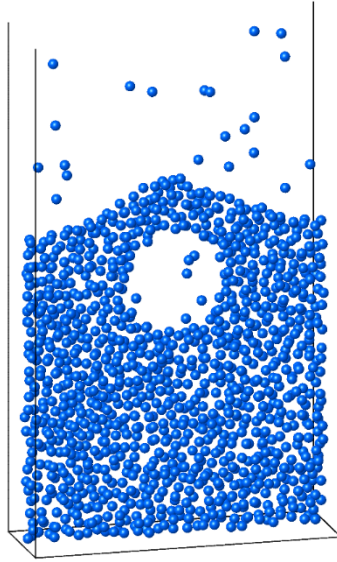


Figure 4: Initial particle configuration for the bursting bubble case showing all particles with centroids within d_p of the centerline $z = L_z/2$.

The x - and z -dimensions are treated as periodic (cyclic BCs). In the y -direction, the inlet plane, $y^* = 0$, is a uniform mass inflow BC and the exit plane, $y^* = L_y^*$, is a pressure outflow BC. The inlet gas velocity is set to $U_0 = 20$ cm/s. With the specification $\rho_g = 1 \times 10^{-3}$ g/cm³, $\mu_g = 2 \times 10^{-4}$ g/cm-s

and $d_p = 100 \mu\text{m}$, the particle Reynolds number at the (single-phase) inlet is unity, $Re_{p0} = \rho_g d_p U_0 / \mu_g = 1$. Gravity is set to zero. The particle density, although irrelevant in this case, is $\rho_p = 1 \text{ g/cm}^3$. The initial condition is uniform with values $\phi(t=0) = 0.18$, $u_g(t=0) = 0$, $v_g(t=0) = 24.4 \text{ cm/s}$, and $w_g(t=0) = 0$. The gauge pressure at the outlet is $p_g(y^* = L_y^*) = 0 \text{ Ba}$. The filter width (FWHM) is initially set to 6 particle diameters, $\delta_f^* = 6$. The BVK drag law (Beetstra et al., 2007) is specified in the baseline case. The baseline case also uses the MFIX recommended Superbee flux-limiter (Waterson and Deconinck, 2007) which is applied to all equations, although not used in the diffusion equation which does not contain first-order spatial derivatives.

3.2 GRID CONVERGENCE

For the bursting bubble cases, 18 different grids have been studied. The number of grid cells in the z -dimension are listed in Table 1. In all cases, the number of cells in the x - and y -dimensions are $N_x = 3N_z$ and $N_y = 20N_z$, consistent with the L_x^*/L_z^* and L_y^*/L_z^* ratios of the domain. This results in a uniform discretization, $\Delta_x^* = \Delta_y^* = \Delta_z^* = \Delta^*$, which becomes simply $\Delta^* \equiv (V_{tot}/N_{xyz})^{1/3}/d_p = L_z^*/N_z = 10/N_z$. Hence, the grids represented in Table 1 span from $\Delta^* = 3.3$ down to 0.16. The finest of these grids reaches down into the resolution range used in DNS of particulate flows, at least for moderate particle Reynolds numbers (Fullmer et al., 2017).

Table 1: Number of grid cells in the z -dimension of grids considered for bursting bubble cases

N_z				r
-	3	-	4	1/2
5	6	7	8	-
10	12	14	16	2
20	24	28	32	4
40	48	56	64	8

The first check performed is to see whether or not the problem appears to converge qualitatively. Indeed, it is observed that as the grid is refined, the concentration and flow profiles do appear to approach a unique solution that should exist in the limit of infinite resolution. Figure 5 shows key profiles along the center ($z^* = L_z^*/2$) of the domain for several grid resolutions. The concentration field is set by the particle configuration and converges to the solution of the diffusion equation as more points are used to represent the Gaussian kernel of the filtering operation. This result is to be expected. The convergence of the velocity fields are more noteworthy. Although the drag force is filtered out over the same distance (volume) defined by the filter width, δ_f^* , the force being filtered depends on a local velocity—namely, the gas-phase velocity in the cell containing the centroid of the particle—which will continue to change as the grid is resolved. Hence, there is not the same guarantee of a unique, convergent solution as in the case of simply filtering particle locations into a concentration field.

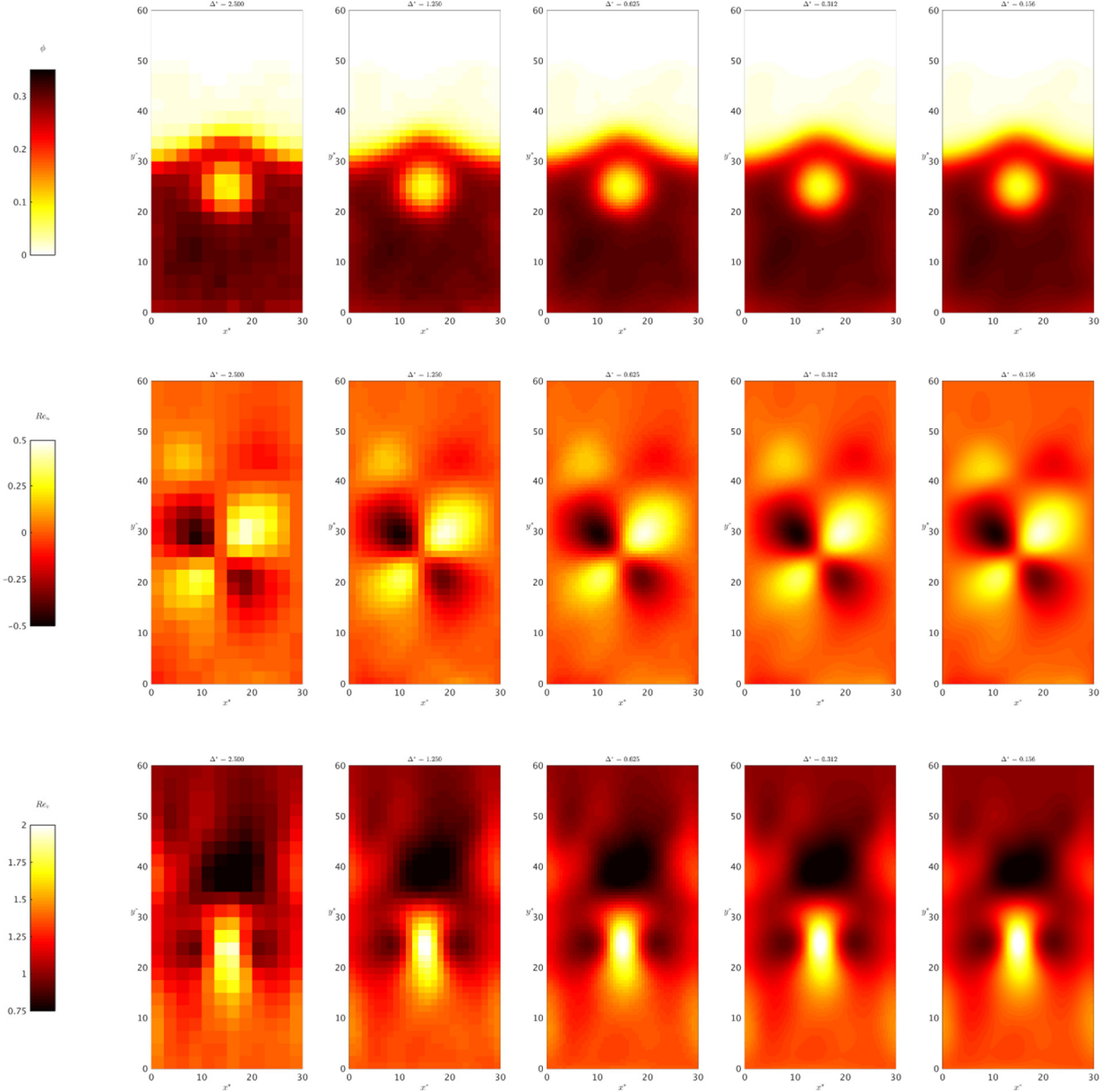


Figure 5: Convergence of the solids volume fraction field (top), gas horizontal (x) velocity field, $Re_u = \rho_p d_p u_g / \mu_g$ (center), and gas vertical (v) velocity field, $Re_v = \rho_p d_p v_g / \mu_g$ (bottom) at the center of the domain, $z^* = L_z^*/2$, with increasing grid refinement.

Before proceeding, we note that the previous discussion may raise the questions: “*Should we resolve below $\Delta^* \sim 1$?*”, and, “*In CFD-DEM should we calculate the drag on a particle using a velocity obtained from a cell smaller than a particle diameter?*” Consider that for $\Delta^* < 1$, the gas-phase velocity field varies “within” a particle, i.e., the gas velocity may vary among the cells within a radius of the particle’s centroid. While perhaps somewhat dissatisfying, the short answer is that this is a validation question (*Does the model accurately represent the physics?*), not the type of verification question addressed in this work (*What is the numerical error of a given solution*

obtained for a particular?). Here, questions of whether or not the model is physically consistent, or whether or not the model is physically accurate are not of concern. This verification study is focused on whether or not the model converges to a solution in the limit of infinite resolution and (to a lesser extent, but more importantly in practice), quantifying the numerical (grid) error of a given solution. It should be pointed out that, at least qualitatively for the $\delta_f^* = 6$ case displayed in Figure 5, the velocity fields appear sufficiently smooth from the drag filtering that even the local velocities in the very fine grid cases are not representative of DNS-type microscale velocities, which may vary appreciably on a length scale smaller d_p , but are more representative of a diffused velocity field, which varies on a length scale proportional $\delta_f^* > d_p$.

3.3 BASELINE RESULTS

The previous section discussed that the problem seems to converge qualitatively. More quantitative metrics are pursued here in the form of an extrapolated least squares curve. In order to apply the methods in Section 2.3, an SRQoI, f , needs to be identified to calculate on each grid. Since this is a fictional problem, there is no obvious choice for the SRQoI. The gas-phase pressure drop is one of the most widely studied metrics in gas-solids flows and seems like a natural choice here. Therefore, the following option was chosen:

$$f = DP_{1-2}^* = \frac{\bar{p}_g(y^*=y_1^*) - \bar{p}_g(y^*=y_2^*)}{\rho_g U_0^2}, \quad (10)$$

where $\bar{p}_g(y^* = y_j^*)$ is the gas-phase pressure averaged across the xz cross-section at an elevation of y_j^* with $y_1^* = 4$ and $y_2^* = y_1^* + 100$. The choice of $y_1^* = 4$ intentionally avoids the first row of grids cells along the inlet in all cases and also does not lie along the cell centerline of any of the grids studied. Second order linear interpolation is used between the two rows of cells adjacent to y_1^* and y_2^* .

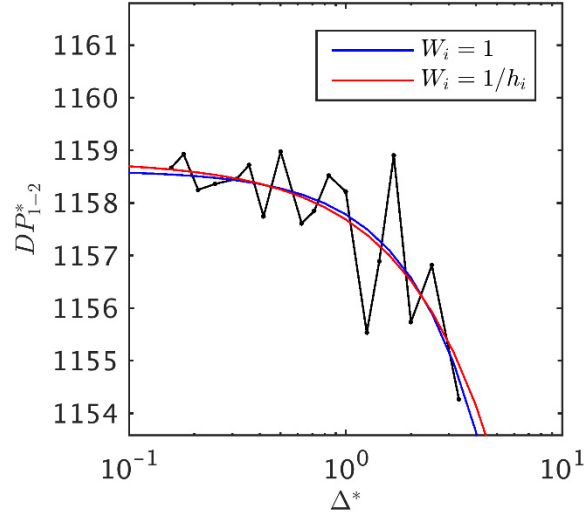


Figure 6: Convergence behavior of the bed pressure drop in the bursting bubble case for uniform (blue) and grid-inverse (red) weightings with a least-squares power law extrapolation.

DP_{1-2}^* for the “baseline” case was shown previously in Figure 3 when testing the transient and iterative residual tolerances and is repeated in Figure 6 with two least-squares curve fits, both using the power law extrapolation of Equation 8. The difference between the extrapolations in Figure 6 is the weighting factors, the blue curve using a uniform weighting, $W_i = 1$, and the red curve using a weighting inversely proportional to the grid cell size, $W_i = 1/h_i$, or, equivalently, $W_i = 1/\Delta_i^*$. There is very little difference between the two weighting schemes. The approximate exact solutions are virtually indistinguishable, $f_0 = 1158.6$ and 1158.8 for uniform and grid inverse weightings, respectively. There is a larger discrepancy in the predicted order of accuracy, $p = 1.29$ and 1.03 , the uniform weighting with the larger order since there is more variability in the coarser solutions. The remainder of this report utilizes the grid inverse weighting factors except where specifically noted. In general, it is observed that the regression-based method of using all of the available data to determine the coefficients of the extrapolation produces a nice, well-fit curve that follows the general, global trend of the results. It can be seen that by picking just two or three points along this curve and performing a traditional (determined) extrapolation might result in an exact solution very far away from the data and an excessively large error. Before moving on, note that possibly the most important take away from the baseline case, beyond the value of regression-based extrapolation, is that the grid has been refined from a typical coarse-grained CFD-DEM grid length scale ($\Delta^* \geq 3$) down to the boundary of DNS grid length scale ($\Delta^* \leq 1/6$) and the pressure drop has changed by less than 0.5% [$(\max f - \min f)/(\max f + \min f)/2 \times 100$], a very welcome result.

3.4 DISCRETIZATION SCHEMES

Next, we deviate from the baseline case/conditions by exploring the impact of two additional numerical schemes: first-order upwind (FOU) and van Leer’s (1979) MUSCL flux limiter. Figure 7 shows the results and the respective (grid inverse weighted) least squared power law

extrapolation. Interestingly, the FOU scheme appears to converge from above, while the higher-order schemes (MUSCL and Superbee scheme in the baseline case) converge from below. This trend may not continue if the grid were coarsened further, i.e., the global trend of the FOU solutions may also show decreasing DP_{1-2}^* with increasing Δ^* . This point will be touched on again in Section 3.8. The predicted convergence rate for these two schemes are $p = 0.14$ and 1.50 , for FOU and MUSCL which may be compared to $p = 1.03$ previously calculated for Superbee with grid inverse weighting. None of the predicted convergence rates calculated here match the expected rates of the schemes, either calculated analytically or from previous performance results on significantly simpler problems (Waterson and Deconinck, 2007). This may not be so surprising, given the complexity of this system, even with the frozen particles. However, it is reassuring that at least the same ranking of the three schemes was recovered from a scalar convection case (i.e., even simpler than single-phase Navier-Stokes): $p^{(\text{FOU})} < p^{(\text{Superbee})} < p^{(\text{MUSCL})}$.

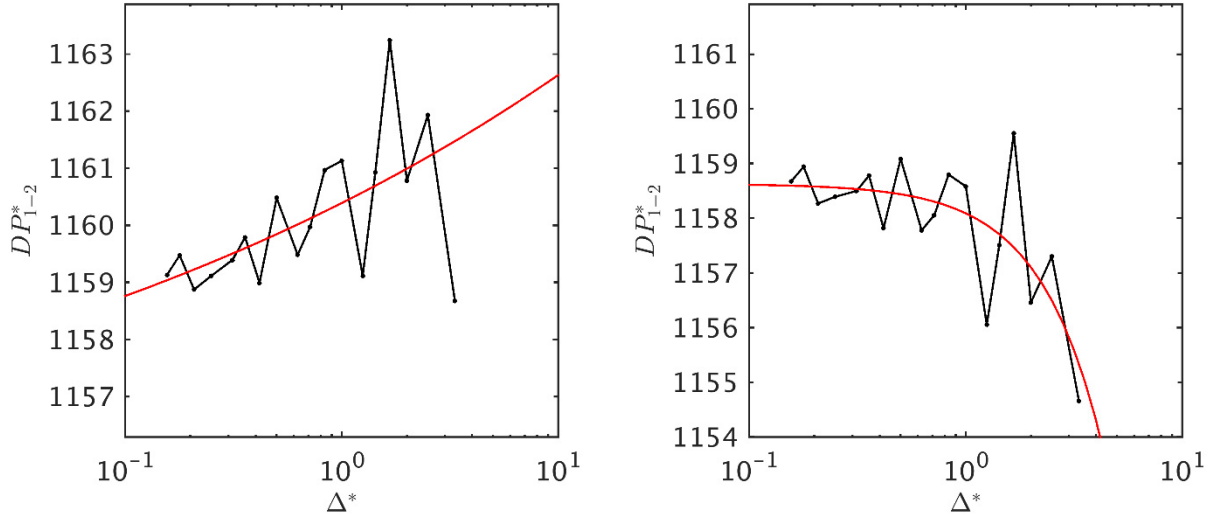


Figure 7: Convergence behavior of the bed pressure drop in the bursting bubble case with the FOU scheme (left) and the MUSCL scheme (right). Both cases show a least-squares power law extrapolation.

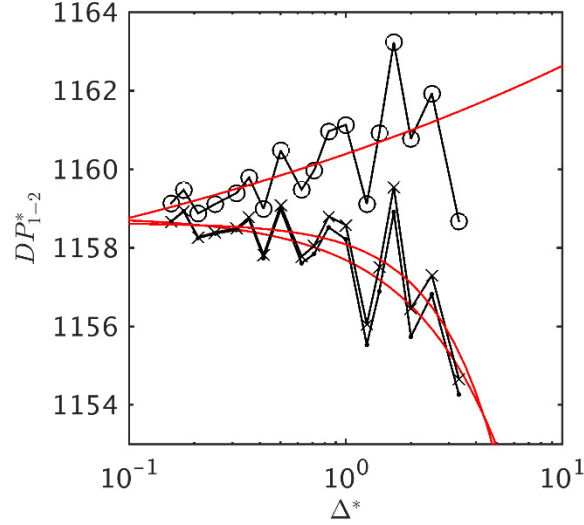


Figure 8: Direct comparison of results in Figure 6 and Figure 7: Superbee (⌈), FOU (○) and MUSCL (×) discretization schemes.

The results and extrapolated curves for the three numerical schemes are compared directly to one another in Figure 8. Again, it is encouraging that the spread in results for all schemes and on all grids is relatively small. The higher-order schemes give a very similar result in Figure 8 and the difference decreases with decreasing Δ^* and visually appears to vanish altogether at the finer grids studied here. The results from the FOU scheme are also converging to the results of the higher-order schemes, though some observable differences are still present in the finest grid. It is also somewhat curious that the oscillation pattern is reproduced nearly identically between the three schemes. It is hypothesized that this pattern emerges due to changes in the flow pattern with grid refinement, i.e., the oscillations are a manifestation of the dependence of drag force calculation on the local velocity which continues to change with refinement. However, note that the magnitude of the oscillations decays with refinement as the values used in the calculations approach point-wise (infinitely resolved) values. It is worth reiterating that while the drag does depend on local values (which is becoming more and more representative of the conditions at the singular centroid locations with grid refinement), these local values do not appear to show the same type of flow microstructure, $\ell \leq d_p$, as in DNS simulations, but are representative of local values on filtered features with $\ell \sim \delta_f$. For an example of DNS-type microstructure flow resolution, see several of the figures in Tenneti and Subramanian (2014) and in many of the references cited therein.

3.5 FILTER WIDTH

In this study, we have tried to make clear the distinction between problems of validation and problems of verification. A filter width of $\delta_f^* = 6$ was selected arbitrarily and has been utilized so far. An assessment of the most appropriate choice of filter length cannot be determined here, however, one question does arise, “*Would we get similar convergence behavior with a different, equally arbitrary choice of δ_f^* ?*” To provide some insight into this question, the impact of δ_f^* on the convergence behavior was explored, yet readers are cautioned that this is not a validation study of which is the most appropriate value of δ_f^* .

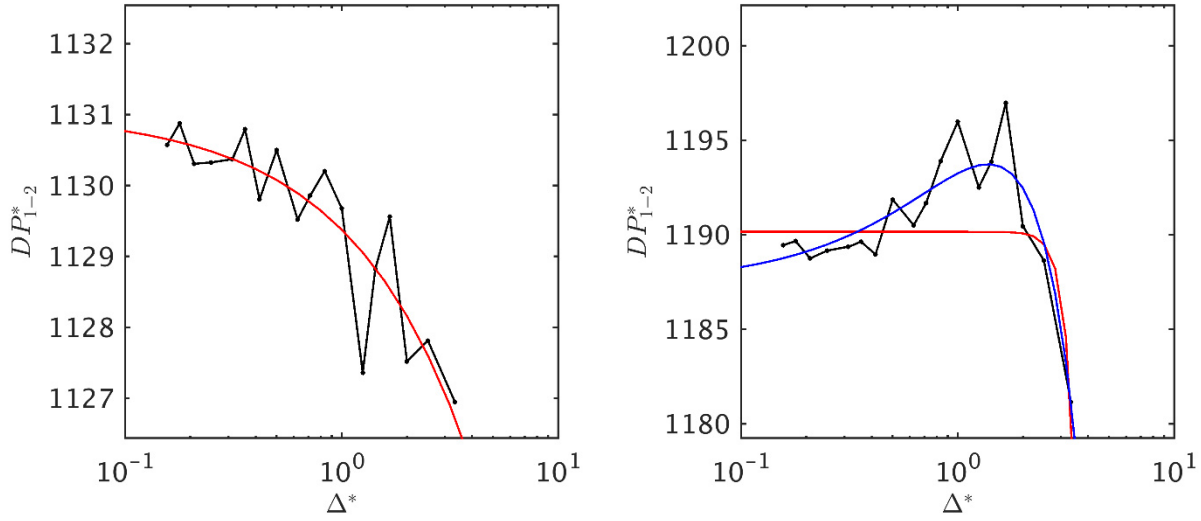


Figure 9: Convergence behavior of the bed pressure drop in the bursting bubble case filter widths of $\delta_f^* = 8$ (left) and $\delta_f^* = 4$ (right). Red least-squares curves are power law extrapolation and blue curve is mixed-order.

Figure 9 is similar to Figure 6 with a slightly larger (left) and slightly smaller (right) filter widths. The larger filter width, $\delta_f^* = 8$, has a slightly lower order of convergence than in the baseline case, $p = 0.79$, which could be inferred from the global trend of the results which are continuing to increase more so than in the baseline case. This is a somewhat surprising result given that the filter resolution essentially increased, i.e., δ_f^*/Δ_i^* has increased.

The smaller filter width, $\delta_f^* = 4$, displays a different global behavior than observed previously. On average, δ_f^* increases and then decreases, again with a superimposed oscillation. In this case, the predicted order of convergence is absurdly high, $p = 9.16$. Although there were no expectations in mind for p before undertaking this study, it may be expected that the p should be in the vicinity of 1 to 2 based on the lowest and highest order of accuracy of the employed numerical schemes. (Note that even Superbee defaults to FOU in the presence of extrema (Waterson and Deconinck, 2007).) A convergence rate of $p = 9.16$ seems too high—and it is. Essentially, the extrapolation based on a single term cannot handle the “globally nonmonotonic” trend in the data which indicates there at least two terms present of opposing signs. For this case, the mixed-order (powers of 1 and 2) regression of Equation 9 were applied. Shown as the blue curve in Figure 9, the mixed-order least squares extrapolation appears to fit the “globally nonmonotonic” data much better than the single power law extrapolation. Note that the solution verification method of Eça and Hoekstra (2014) also provide guidelines on when to deviate from the single power law extrapolation, which is also based on anomalous p predictions.

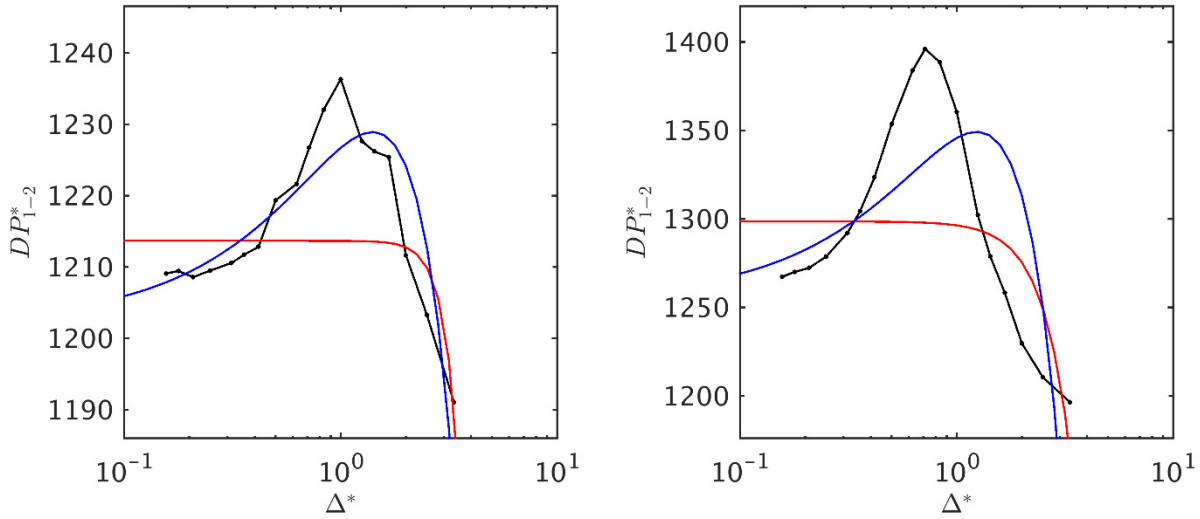


Figure 10: Convergence behavior of the bed pressure drop in the bursting bubble case filter widths of $\delta_f^* = 3$ (left) and $\delta_f^* = 2$ (right). Red least-squares curves are power law extrapolation and blue curves are mixed-order.

The filter width was further reduced to $\delta_f^* = 3$ and 2 on the left and right side of Figure 10, respectively. Like the $\delta_f^* = 4$ case, both smaller filter width cases present “globally nonmonotonic” behavior and the single power law extrapolation predicts artificially large convergence rates of 6.40 and 3.37. The mixed-order method again provides a better fit, however the strong nonmonotonic behavior for $\delta_f^* = 2$ is not fit particularly well.

It can also be observed, for the cases presented in Figure 10, that the amplitude of the oscillations decay with decreasing δ_f^* and are almost absent in the $\delta_f^* = 2$ case. The reason for this change in behavior seems due to the difference in filter resolution, δ_f^*/Δ^* . For $\delta_f^* \leq 4$ and below, the filter is not well resolved on the coarser grids or even resolved at all. In these cases, the grid needs to be refined to a certain level before the filter width even begins to have an impact on the solution. This is a completely different type of convergence than in the larger cases, $\delta_f^* > 4$, where the filter had an impact on the solution at all grids.

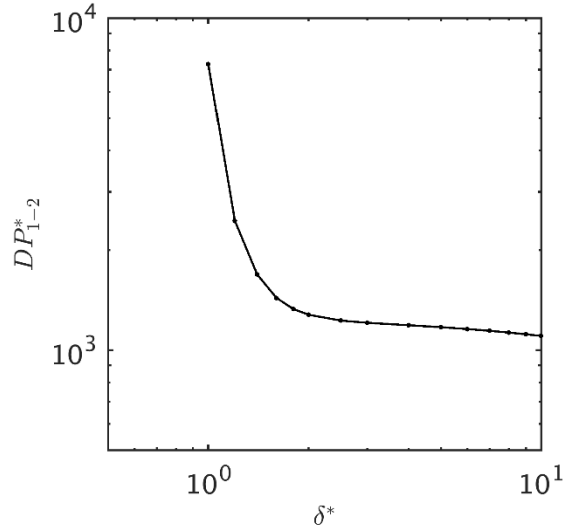


Figure 11: Variation in bed pressure drop with filter width, δ^* , on a numerical grid of $N_z = 40$ ($\Delta^* = 0.25$).

It can also be observed from Figure 9 and Figure 10 that the extrapolated exact solution(s) are increasing with decreasing filter width. To study this trend further, the grid resolution of $N_z = 40$ ($\Delta^* = 1/4$) is used to simulate filter width varying from $\delta_f^* = 1$ to 12. The results are summarized in Figure 11. For a wide range of δ_f^* the pressure drop remains relatively stable, i.e., it appears that the choice of δ_f^* does not strongly affect the solution of DP_{1-2}^* . However, this trend changes abruptly around $\delta_f^* = 3$, below which the SRQoI is strongly dependent on the modeling choice in δ_f^* (note the log-log scale).

One concern is that perhaps the diffusion scheme is failing under such small filter widths and the particle volume fraction is not being conserved. The left panel of Figure 12 shows that this is not the case. The mean concentration in all cases agrees very well with the expected value (calculated from the integer number of particles in the domain) and, if anything, only agrees better at small filter widths. Perhaps as some readers have already guessed, the sharp increase in DP_{1-2}^* with δ_f^* of around two and below is really the essence of the validation question, i.e., the filter width should not be this small. Rather than problems with the total concentration, it is the local concentration that is the issue. The right-hand side of Figure 12 shows that the largest concentration calculated in any cell increases with decreasing δ_f^* . The smaller values of δ_f^* have exceeded the dense bed concentration of 0.30, even what might reasonably be expected to be the densest region from the random packing configuration. Actually, for the smallest values of δ_f^* studied, the maximum concentration exceeds the physical limit for monodisperse spheres in random close packing, ~ 0.62 .

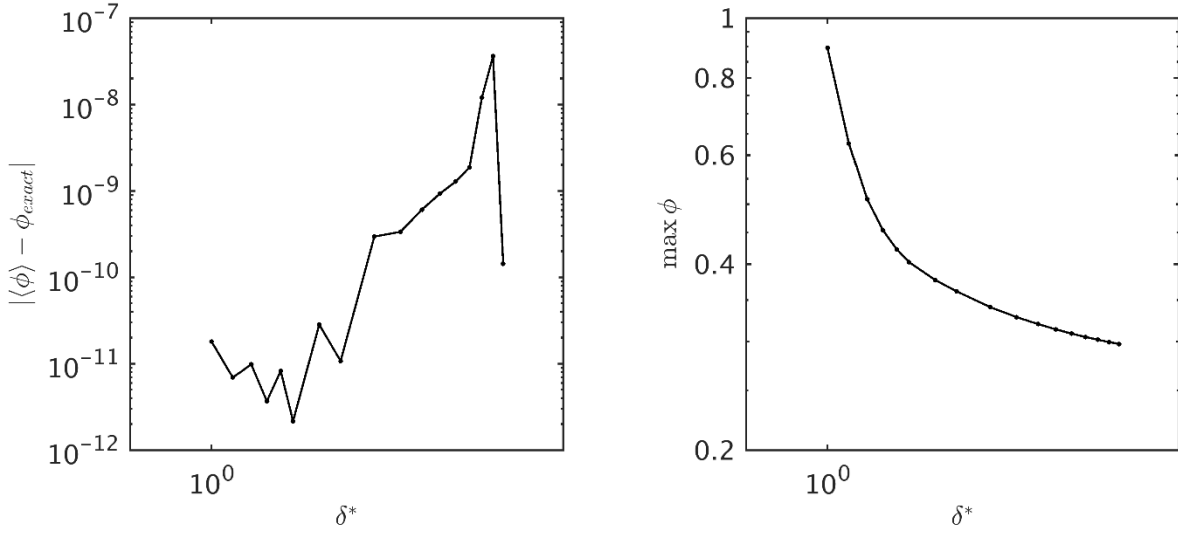


Figure 12: At left, absolute difference between mean calculated concentration and exact, analytical value. At right, maximum calculated solids concentration within the domain. Both shown as a function of the filter width, δ^* , on a numerical grid of $N_z = 40$ ($\Delta^* = 0.25$).

The concentration and flow profiles along the center $z^* = L_z^*/2$ plane are provided in Figure 13 for several filter widths. For $\delta_f^* = 1$ (left-hand side of Figure 13), individual particles have been “fuzzily” resolved which results in microscale flows that are very low within particles and very large in the interstices with very low concentration. While this may be somewhat physically reminiscent of DNS flow patterns, this is clearly not a condition that the mean drag law (constituted in a domain of several/many particles) was derived to model. Note that on the other end, beyond $\delta_f^* > 12$, the maximum concentration is just barely reaching 0.30 of the dense bed. Further increase would begin to cause excessive diffusion until all identifiable patterns of the problem have been smeared out into uniformity. Even for $\delta_f^* = 12$ the bubble structure on the right-hand side of Figure 13 has disappeared. While a validation statement cannot be made on what the filter width *should be* without validation data, clearly it seems to need to fall within the range of $\delta_f^* \sim [2, 12]$. This range is consistent with other works which utilize the diffusion filtering approach. Capecelatro and Desjardins (2013) use δ_f^* of 3, 3.5 and a range of 2 to 8 for a poly disperse case. Using a locally adaptive filter, Capecelatro et al. (2015) specify δ_f^* which varies from 2.5 to 12.6 for concentrations ranging from 0.6 to 0.005, respectively.

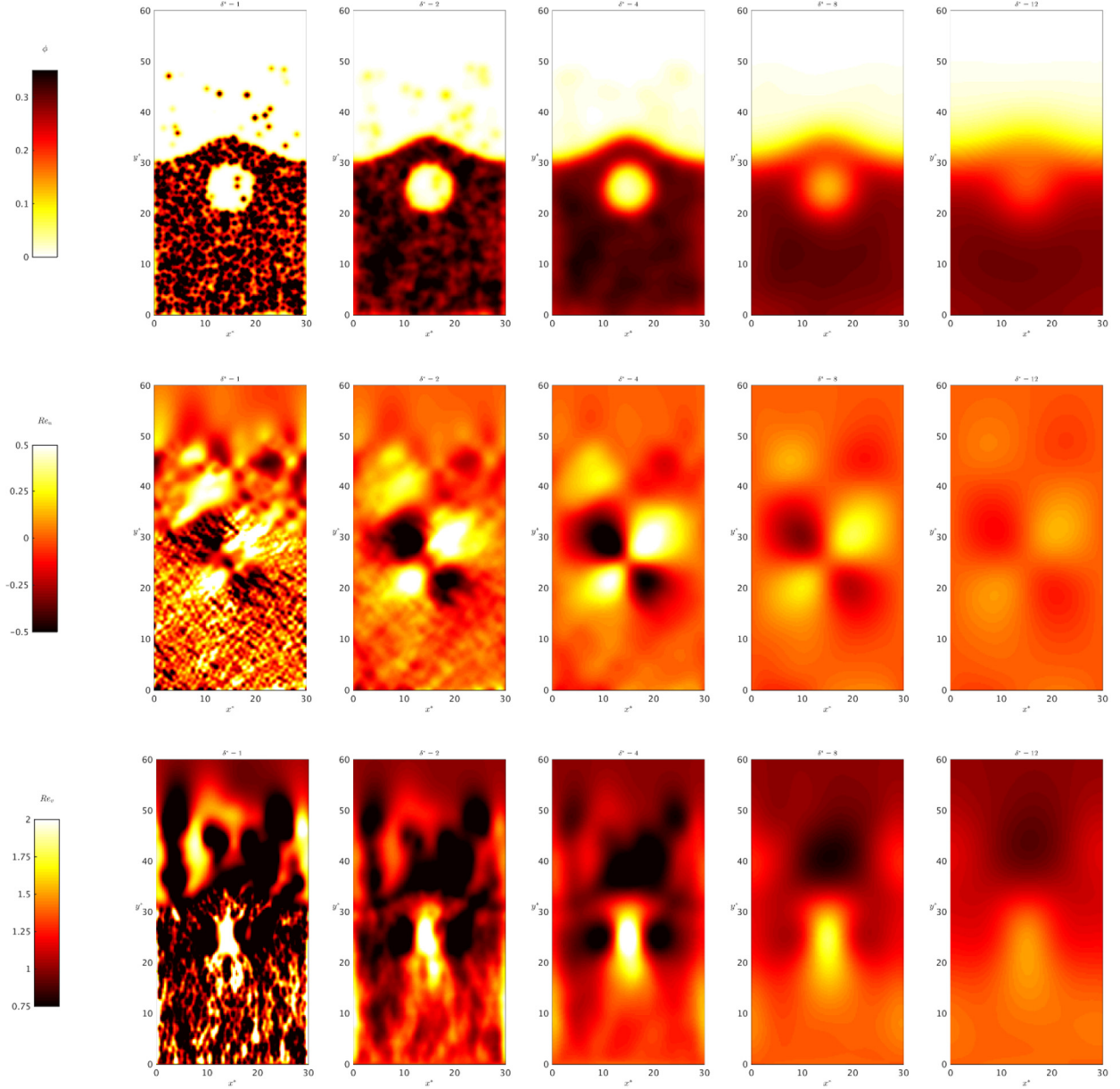


Figure 13: Convergence of the solids volume fraction field (top), gas horizontal (x) velocity field, $Re_u = \rho_p d_p u_g / \mu_g$ (center), and gas vertical (v) velocity field, $Re_v = \rho_p d_p v_g / \mu_g$ (bottom) at the center of the domain, $z^* = L_z^*/2$, with increasing grid refinement.

As a side note, it was found that for convergence behavior such as that observed in Figure 10, a better estimate of the approximate exact solution apparently can be found from the single power law extrapolation with a severe inverse grid size weighting. Figure 14 shows the fit to single power law extrapolations with $W_i = (h_i)^{-8}$ weighting factors. The coarse grid solutions have been essentially neglected, although not entirely. Hence the extrapolated curve does not predict the coarse grid solution at all (if this curve were used for solution verification these solutions would have a fictitiously high error), but provides a better extrapolation of the high-resolution trends.

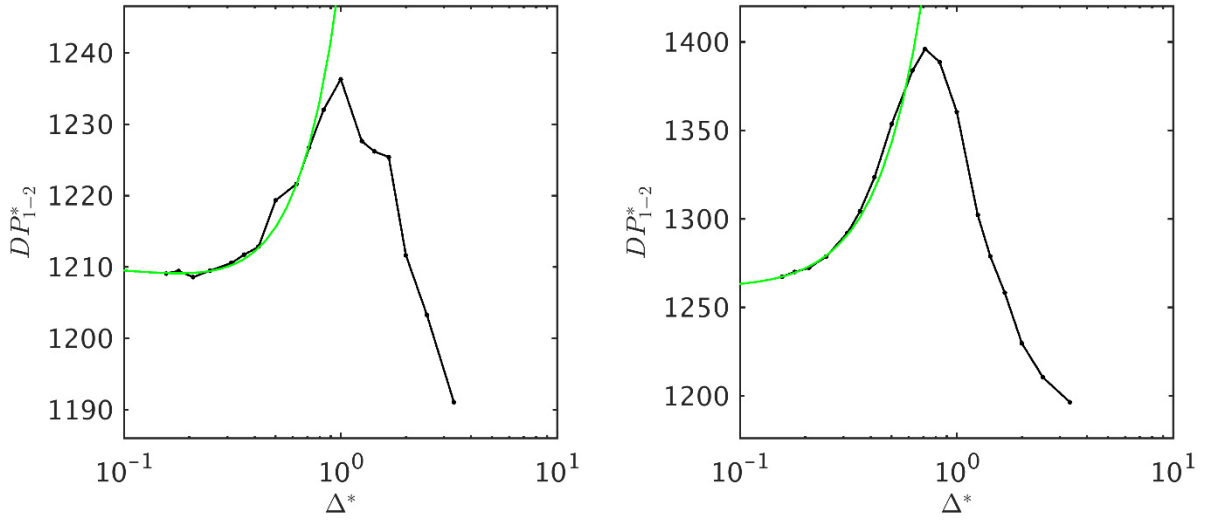


Figure 14: Same data as in Figure 10 here using least-squares power law extrapolation with $W_i = h_i^{-8}$.

3.6 GARG_2012 INTERPOLATION

Although this work focuses on grid-independent filtering using the diffusion method, a few grid refinement cases were also compared using the GARG_2012 interpolation method. In this case only a subset of the grids outlined in Table 1 can be studied and the focus of this study was restricted to $N_z > 10$ corresponding to $\Delta^* \geq 1$. The results are presented in Figure 15. Neither the power law nor the mixed order extrapolation schemes give a particularly convincing fit to the data. Using only the finest two or three solutions to perform an extrapolation would provide an even more ambiguous result as they are divergent. It is hard to tell from the data available in Figure 15 what the value of DP_{1-2}^* should be in the limit of infinite resolution, which is part of the challenge of placing grid/resolution restrictions on solution verification studies.

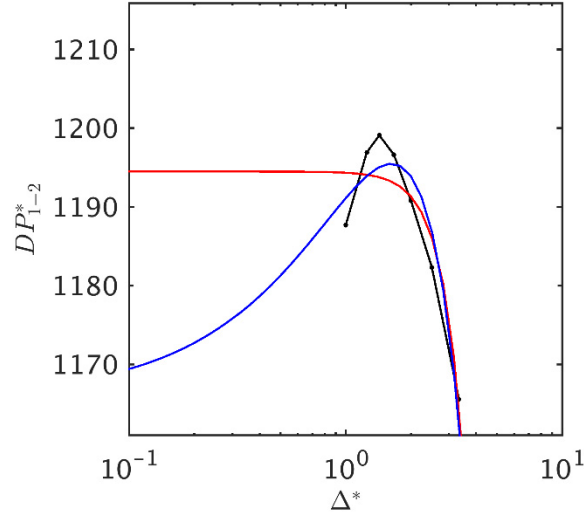


Figure 15: Convergence behavior of the bed pressure drop in the bursting bubble case with the GARG_2012 interpolation scheme. Red least-squares curves are power law extrapolation and blue curves are mixed-order.

It should also be noted that using the grid-coupled GARG_2012 interpolation scheme is not as pure of a solution verification study as with the grid-independent filtering method. Instead, two things are occurring at once, which was mentioned in Section 1. Since the GARG_2012 scheme is coupled to the grid, when the grid is refined the CFD-DEM model itself is also changing, via the filtering method, specifically the length scale over which particle data is transferred. While solution verification may be significantly complicated in this case, the results are similar to those in Figure 11 with the variable filter width (single grid). The comparison here is a little ambiguous, but depending on where a particle is within a given grid cell, the GARG_2012 interpolation will give a weighted filtering between two and three cells. Therefore, in Figure 16 compared δ_f^* from the diffusion filtering method to $2\Delta^*$ to $3\Delta^*$ (horizontal red lines).

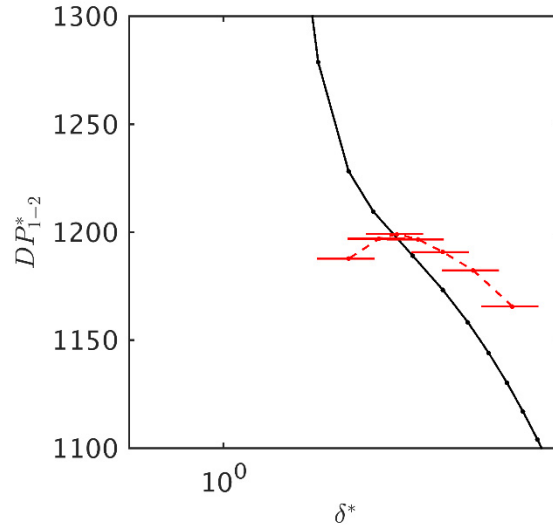


Figure 16: Data of Figure 15 compared to data in Figure 11, i.e., variation in bed pressure drop calculated with Garg_2012 interpolation on different numerical grids (red) compared to diffusion interpolation with different filter widths, δ^* , on a single numerical grid of $N_z = 40$ ($\Delta^* = 0.25$) (black). The horizontal lines span from $2\Delta^*$ to $3\Delta^*$ with the connecting curve at $2.5\Delta^*$.

3.7 DIFFERENT PARTICLE CONFIGURATIONS

This section explores two different particle configurations. Since the particles in this case are generated randomly, the (pseudo-) random number generator can be seeded with a different starting point to get a different configuration which is statistically equivalent. The particle configuration (PC) used previously will now be referred to as PC1 to differentiate it from two new randomized replicates PC2 and PC3. Particles near the center-plane for the three PCs are compared in Figure 17. Clearly the general problem remains the same (statistically equivalent based on geometry/concentration of the problem), but the exact particle locations have changed. As a slightly different variant, a fixed but non-static assembly for PC3 was also explored. In this case, labeled PC3t, the particles are given a uniform (Gaussian) random distribution of velocities with zero mean. Since the particles have no net motion, this condition was considered a “thermal” fixed-bed. The magnitude of the particle velocities can be quantified as a pseudo-granular-temperature, $T = \sum \mathbf{v}_p \cdot \mathbf{v}_p / 3N_p$. The thermal Reynolds number, $Re_T = \rho_p d_p T^{1/2} / \mu_g$, is set to 0.025. The thermal condition of PC3, PC3t, is shown in Figure 17 with individual particle velocity magnitudes normalized by the inflow gas-phase velocity, $\|\mathbf{v}_p\|/U_0$.

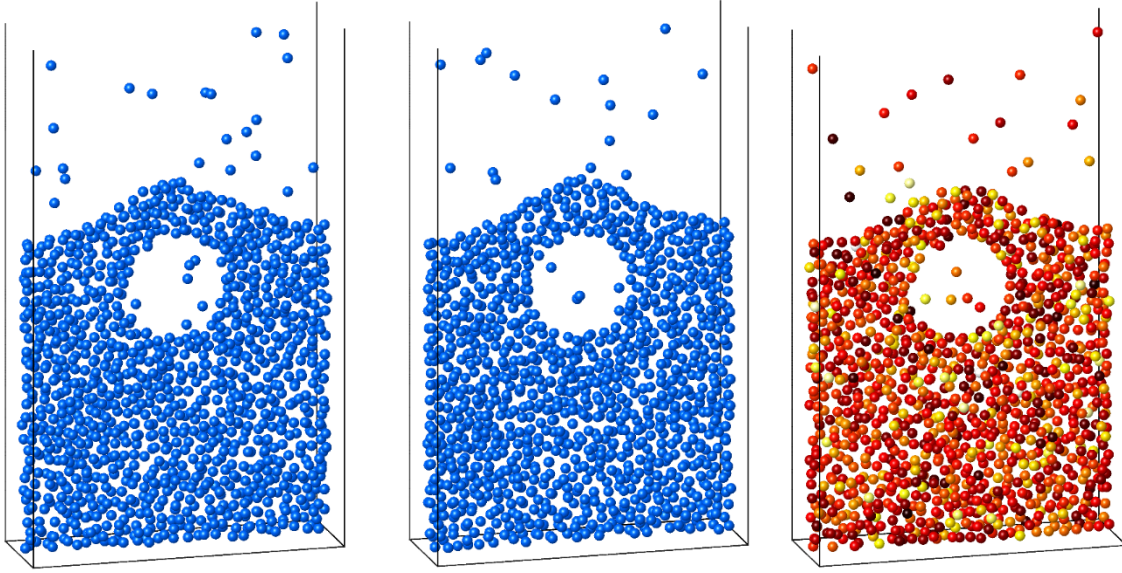


Figure 17: Initial particle configuration of original particle configuration (PC1) on left, a second replicate PC2 in center and a third replicate PC3 at right, which is also considered in a thermal condition (as shown), PC3t. For PC3t colors indicate particle velocity magnitude normalized by U_0 varying from 0 (black) to 0.1 (white). All particles shown have centroids within d_p of the centerline, $z=L_z/2$.

Similar to Figure 6 for the baseline PC1 case, the convergence of the SRQoI is shown in Figure 18 for PC2 (left) and the static and thermal PC3 (right). The overall trends are generally the same as in the baseline case. For PC2 and PC3, the convergence rates are quite a bit higher, $p = 2.28$ and 2.60 , respectively. On the whole, there is very little difference between PC3 and PC3t. This may not be so surprising since the pressure drop is largely determined by the dense region. Since there are several particles within a given diffusion filter length and velocities have a zero mean, much of the individual particle velocities have been smeared out onto an even lower solids velocity field. It is possible to circumvent this by providing a correlated particle motion to see how mean solids (filtered) velocities affect the results, but more realist PCs will be explored with correlated motion in Section 4, which will move away from the fictional cases produced with a random particle generator. It may have been expected that providing a particle velocity would increase the pressure drop yet, $DP_{1-2}^*(PC3t) < DP_{1-2}^*(PC3)$. Since the particle velocities are thermal (random, zero mean), this simply means that, on average, more particles have randomly been aligned with the gas-phase velocity rather than oppose it.

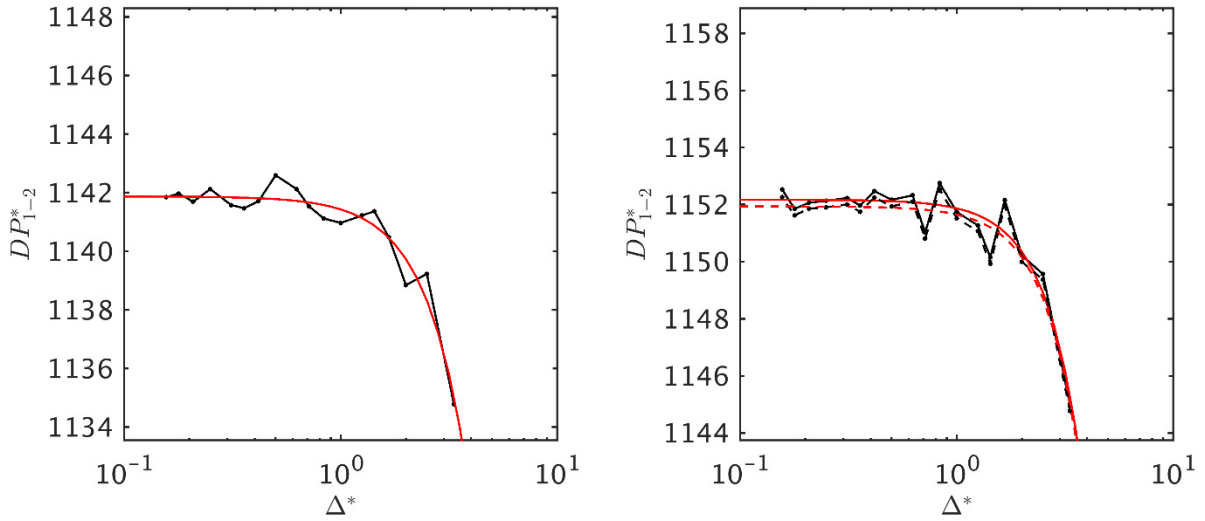


Figure 18: Convergence behavior of the bed pressure drop in PC2 (left) and PC3 (right) showing least-squares power law extrapolations (red). On the right, the dashed line indicates the thermal case, PC3t.

While the general trends of the SRQoI convergence behavior are quite similar for the different configurations, the magnitudes are slightly different. In Figure 19, DP_{1-2}^* for the three (static) PCs are directly compared. It is observed that the variation between the configurations is comparable to the variation between the (approximate) exact and coarsest solution any one PC. The differences are not surprising given that these are not identical PCs. However, they are statistically equivalent and the magnitude of their relative variation to convergence on one replicate is meaningful. Consider the question, “*What is the PC-independent pressure drop for this bed geometry?*”, i.e., the ensemble average over all [many] replicate PCs. Unless many replicates are used, the error due to the ensemble averaging itself will be as large as the numerical error of the grid. (However, using the coarsest grids would bias the result). In a broader sense, CFD-DEM simulations are transient and data is typically taken as time average, i.e., the average is taken over many different reference states, which are oftentimes very different, see Section 4. If the error introduced from two statistically equivalent cases matches the spatial discretization error, this is an indication that, for more practical conditions which are highly dynamic and only time averaged over computationally tractable periods of time, the contribution to the error in the measurements from the numerical grid—even for relatively coarse grids—is likely much less than the error (or uncertainty) of (incomplete) time averaging.

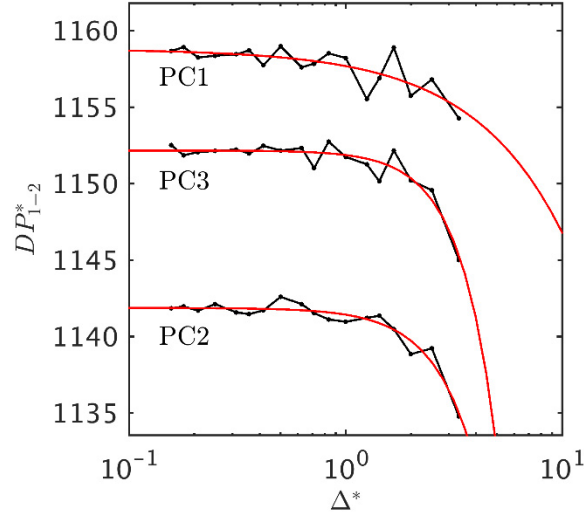


Figure 19: Comparison of convergence behavior of the bed pressure drop in PC1, PC2 and PC3 and their respective least-squares power law extrapolations (red).

3.8 DISCRETIZATION SCHEMES ON PC2

Here, the robustness of the results presented in Section 3.4 are tested on PC2 using FOU and MUSCL discretization schemes. Skipping first to the MUSCL scheme, which is straightforward, the right panel of Figure 20 shows a similar convergence behavior to most previous results. The order of convergence is approximated at $p = 3.20$. In the case of the FOU scheme, the power law extrapolation failed. The numerical method could not find a root of the nonlinear equations between a small number and order 20, indicating that the order of convergence was zero. This occurs due to the globally nonmonotonic behavior. As before, a mixed-order (first-second) extrapolation scheme produces a nice fit to this type of data. Note that it is primarily the coarsest grid, $i = n_g$, causing this behavior. If either this point is neglected, $W_{i=ng} = 0$, or if an inverse squared weighting is used, $W_i = 1/h_i^2$, then the power law extrapolation succeeds with predicted convergence rates of $p = 0.19$ and 0.21 , respectively. These two curves are shown in Figure 20 as the solid and dashed red lines. Although the convergence rates are again quite different than the expected order of accuracy of the numerical scheme (approximately between one and two), $p^{(\text{FOU})} < p^{(\text{Superbee})} < p^{(\text{MUSCL})}$, are still found as before with PC1.

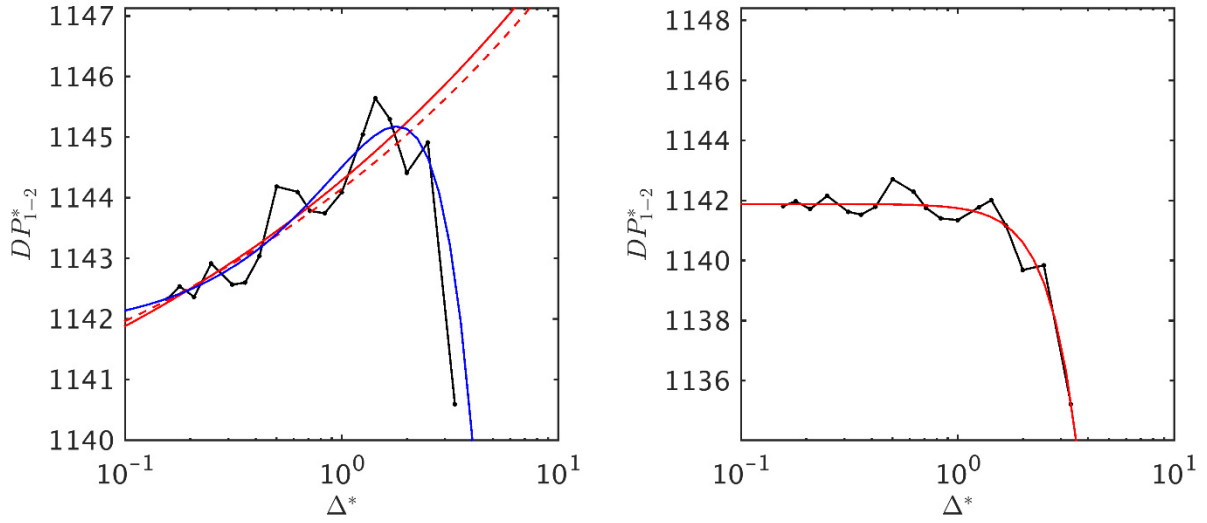


Figure 20: Convergence behavior of the bed pressure drop in the bursting bubble case with the FOU scheme (left) and the MUSCL scheme (right). Both compare to a least-squares power law extrapolation (red) with the coarsest grid neglected in the FOU case. Also shown for FOU are power law fit with inverse fitted to FOU data, the coarsest grid has been neglected law with $W_i = h_i^{-2}$ (dashed) weighting and the mixed-order method (blue).

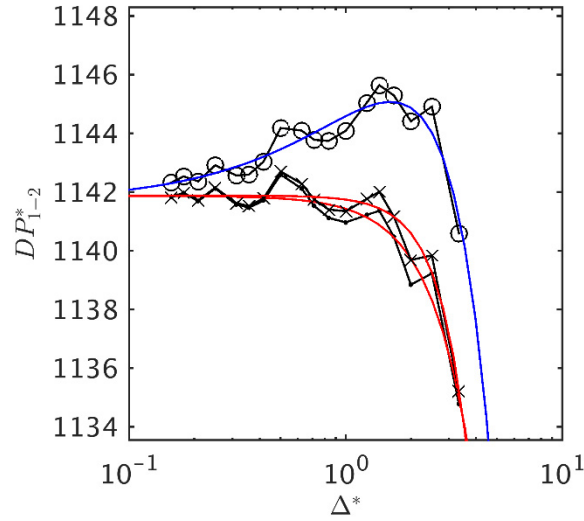


Figure 21: Direct comparison of results in Figure 18 (left) and Figure 20 Superbee ($\square$), FOU ($\circ$), and MUSCL ($\times$) discretization schemes on replicate PC2.

The convergence behavior of the three schemes are compared in Figure 21. The two higher-order methods converge to each other in this resolution range. The FOU scheme is still in the process of converging to the higher order results, but difference is quite small at the highest resolutions considered here. Overall, the discrepancy due to the numerical method seems to be rather small, even the FOU scheme which may be partially attributable to the larger filter size ($\delta_f^* = 6$) smoothing out the discontinuities of the bubble and bed surface.

3.9 DRAG LAWS ON PC3t

Before moving onto a new fixed bed problem, the thermal fictional bursting bubble case, PC3t, was used to study the impact of the drag law. Again, the focus here is on verification, specifically solution verification and cannot make any assertions about which drag law is the most appropriate. This is a validation question and one that has been at the heart of countless studies over several decades with no universally accepted answer. Here, it is determined how the drag law affects convergence behavior.

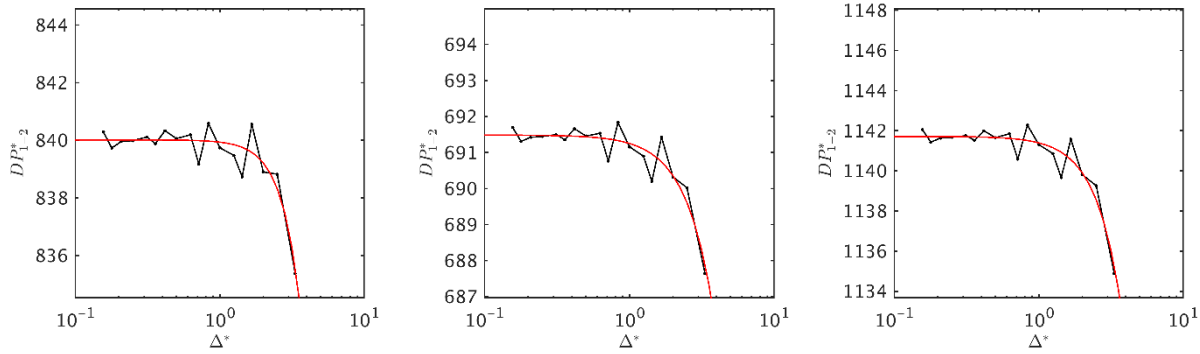


Figure 22: Convergence behavior of the bed pressure drop in PC3t using the GIDASPOW_BLEND (left), WEN_YU (center), and KOCH_HILL (right) drag laws showing least-squares power law extrapolations (red).

All results reported thus far have followed the baseline configuration and employed the BVK (Beetstra et al., 2007) drag law. On PC3t, the drag laws (MFiX nomenclature) are applied: GIDASPOW_BLEND (Ding and Gidaspow, 1990; Lathouwers and Bellan, 2001), WEN_YU (Wen and Yu, 1966); and KOCH_HILL (Hill et al., 2001a; Hill et al., 2001b; Benyahia et al., 2006). The convergence of the SRQoI for the three drag laws is almost identical, as shown in Figure 22. The extrapolated convergence rates are also quite similar, $p = 3.38, 2.27$, and 2.49 . The only major differences in the data or the extrapolated curves are the approximate exact solutions. Without making an assessment of the validity of any model (other than noting that all of these models have been applied extensively in a large number of previous works), it should be noted that the difference between models is large compared to the spread of the results in the grid convergence of any particular model. The large absolute difference between an extrapolated exact value and the coarsest solution occurs for the baseline case which is $\left| f_0^{(BVK)} - f_{i=ng}^{(BVK)} \right| = 7.16$. On the other hand, the difference between the extrapolated exact solutions with BVK and WEN_YU drag laws is $\left| f_0^{(BVK)} - f_0^{(WEN_YU)} \right| = 467$, over **65x larger**. Within the VVUQ

framework, this would indicate that model form uncertainty (and potentially input uncertainties that drag models depend on) dwarf numerical uncertainties, a welcome result given that a more comprehensive VVUQ study of a gas-solids system (although utilizing the TFM) had previously indicated that numerical uncertainty may be excessively large (Gel et al., 2013). Figure 23 compares the results of the four different drag laws against one another. At this scale, the variation due to the grid has almost vanished and the simulation results collapse to nearly horizontal lines. Only the tails of the extrapolations vary appreciably, and at a resolution not frequently utilized in CFD-DEM. Although caution should be exercised in generalizing the results from a small, fictitious, fixed particle bed to more general, practical CFD-DEM simulations, these are encouraging findings.

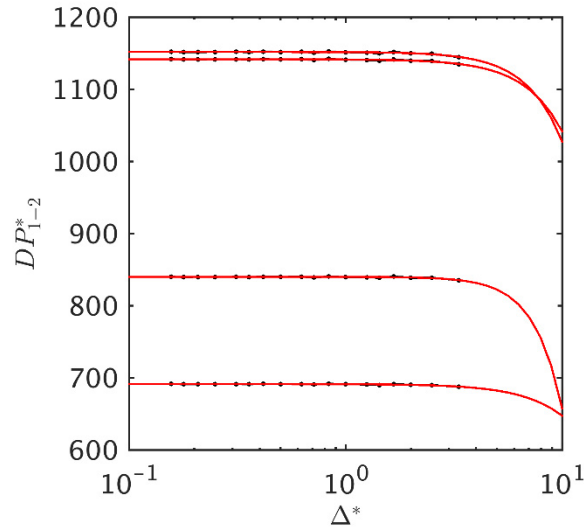


Figure 23: Convergence behavior of the bed pressure drop in PC2 (left) and PC3 (right) showing least-squares power law extrapolations (red).

3.10 SUMMARY OF EXTRAPOLATED DATA

Throughout Sections 3.3–3.9, a least-squares method was used, outlined by Eça and Hoestra (2014) to compute the minima of Eqs. 8–9 producing power law and mixed-order curve fits to the solution data. The curve fits extrapolate the data to the limit of infinite resolution which provide estimates of the grid-free solutions. Some of the coefficients of the curve fits have been provided where discussed, primarily the extrapolated order of convergence, p . In Table 2 and Table 3, all three coefficients of both types of regression are summarized for all cases discussed in Sections 3.3–3.9.

Table 2: Summary of coefficients from least-squares extrapolation using power-law regression

case	f_0	p	c
baseline, $W_i = 1/h_i$	1.158800E+03	1.0326910	-1.112676E+00
baseline, $W_i = 1$	1.158618E+03	1.2877890	-8.354265E-01
FOU	1.154426E+03	0.1384588	5.964083E+00
MUSCL	1.158627E+03	1.5037070	-5.375400E-01
$\delta_f^* = 2$	1.298589E+03	3.3736310	-2.233332E+00
$\delta_f^* = 3$	1.213669E+03	6.3961730	-1.065858E-02
$\delta_f^* = 4$	1.190161E+03	9.1643680	-1.461461E-04
$\delta_f^* = 8$	1.131035E+03	0.7939152	-1.656922E+00
GARG_2012	1.194496E+03	4.3984070	-1.492723E-01
PC2	1.141871E+03	2.2775130	-4.406736E-01
PC3	1.152168E+03	2.5982400	-2.996187E-01
PC3t	1.151936E+03	2.6380250	-2.861210E-01
PC2, FOU	-	-	-
PC2, FOU, neglecting $N_z = 3$	1.137526E+03	0.1910382	6.762096E+00
PC2, FOU, $W_i = 1/h_i^2$	1.138479E+03	0.2111117	5.669678E+00
PC2, MUSCL	1.141883E+03	3.1991670	-1.397391E-01
PC3t, GIDASPOW_BLEND	8.400106E+02	3.3755370	-7.677591E-02
PC3t, WEN_YU	6.914840E+02	2.2747090	-2.330144E-01
PC3t, KOCH_HILL	1.141717E+03	2.4903680	-3.228197E-01

Table 3: Summary of coefficients from least-squares extrapolation using mixed-order regression

case	f_0	c_1	c_2
baseline, $W_i = 1/h_i$	1.158793E+03	-1.036744E+00	-4.615858E-02
baseline, $W_i = 1$	1.158673E+03	-7.530477E-01	-1.386442E-01
FOU	1.158630E+03	2.800858E+00	-7.556195E-01
MUSCL	1.158674E+03	-4.114950E-01	-1.815690E-01
$\delta_f^* = 2$	1.254384E+03	1.534158E+02	-6.203141E+01
$\delta_f^* = 3$	1.202255E+03	3.795341E+01	-1.349758E+01
$\delta_f^* = 4$	1.187415E+03	9.148216E+00	-3.309191E+00
$\delta_f^* = 8$	1.130912E+03	-1.720572E+00	1.587567E-01
GARG_2012	1.165907E+03	3.643732E+01	-1.121604E+01
PC2	1.141820E+03	3.117257E-01	-6.945055E-01
PC3	1.152106E+03	4.411637E-01	-7.093355E-01
PC3t	1.151864E+03	4.823467E-01	-7.213819E-01
PC2, FOU	1.141671E+03	4.233970E+00	-1.316126E+00
PC2, FOU, neglecting $N_z = 3$	1.141770E+03	3.791499E+00	-1.054393E+00
PC2, FOU, $W_i = 1/h_i^2$	1.141748E+03	3.946489E+00	-1.197082E+00
PC2, MUSCL	1.141677E+03	9.988265E-01	-8.381551E-01
PC3t, GIDASPOW_BLEND	8.399329E+02	4.815103E-01	-4.961429E-01
PC3t, WEN_YU	6.914782E+02	9.119092E-02	-3.402114E-01
PC3t, KOCH_HILL	1.141675E+03	3.330353E-01	-6.506675E-01

4. SSCPI

4.1 SYSTEM AND CONDITIONS

The results presented in Section 3 were quite promising and showed rather definitively that solution verification via grid refinement is applicable to the CFD-DEM method, at least with a grid-independent filtering scheme. One remaining concern is with the relatively simple and completely fictitious bursting bubble problem. *Do the previous results hold if you freeze an actual CFD-DEM simulation so that the particles are in a realistic (but frozen) configuration?* In an effort to address this question, the particle configurations were taken from an actual simulation of the NETL Small Scale Challenge Problem (SSCP-I), which was run previously (Gopalan et al., 2016).

The SSCP-I is a rectangular bubbling bed of Geldart Group D particles. The bed is 23.0 cm wide, 122.0 cm tall, and 7.5 cm deep. The median particle diameter, d_{p50} , is 3256 μm . The particles have a fairly narrow size distribution and a single particle diameter (monodisperse) is applied, $d_p = d_{p50}$. The nondimensionalized domain is therefore $L_x^* = 70.64$, $L_y^* = 374.69$, and $L_z^* = 23.03$, roughly two times larger in each dimension than the previous fictional case. However, the mass inventory in this case is such that the particle count is 92,948, almost a 20-fold increase over the fictional case. The fluid properties are set as $\rho_g = 1.2 \times 10^{-3} \text{ g/cm}^3$, $\mu_g = 1.8 \times 10^{-4} \text{ g/cm-s}$. Although not significant for the resulting fixed bed simulations, the particle properties may affect the states observed (i.e., frozen for study here) and they are: $\rho_p = 1.131 \text{ g/cm}^3$, $e_{pp} = 0.84$, $e_{pw} = 0.92$, $\mu_{pw} = \mu_{pw} = 0.35$. The particle-particle and particle-wall spring stiffness coefficients of the linear spring dashpot model are both 10^6 . One minor discrepancy should be noted. The particles are slightly nonspherical, sphericity of 94%. The sphericity was taken into account during the original simulation used to produce the PCs, but not here in the fixed bed CFD-grid convergence studies for consistency with the results presented in Section 3. The original simulation used the WEN_YU drag law (Wen and Yu, 1966) and the Superbee flux limiter. However, the interpolation method of GARG_2012 was used. Similar to the fictional case, a uniform mass inflow BC is set at the inlet plane, $y^* = 0$ to model a distributor. The inlet gas velocity is $U_0 = 219 \text{ cm/s}$, hence $Re_{p0} = \rho_g d_p U_0 / \mu_g = 475$. A pressure outflow BC is specified at the exit plane, $y^* = L_y^*$. The front, back, left side, and right-side domain boundaries are all specified as no-slip wall BCs.

The original simulation used to generate the particle configuration used a grid of $N_x \times N_y \times N_z = 18 \times 96 \times 6$; where there was a grid number spacing of $N_x = 3N_z$ and $N_y = 16 \times N_z$. However, this grid (and all others used here) is not uniform (cubic), but close, with $\Delta_x^* = 3.92$, $\Delta_y^* = 3.90$, and $\Delta_z^* = 3.84$ for $\Delta^* = 3.89$. In general, $\Delta^* = \sqrt[3]{(V_{tot}^*/48)/N_z} \approx 23.332/N_z$. For the fixed-bed convergence tests, grids of $N_z = 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32$, and 40, are considered, each with $N_x = 3N_z$ and $N_y = 16 \times N_z$. This spans a grid cell size range from $\Delta^* = 4.67$ to 0.58.

In the fixed-bed convergence tests used here, only the particle positions and velocities, PCs, are used for the fixed bed simulations using the same code modifications as in the fictional case discussed in Section 2. The fluid domain is reinitialized as single phase, hydrostatic pressure profile and $u_g(t=0) = 0$, $v_g(t=0) = 200 \text{ cm/s}$, and $w_g(t=0) = 0$. The diffusion filtering scheme is applied with $\delta_f^* = 6$. The BVK drag law (Beetstra et al., 2007) is specified in the baseline case and the GIDASPOW_BLEND (Ding and Gidaspow, 1990; Lathouwers and Bellan, 2001) model is also used for comparison. The PCs are taken from the “DES_DATA” code output files which were saved at a frequency of 1 Hz. The original simulation was run for a period of 65 s, the first 5 s was

neglected and the remaining 60 s used for time averaging statistical properties of the system. Here, the first six cases outside of the transient window were considered and saved at: $t = 5, 6, 7, 8, 9$ and 10s which are denoted PC5, PC6, etc.

4.2 CONVERGENCE RESULTS

The bed pressure drop of Equation 10 is again considered as the SRQoI. Here, however, since this is a real problem there is a more natural choice for the locations y_1^* and y_2^* . Although the experimental data will not be compared, the original simulation did and it seems a natural choice to use the experimental pressure tap locations here as well. Therefore, $y_1 = 4.13$ cm and $y_2 = 34.61$ cm, or $y_1^* \cong 12.68$ and $y_2^* = 106.3$, were used in calculating DP_{1-2}^* .

A slice of particles lying along the center-plane of the configurations PC5 to PC10 are provided in Figure 24 to Figure 29. The particles are colored by their velocity magnitude normalized by the inlet gas velocity. Unlike the thermal PC3t of the fictitious case, these snapshots of the simulation “frozen” in time show mean, correlated motion of the particles. The conditions of PC5–PC10 are clearly more realistic than the fictional ones created with a random particle generator. Perhaps one of the most significant differences is that the simple random generator did not allow for particles to be in contact to avoid excessive overlap. In these cases, the dense regions are much denser than just 30% (near the maximum concentration limit for a random generator to continue seeding particles without contact) and some particles (likely many particles) are in contact. However, some elements that the bursting bubble geometry attempted to represent can be identified in PC5–PC10 can be identified.

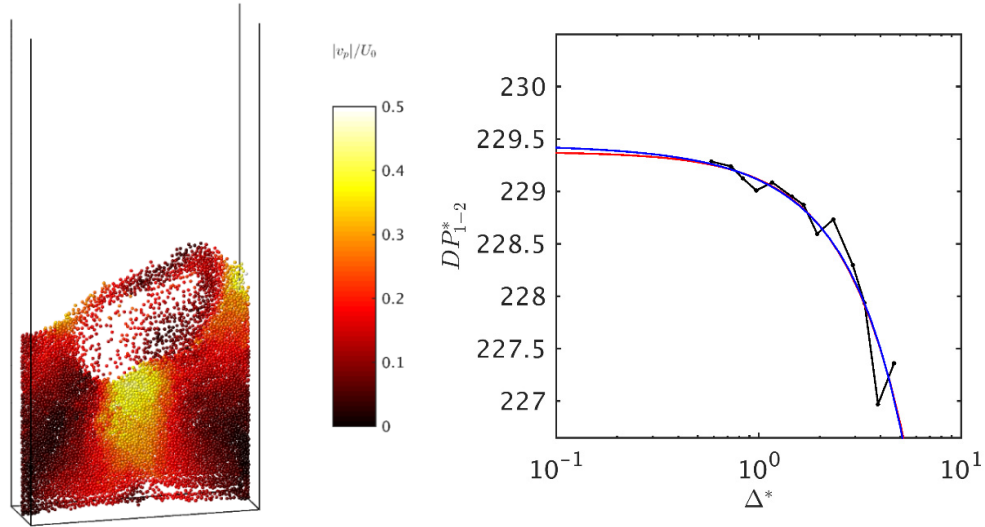


Figure 24: On left, slice of PC5, particle configuration of SSCP-I simulation at time $t = 5$ s, showing particles with centroids within d_p of the centerline, $z=L_z/2$, colored by normalized velocity magnitude. At right, convergence behavior of the bed pressure drop compared to least-squares extrapolations of power law (red) and mixed-order (blue) regression types.

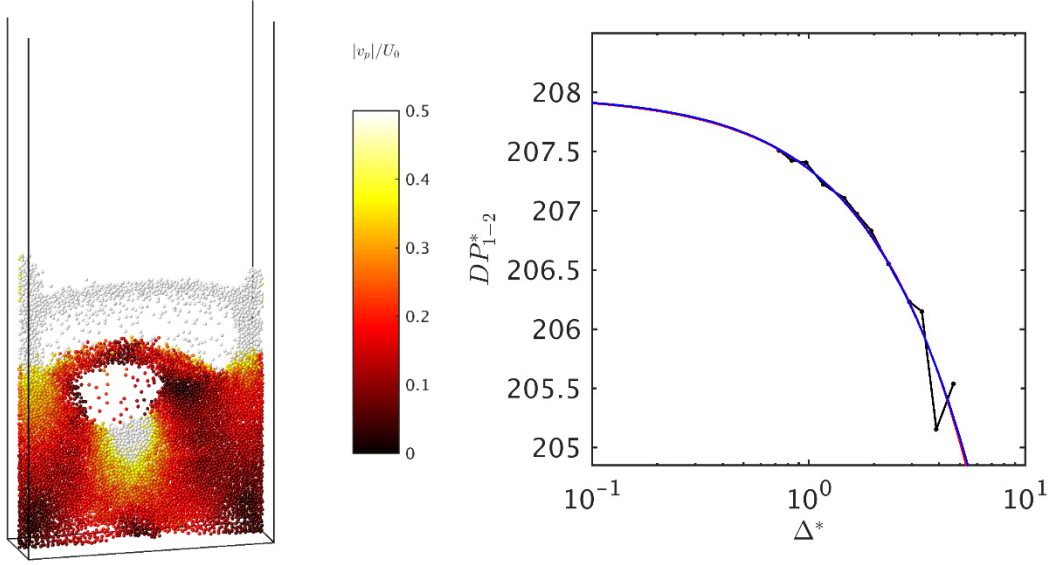


Figure 25: Same as Figure 24 frozen at $t = 6$ s, i.e., PC6.

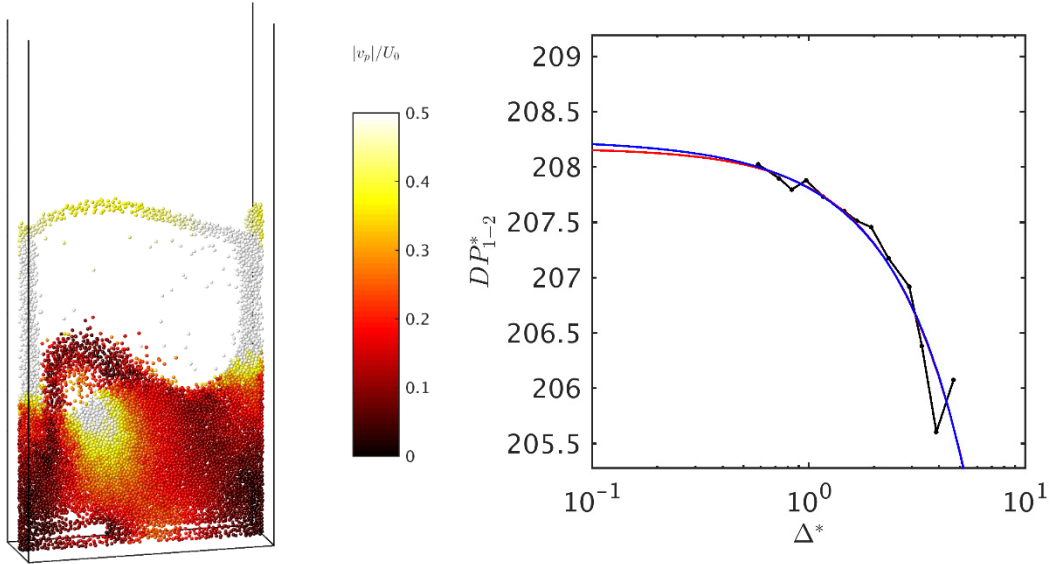


Figure 26: Same as Figure 24 frozen at $t = 7$ s, i.e., PC7.

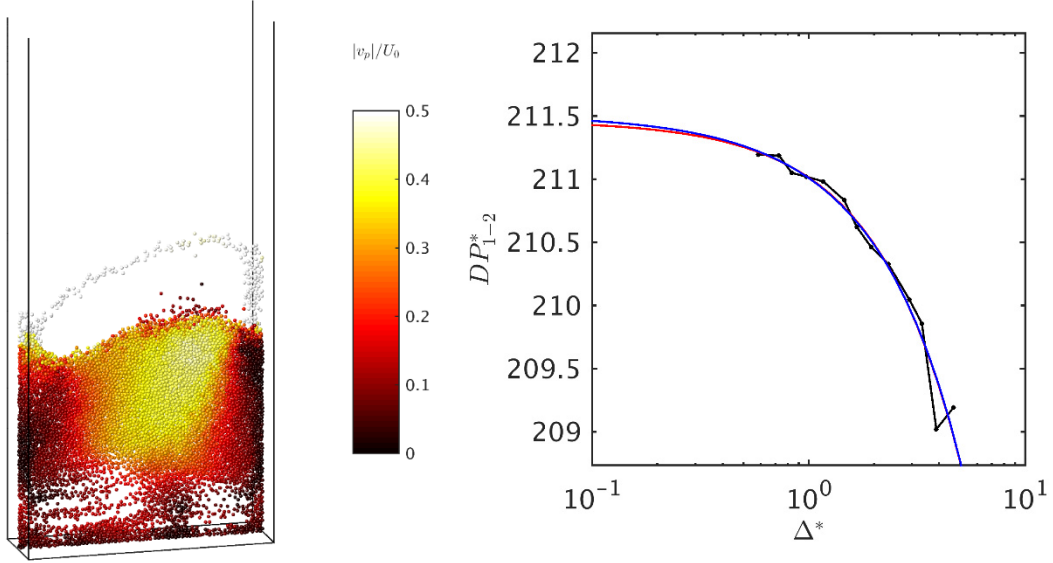


Figure 27: Same as Figure 24 frozen at $t = 8$ s, i.e., PC8.

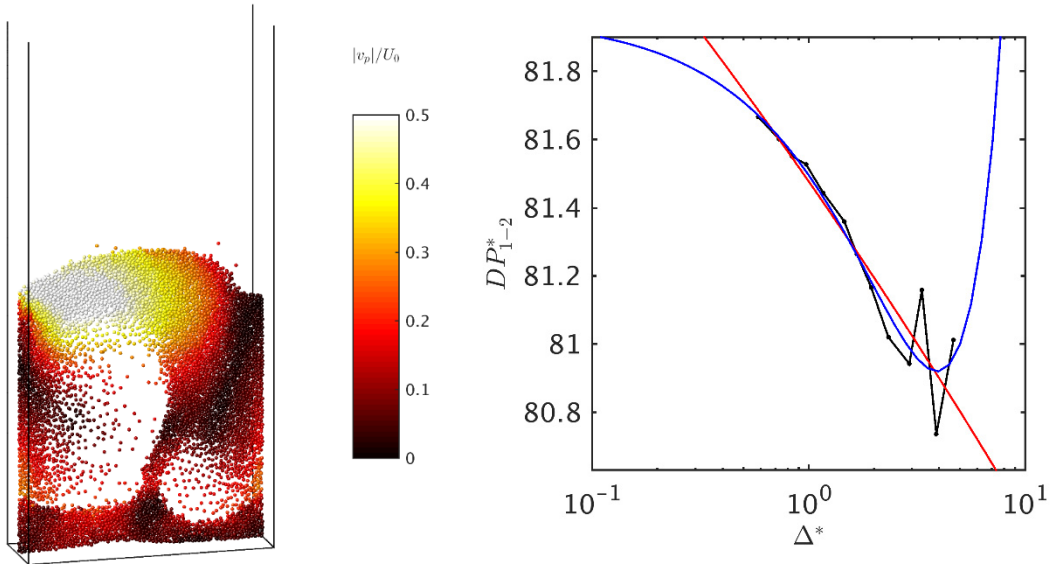


Figure 28: Same as Figure 24 frozen at $t = 9$ s, i.e., PC9.

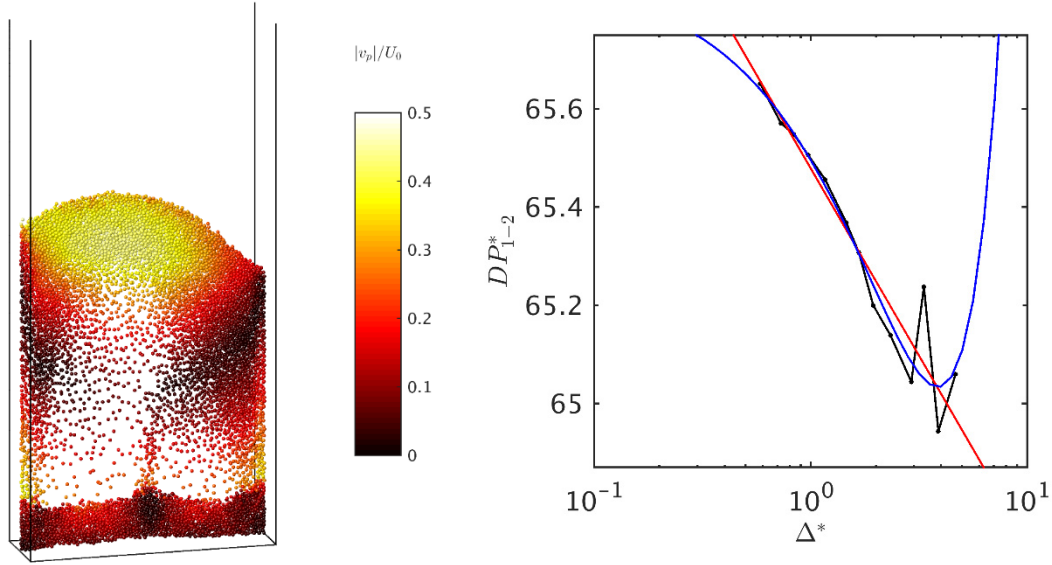


Figure 29: Same as Figure 24 frozen at $t = 10$ s, i.e., PC10.

The convergence behavior for PC5 to PC10 are shown on the right-hand side in Figure 24 to Figure 29. In all instances, the BVK drag law has been used and both power-law (red) and mixed-order (blue) least-squares extrapolation curve fits have been compared. It is rather apparent that the convergence behavior is somewhat degraded. Although the grid has not been refined quite as far, the calculated solutions in all six cases have yet to level off, which occurred in the fictional case around $\Delta^* = 1$. However, for the first four cases, PC5-PC8, the solution is at least clearly converging. The predicted convergence rates from the power law based extrapolation in these cases is between 0.96 and 1.4, quite a bit lower than some of the convergence rates previously observed, but more reasonable for the numerical methods employed. For the last two cases, PC9 and PC10, the solutions have not yet clearly approached an asymptote. The power-law extrapolations predict very low convergence rates, essentially zero in the case of PC10. In these cases, the mixed-order extrapolations appear to provide a somewhat more acceptable fit. The coefficients of both types of least squares regression are provided in Section 4.4. Unfortunately, even the mixed-order regression predicts an exact solution relatively far away from the calculated data. This would indicate a comparatively large numerical (grid) uncertainty, ε_i , which is largely determined from the difference between a computed solution and the extrapolated exact solution, see Equation 11.

4.3 GIDASPOW BLEND DRAG LAW

Solution verification studies have also been performed on PC5–PC10 with the same set of CFD grids using the GIDASPOW_BLEND drag law. The results are summarized in Figure 30 to Figure 32. Qualitatively, the trends are nearly identical to the convergence behavior with the BVK drag law. Quantitatively, however, it can be seen that the nondimensional pressure drop is roughly 35 to 50% larger than with the BVK drag law. Interestingly, this is the opposite of the trend observed in Figure 23 for the fictional case, however the SSCPI conditions are at a much higher Reynolds number.

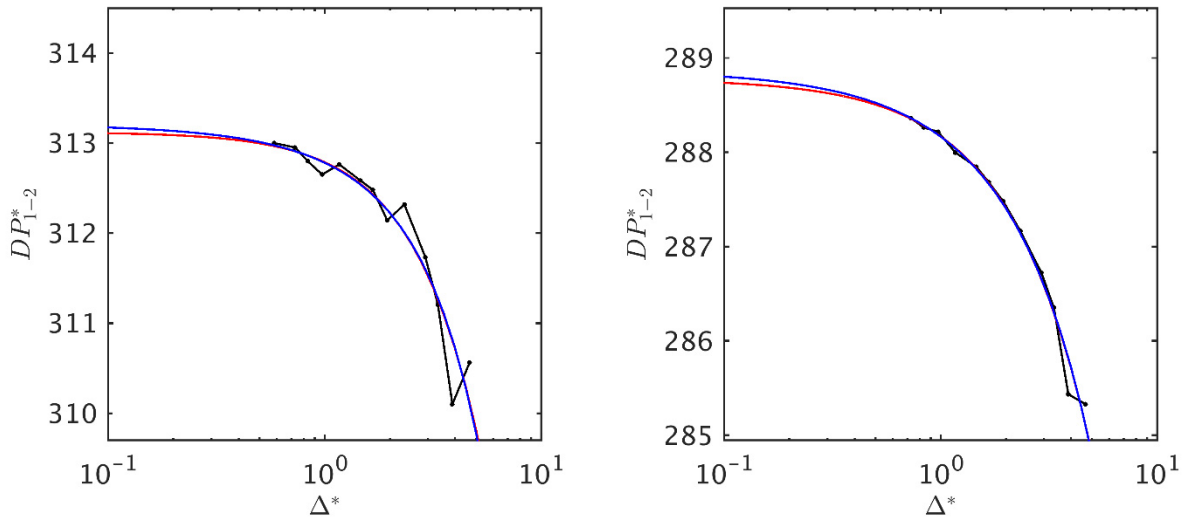


Figure 30: Convergence behavior of the bed pressure drop for PC5 (left) and PC6 (right) using the GIDASPOW_BLEND drag law compared to least-squares extrapolations of power law (red) and mixed-order (blue) regression types.

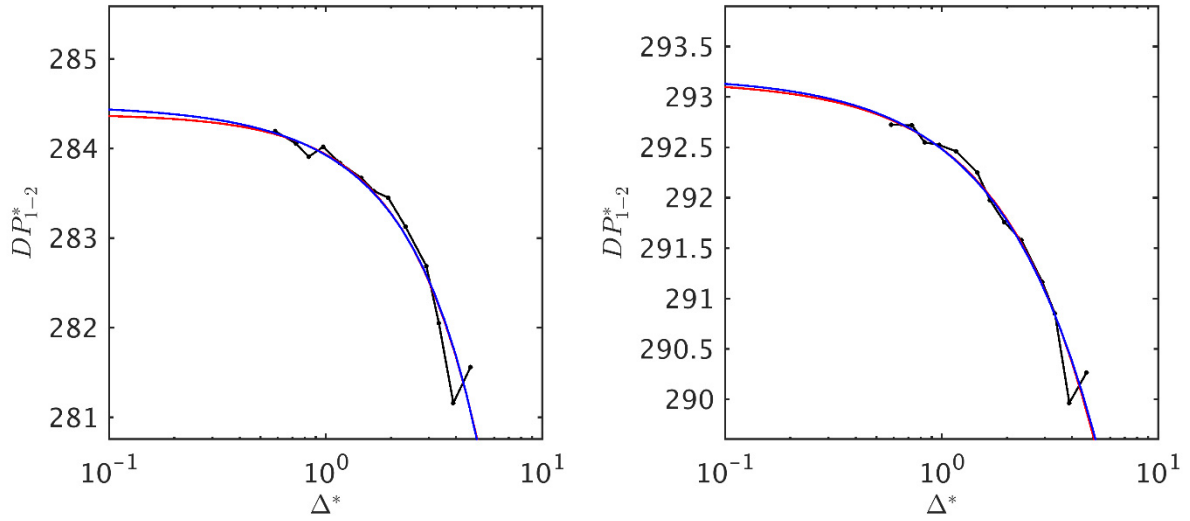


Figure 31: Same as Figure 30 for PC7 (left) and PC8 (right).

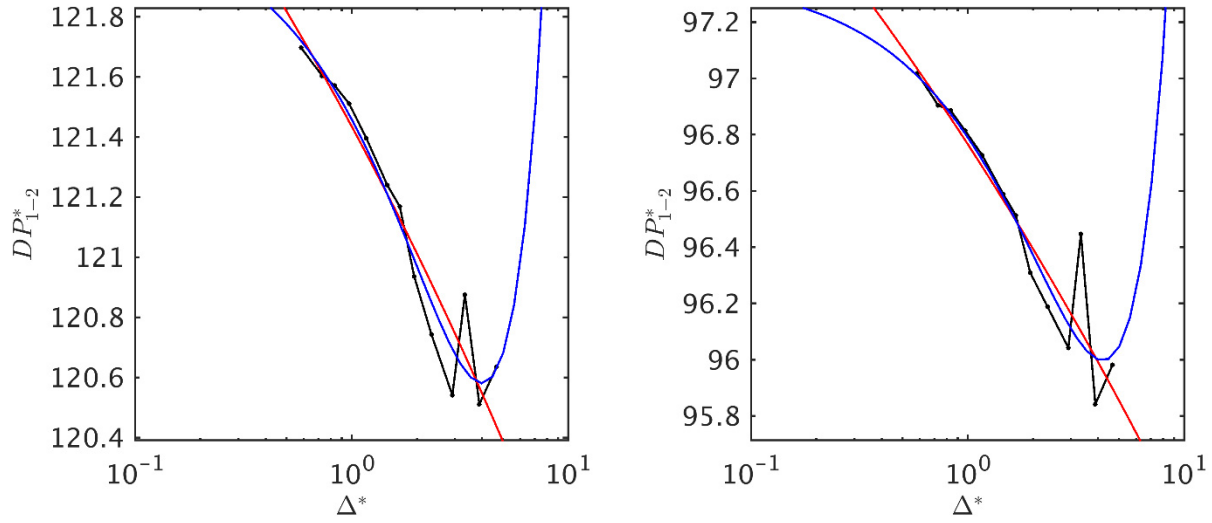


Figure 32: Same as Figure 30 for PC9 (left) and PC10 (right).

4.4 SUMMARY OF EXTRAPOLATED DATA

The results of the least-squares regression for the Richard extrapolation are summarized here for PC5-PC10. Table 4 provides the power-law regression coefficients and Table 5 the mixed-order regression coefficients. Results for both BVK and GIDASPOW_BLEND drag models are provided

Table 4: Summary of coefficients from least-squares extrapolation using power-law regression

	f_0	p	c
BVK, PC5	2.293768E+02	1.418199E+00	-2.638573E-01
BVK, PC6	2.079770E+02	9.605520E-01	-6.250076E-01
BVK, PC7	2.081712E+02	1.267951E+00	-3.574525E-01
BVK, PC8	2.114639E+02	1.108870E+00	-4.525479E-01
BVK, PC9	8.719776E+01	6.918749E-02	-5.721289E+00
BVK, PC10	5.785591E+03	5.778374E-05	-5.720112E+03
GIDASPOW_BLEND, PC5	3.131216E+02	1.422975E+00	-3.342532E-01
GIDASPOW_BLEND, PC6	2.887770E+02	1.163298E+00	-6.107155E-01
GIDASPOW_BLEND, PC7	2.843857E+02	1.292549E+00	-4.517571E-01
GIDASPOW_BLEND, PC8	2.931625E+02	1.023624E+00	-6.771372E-01
GIDASPOW_BLEND, PC9	1.254982E+02	1.425360E-01	-4.063413E+00
GIDASPOW_BLEND, PC10	1.011310E+02	1.177375E-01	-4.365198E+00

Table 5: Summary of coefficients from least-squares extrapolation using mixed-order regression

	f_0	c_1	c_2
BVK, PC5	2.294458E+02	-2.880215E-01	-4.975930E-02
BVK, PC6	2.079757E+02	-6.330566E-01	1.083315E-02
BVK, PC7	2.082474E+02	-4.064803E-01	-3.215461E-02
BVK, PC8	2.115106E+02	-4.905091E-01	-1.147158E-02
BVK, PC9	8.195706E+01	-5.298259E-01	6.765167E-02
BVK, PC10	6.587477E+01	-4.358104E-01	5.643766E-02
GIDASPOW_BLEND, PC5	3.132099E+02	-3.648351E-01	-6.405998E-02
GIDASPOW_BLEND, PC6	2.888680E+02	-6.759751E-01	-2.776830E-02
GIDASPOW_BLEND, PC7	2.844867E+02	-5.125789E-01	-4.690295E-02
GIDASPOW_BLEND, PC8	2.932022E+02	-7.233375E-01	4.329973E-03
GIDASPOW_BLEND, PC9	1.221426E+02	-7.821042E-01	9.790477E-02
GIDASPOW_BLEND, PC10	9.736067E+01	-6.476843E-01	7.692264E-02

4.5 NUMERICAL UNCERTAINTY

This section performs the next step in a solution verification study and calculates the numerical uncertainty, ε . In a practical computational science and engineering study, calculating or approximating ε is really the goal of a solution verification study. Understanding if a grid-refinement procedure was even possible for the CFD-DEM numerical method was the primarily motivation of this work. However, the less than ideal extrapolations for PC5–PC10 beg the question, “*How large is the predicted numerical uncertainty in these cases?*”.

With the Richard extrapolation curve-fits at hand, it is relatively straightforward to estimate the numerical uncertainty. A quick comment on the difference between numerical *error* and numerical *uncertainty*. In theory, if the extrapolation of solution data into the limit of zero grid spacing is credible, it is possible to correct the discrete solutions with the numerical error, i.e., use the exact (grid-free) solution, and the solution can be treated as being free of numerical errors or uncertainties. In practice, it is much more frequently the case that there is not a high level of confidence and credibility in the extrapolation. In this case, it is advisable to use the calculated error bound of the exact solution to within an uncertainty interval from the discrete solution (Oberkampf and Roy, 2010). The conversion of error to uncertainty essentially amounts to taking the unsigned error multiplied by a safety factor commensurate with the degree of confidence in the estimate. Here, the numerical uncertainty is estimated by,

$$\epsilon_i = F_s(|f_0 - f_i| + \sigma_{RE}), \quad (11)$$

where F_s is the factor of safety, σ_{RE} is the standard deviation of the regression,

$$\sigma_{RE} = \sqrt{\frac{n_g}{n_g-3} \sum_{i=1}^{n_g} (f_i - \hat{f}_i)^2}, \quad (12)$$

and $\hat{f}_i$ is the extrapolated solution at the same level of grid refinement as the calculated solution, i.e., $\hat{f}_i = f_0 + ch_i^p$ for power-law regression and for $\hat{f}_i = f_0 + c_1 h_i + c_2 h_i^2$ mixed-order regression. The factor of safety is taken to be $F_s = \max[1.25, 3\sigma_{RE}/R_f]$ where $R_f = (\max f_i - \min f_i)/(n_g - 1)$ is the data range parameter. The ratio of the standard deviation of the regression to the data range parameter, σ_{RE}/R_f , can be used to quantify the “goodness of fit,” with values less than unity being considered good. It should be noted that the definitions of ε_i and F_s differ slightly than in the methodology proposed by Eça and Hoekstra (2014), which have slightly different forms depending on the goodness of fit. A single, consistent definition is used here as some regressions (of PC5 – PC10) fail the test ($\sigma_{RE}/R_f > 1$), while others pass ($\sigma_{RE}/R_f < 1$).

Table 6: Estimated numerical uncertainties for PC5 – PC10 on grid $N_z = 6$

	BVK	GIDASPOW_BLEND	
PC5	1.422479E+01	1.825696E+01	power-law
PC6	5.915502E+00	1.054684E+01	power-law
PC7	1.159393E+01	1.553671E+01	power-law
PC8	6.246764E+00	5.642258E+00	power-law
PC9	1.717361E+01	2.625377E+01	mixed-order
PC10	1.453015E+01	1.925627E+01	mixed-order

The least-squares Richardson extrapolations displayed in Figure 24 to Figure 32 are used to estimate ε_i as defined in Equation 11. For PC5–PC8, the power-law form is applied and the mixed-order method is applied for PC9 and PC10. This change in form is consistent with the methodology of Eça and Hoekstra (2014), which also recommend a mixed-order regression if the power-law regression predicts an order of convergence less than 0.5. Table 6 provides the numerical uncertainties for the grid $N_z = 6$ ($\Delta^* = 3.89$) for both drag laws. The errors are appended to the $N_z = 6$ grid solutions and shown along with the complete transient of DP_{1-2}^* calculated in the original simulation in Figure 33. Note that the original simulation also used the $N_z = 6$ grid, but utilized GARG_2012 interpolation and the WEN_YU drag law. There are several noteworthy results displayed in Figure 33, including:

- The predicted numerical uncertainty is neither non-negligable nor (seemingly) excessively large. For DP5–DP8, the relative discretization error (RDE), ε_i/f_0 , ranges from 1–7%. However, it should be noted that the case considered here, $N_z = 6$, is, on average, the worst one, i.e., farthest from the extrapolated exact solution.
- Not surprisingly, the largest uncertainty is predicted in the PC9 and PC10 cases which have an average RDE of approximately 21%.

- In general, the uncertainty due to the drag law is larger than the uncertainty due to the grid, however the error bars do overlap in the PC9 and PC10.
- Larger yet is simply the natural variation in DP_{1-2}^* from the dynamics of the bubbling bed.

While not conclusive, the results suggest a hierarchy for practical CFD-DEM simulations as 1) uncertainty due to time averaging, 2) model form uncertainty of important sub-models, e.g., drag, and 3) numerical uncertainty due to the grid. Where model input uncertainties rank in this list can not be assessed outside of a particular validation exercise. Additionally, it should also be cautioned that numerical uncertainties due to the temporal discretization cannot be assessed in a frozen state.

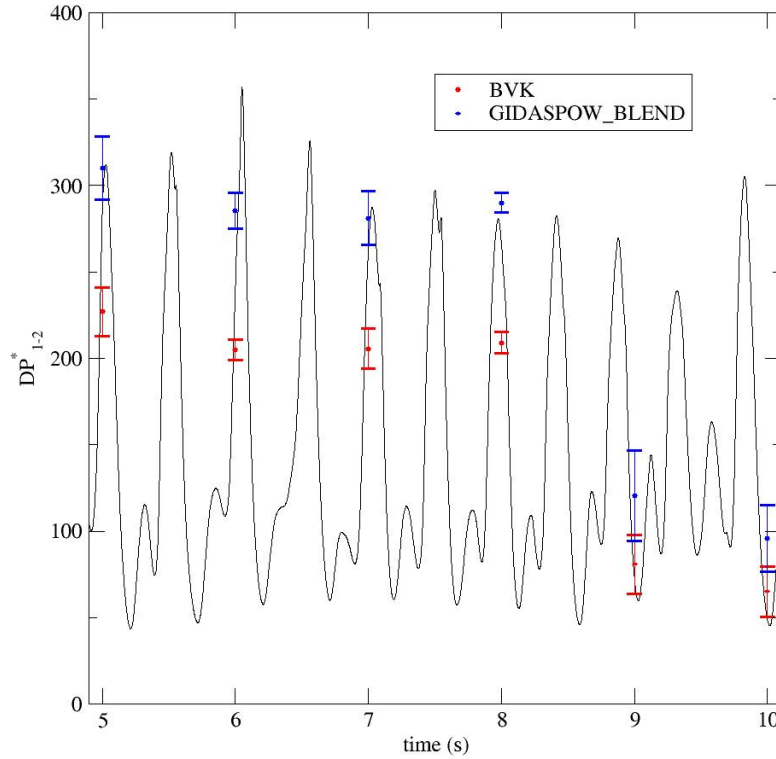


Figure 33: Comparison of the fixed-bed pressure drops with the BVK (red) and GIDASPOW_BLEND (blue) drag laws on the $N_z = 6$ grid with estimated numerical uncertainties. The results from the six fixed-bed PCs are superimposed over the complete transient results of the bed pressure drop predicted from the original simulation.

5. CONCLUSIONS

The most salient findings of this work are outlined below.

- Is solution verification even possible for CFD-DEM? Yes. Several fixed-bed cases (particles are frozen in place) were studied which showed that as the CFD grid is refined below the scale of the particle, the model does converge to a unique solution.
- While the solution did converge, here the SRQoI being a bed pressure drop, the convergence behavior was quite oscillatory and at times, an oscillatory behavior superimposed on a (globally) non-monotonic behavior. Using determined two or three point traditional Richard extrapolation methods would result in inaccurate and unreasonably large error estimates, i.e., extrapolate the exact solution far from the data. To circumvent this issue many grid resolutions were studied and extrapolations were found as the least-squares solutions to the overdetermined system, a method largely developed by Eça and Hoekstra (2014). The value and utility of the regression based extrapolations were clearly demonstrated in this work.
- The predicted rates of convergence are quite different than previously observed for first-order upwind, Superbee and MUSCL flux-limiters on a much simpler continuum problem (Waterson and Deconinck, 2007). However, the ranking $p^{(\text{FOU})} < p^{(\text{Superbee})} < p^{(\text{MUSCL})}$ is preserved.
- Qualitatively similar convergence behavior can be observed with a different diffusion filter width than the baseline choice $\delta_f^* = 6$. This was not a validation study and the (most) appropriate value of δ_f^* cannot be determined from the verification exercises herein. However, it seems that δ_f^* between 2 and approximately 12 seems to be advisable, consistent with other studies using a diffusion filtering scheme (Capecelatro and Desjardins 2013; Capecelatro et al., 2015).
- Qualitatively similar convergence behavior can also be observed with a different choice of drag law, BVK, GIDASPOW_BLEND, KOCH_HILL, and WEN_YU being studied.
- Quantitatively, the impact of modeling choices (e.g., drag, filter width) and the specific particle configuration (i.e., a particular instantaneous state of a transient, dynamic problem) appear to have a bigger impact on the solution than the numerical grid, at least for $\Delta^* \lesssim 5$.

In closing, we suggest exercising caution using this study as a more general path forward for CFD-DEM solution verification. That is, for a given problem, freeze the simulation at several random instances, perform grid refinement, calculate the uncertainty in the SRQoIs, average the uncertainties over an ensemble of frozen PCs, and attribute the resulting ensemble average uncertainty to the full, time-averaged solution. However, drawbacks are that: 1) such a method cannot quantify the time discretization error/uncertainty, and 2) since different states (PCs) may have very different uncertainties—as in Figure 33, also see data in Table 4 to Table 6, where the RDE of PC10 is roughly an order of magnitude larger than the RDE present in PC8—the number of PCs in the ensemble may need to be large and should equally sample the different states of the trajectory being time averaged, i.e., such an approach may end up being as computationally expensive as simply performing full transient simulations on different numerical grids. Although there are some clear drawbacks to such an approach, one advantage is that grid discretization error

can be calculated independently from time averaging error. Future studies aimed at the feasibility solution verification of transient CFD-DEM, i.e., calculating numerical error in the presence of time-averaging error, are currently being planned.

6. REFERENCES

- Baran, O.; Greening, D.; Kodl, P. Numerical simulations of fluidized beds in fine-grid geometries using source smoothing with cell clustering method for DEM-CFD coupling. Presented at the 2016 AIChE Annual Meeting, San Francisco, CA, Nov 13–18, 2016.
- Beetstra, R.; van der Hoef, M. A.; Kuipers, J. A. M. Drag force of intermediate Reynolds number flow past mono- and bidisperse arrays of spheres, *AIChE Journal* **2007**, *53*, 489–501.
- Benyahia, S.; Syamlal, M.; O'Brien, T. J. Extension of Hill-Koch-Ladd drag correlation over all ranges of Reynolds number and solids volume fraction. *Powder Technology* **2006**, *162*, 166–174.
- Capecelatro, J.; Desjardins, O. An Euler-Lagrange strategy for simulating particle-laden flows. *Journal of Computational Physics* **2013**, *238*, 1–31.
- Capecelatro, J.; Desjardins, O.; Fox, R. O. Effect of domain size on fluid-particle statistics in homogeneous, gravity-driven, cluster-induced turbulence. *Journal of Fluids Engineering* **2016**, *138*.
- Capecelatro, J.; Desjardins, O.; Fox, R. O. On fluid–particle dynamics in fully developed cluster-induced turbulence. *Journal of Fluid Mechanics* **2015**, *780*, 578–635.
- Celik, I. B.; Cehreli, Z.; Yavuz, I. Index of resolution quality for Large Eddy Simulations. *Journal of Fluids Engineering* **2005**, *127*, 949–958.
- Celik, I. B.; Ghia, U.; Roache, P. J.; Freitas, C. J. Procedure for estimation and reporting of uncertainty due to discretization in CFD applications. *Journal of Fluids Engineering* **2008**, *130*.
- Celik, I. B.; Li, J. Assessment of numerical uncertainty for the calculations of turbulent flow over a backward-facing step. *International Journal for Numerical Methods in Fluids* **2005**, *49*, 1015–1031.
- Ding, J.; Gidaspow, D. A bubbling fluidization model using kinetic theory of granular flow. *AIChE Journal* **1990**, *36*, 523–538.
- Eça, L.; Hoekstra, M. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *Journal of Computational Physics* **2014**, *262*, 104–130.
- Fullmer, W. D.; Hrenya, C. M. The clustering instability in rapid granular and gas-solid flows. *Annual Review of Fluid Mechanics* **2017**, *49*, 485–510.
- Fullmer, W. D.; Liu, G.; Yin, X.; Hrenya, C. M. Clustering instabilities in sedimenting fluid-solid systems: Critical assessment of kinetic-theory-based predictions using direct numerical simulation data. *Journal of Fluid Mechanics* **2017**, *823*, 433–469.
- Garg, R.; Galvin, J.; Li, T.; Pannala, S. *Documentation of open-source MFIx-DEM software for gas-solids flows*; Technical Report; National Energy Technology Laboratory: Morgantown, WV, 2012a. https://mfix.netl.doe.gov/documentation/dem_doc_2012-1.pdf
- Garg, R.; Galvin, J.; Li, T.; Pannala, S. Open-source MFIx-DEM software for gas-solids flows: Part I--Verification studies. *Powder Technology* **2012b**, *220*, 122–137.
- Garg, R.; Narayanan, C.; Lakehal, D.; Subramaniam, S. Accurate numerical estimation of interphase momentum transfer in Lagrangian-Eulerian simulations of dispersed two-7phase flows. *International Journal of Multiphase Flow* **2007**, *33*, 1337–1364.

- Gel, A.; Li, T.; Gopalan, B.; Shahnam, M.; Syamlal, M. Validation and uncertainty quantification of a multiphase computational fluid dynamics model. *Industrial & Engineering Chemistry Research* **2013**, *52*, 11424–11435.
- Gopalan, B.; Shahnam, M.; Panday, R.; Tucker, J.; Shaffer, F.; Shadle, L.; Mei, J.; Rogers, W.; Guenther, C.; Syamlal, M. Measurements of pressure drop and particle velocity in a pseudo 2-D rectangular bed with Geldart Group D particles. *Powder Technology* **2016**, *291*, 299–310.
- Hill, R. J.; Koch, D. L.; Ladd, A. J. Moderate-Reynolds-number flows in ordered and random arrays of spheres. *Journal of Fluid Mechanics* **2001a**, *448*, 243–278.
- Hill, R. J.; Koch, D. L.; Ladd, A. J. The first effects of fluid inertia on flows in ordered and random arrays of spheres. *Journal of Fluid Mechanics* **2001b**, *448*, 213–241.
- Lathouwers, D.; Bellan, J. Modeling of dense gas-solid reactive mixtures applied to biomass pyrolysis in a fluidized bed. *International Journal of Multiphase Flow* **2001**, *27*, 2155–2187.
- Liu, P.; LaMarche, C. Q.; Kellogg, K. M.; Hrenya, C. M. Fine-particle defluidization: Interaction between cohesion, Young's modulus and static bed height. *Chemical Engineering Science* **2016**, *145*, 266–278.
- Oberkampf, W. L.; Roy, C. J. *Verification and Validation in Scientific Computing*; Cambridge University Press: Cambridge, UK, 2010.
- Rider, W.; Witkowski, W.; Kamm, J. R.; Wildey, T. Robust verification analysis. *Journal of Computational Physics* **2016**, *307*, 146–163.
- Roache, P. J. Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering* **1994**, *116*, 405–413.
- Roache, P. J. *Verification and Validation in Computational Science and Engineering*; Hermosa Publisheres: Albuquerque, NM, 1998.
- Roy, C. J. Grid convergence error analysis for mixed-order numerical schemes. *AIAA Journal*, **2003**, *41*, 595–604.
- Sun, R.; Xiao, H. Diffusion-based coarse graining in hybrid continuum–discrete solvers: Applications in CFD–DEM. *International Journal of Multiphase Flow* **2015**, *72*, 233–247.
- Tenneti, S.; Subramaniam, S. Particle-resolved direct numerical simulation for gas-solid flow model development. *Annual Review of Fluid Mechanics* **2014**, *46*, 199–230.
- van Leer, B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics* **1979**, *32*, 101–136.
- Waterson, N. P.; Deconinck, H. Design principles for bounded higher-order convection schemes—a unified approach. *Journal of Computational Physics* **2007**, *224*, 182–207.
- Wen, C. Y.; Yu, Y. H. Mechanics of fluidization. *Chemical Engineering Progress Symposium Series* **1966**, *62*, 100–111.

A.1 USER MODIFICATIONS

MODULE usr

usr0.f

```

! user-definable. The user may insert code in this routine C
! or call appropriate user defined subroutines. This C
! can be used for setting constants and checking errors in C
! data. This routine is not called from an IJK loop, hence C
! all indices are undefined. C
! C
! Author: Date: dd-mmm-yy C
! Reviewer: Date: dd-mmm-yy C
! C
! Revision Number: C
! Purpose: C
! Author: Date: dd-mmm-yy C
! Reviewer: Date: dd-mmm-yy C
! C
! Literature/Document References: C
! C
! Variables referenced: C
! Variables modified: C
! C
! Local variables: C
! C
! ~~~~~C
!
SUBROUTINE USR0
!...Translated by Pacific-Sierra Research VAST-90 2.06G5 12:17:31 12/09/98
!...Switches: -xf
      Use usr
      USE compar, ONLY: myPE, PE_IO
      IMPLICIT NONE
!-----
!
! Include files defining common blocks here

```

```

!
!
! Define local variables here
!
!
! Include files defining statement functions here
!
!
! Insert user-defined code here
!
  IF (myPE .EQ. PE_IO) THEN
    OPEN(UNIT=801,FILE='norms.dat',STATUS='UNKNOWN')
    CLOSE(801)
    max_Linf_norm = 0.0d0
  END IF

  RETURN
END SUBROUTINE USR0

```

usr1.f

```

!~~~~~C
!
! Module name: USR1 C
! Purpose: This routine is called from the time loop and is C
! user-definable. The user may insert code in this routine C
! or call appropriate user defined subroutines. This C
! can be used for setting or checking errors in quantities C
! that vary with time. This routine is not called from an C
! IJK loop, hence all indices are undefined. C C
!
! Author: Date: dd-mmm-yy C
! Reviewer: Date: dd-mmm-yy C
!
! Revision Number: C
! Purpose: C
! Author: Date: dd-mmm-yy C
! Reviewer: Date: dd-mmm-yy C
!
! Literature/Document References: C
!
! Variables referenced: C
! Variables modified: C
!
! Local variables: C
!
!~~~~~C
!
SUBROUTINE USR1
!...Translated by Pacific-Sierra Research VAST-90 2.06G5 12:17:31 12/09/98
!...Switches: -xf
  USE usr
  USE functions
  USE mpi_utility

```

```

USE indices, ONLY: I_OF, J_OF, K_OF
USE toleranc, ONLY: TOL_RESID
USE run, ONLY: DT, DT_MIN, TIME, TSTOP
USE fldvar, ONLY: P_G, U_G, V_G, W_G
USE fldvar, ONLY: P_GO, U_GO, V_GO, W_GO
USE physprop, ONLY: RO_G0, D_p0
USE bc, ONLY: BC_V_G
USE compar, ONLY: myPE, PE_IO

IMPLICIT NONE

INTEGER :: ii, jj, kk, ijk, Nquickly, ierr
DOUBLE PRECISION :: norm_dudt, norm_dpdt
DOUBLE PRECISION, DIMENSION(4) :: Linf_norms, Linf_norms_global
DOUBLE PRECISION :: old_max_Linf_norm, TOL_TRANS_MAX

!-----
!
! Include files defining common blocks here
!
!
! Define local variables here
!
    TOL_TRANS_MAX = 1.0d-6
    norm_dudt = D_p0(1)/BC_V_G(1)**2
    norm_dpdt = D_p0(1)/(RO_G0*BC_V_G(1)**3)
!
! Include files defining statement functions here
!
!
! Insert user-defined code here
!

! Calc L_inf norms
Linf_norms(:) = 0.0d0
DO ijk = IJKSTART3, IJKEND3
IF(.NOT.IS_ON_myPE_wobnd(I_OF(IJK), J_OF(IJK), K_OF(IJK))) CYCLE
IF (FLUID_AT(IJK)) THEN
    Linf_norms(1) = DMAX1(Linf_norms(1), DABS(P_G(ijk)-P_GO(IJK)))
    Linf_norms(2) = DMAX1(Linf_norms(2), DABS(U_G(ijk)-U_GO(IJK)))
    Linf_norms(3) = DMAX1(Linf_norms(3), DABS(V_G(ijk)-V_GO(IJK)))
    Linf_norms(4) = DMAX1(Linf_norms(4), DABS(W_G(ijk)-W_GO(IJK)))
END IF
END DO

CALL GLOBAL_MAX(Linf_norms(1), Linf_norms_global(1), PE_IO, ierr)
CALL GLOBAL_MAX(Linf_norms(2), Linf_norms_global(2), PE_IO, ierr)
CALL GLOBAL_MAX(Linf_norms(3), Linf_norms_global(3), PE_IO, ierr)
CALL GLOBAL_MAX(Linf_norms(4), Linf_norms_global(4), PE_IO, ierr)

805 FORMAT (5(E16.8, 2x))
IF (myPE .EQ. PE_IO) THEN
! normalize
    Linf_norms(1) = Linf_norms_global(1)/DT*norm_dpdt
    Linf_norms(2:4) = Linf_norms_global(2:4)/DT*norm_dudt
! write out

```

```

      OPEN(UNIT=801, FILE='norms.dat', ACCESS='APPEND', STATUS='OLD')
      WRITE(801,805) TIME, Linf_norms(1), Linf_norms(2), &
        Linf_norms(3), Linf_norms(4)
      CLOSE(801)
! stop if max Linf and previous max Linf meet tolerance
      old_max_Linf_norm = max_Linf_norm
      max_Linf_norm = MAXVAL(Linf_norms)
      IF ((max_Linf_norm + old_max_Linf_norm)/2.0 &
        .LE. TOL_TRANS_MAX) THEN
        TSTOP=0.0
        WRITE(*,*) 'Convergence tolerances met: '
        WRITE(*,*) 'old_max_Linf_norm = ', old_max_Linf_norm
        WRITE(*,*) ' max_Linf_norm = ', max_Linf_norm
      END IF
    END IF

! send PE_IO's TSTOP to all processors
    CALL BCAST(TSTOP, PE_IO)

  RETURN
END SUBROUTINE USR1

```

A.2 BASELINE INPUT DECK FOR BURSTING BUBBLE CASE

! attempte at fixed bed cfd-dem simulation
! wdf 7/2017

```

! Run-control
RUN_NAME      = 'fixbed'
DESCRIPTION    = 'fixed (static) bed'
RUN_TYPE      = 'new'
UNITS         = 'cgs'
TIME          = 0.0
TSTOP         = 10000.0
DT            = 1.0d-4
DT_MIN        = 2.0d-12
DT_MAX        = 100.0d0
DT_FAC        = 0.75d0

ENERGY_EQ     = .FALSE.
SPECIES_EQ    = .FALSE. .FALSE.

MOMENTUM_X_EQ(0) = .true.
MOMENTUM_Y_EQ(0) = .true.
MOMENTUM_Z_EQ(0) = .true.
MOMENTUM_X_EQ(1) = .FALSE.
MOMENTUM_Y_EQ(1) = .FALSE.
MOMENTUM_Z_EQ(1) = .FALSE.

DRAG_TYPE = 'BVK'

GRAVITY = 0

```

```

MAX_NIT = 500
DETECT_STALL = .FALSE.
NORM_G = 0
TOL_RESID = 1.0d-5
LEQ_TOL(1:9) = 9*1.0d-8
LEQ_IT(1:9) = 9*20
LEQ_TOL(10) = 1.0d-10
LEQ_IT(10) = 100
LEQ_PC(10) = 'NONE'
Max_Inlet_Vel_Fac = 10000.0
DISCRETIZE(1:10) = 10*2
! UR_FAC(1:10) = 10*0.75

! Domain
COORDINATES      = 'cartesian'
XLENGTH          = 0.3d0
!runtime IMAX      = KMAX x3
YLENGTH          = 2.0d0
!runtime JMAX      = KMAX x20
ZLENGTH          = 0.1d0
!runtime KMAX      = (4, 5, 6, 7) x2 x2 x2 x2

! DES Domain
!runtime DESGRID = FLUID GRID

! Gas properties
MU_g0            = 2.0d-4
RO_g0            = 1.0d-3

! Particle properties
RO_s0            = 1.0d0
D_p0             = 0.01d0

! Initial Conditions Section
IC_X_w(1)        = 0.0
IC_X_e(1)        = 0.3
IC_Y_s(1)        = 0.0
IC_Y_n(1)        = 2.0
IC_Z_b(1)        = 0.0
IC_Z_t(1)        = 0.1

IC_EP_g(1)       = 0.82
IC_U_g(1)        = 0.0
IC_V_g(1)        = 24.4
IC_W_g(1)        = 0.0
IC_U_s(1,1)      = 0.0
IC_V_s(1,1)      = 0.0
IC_w_s(1,1)      = 0.0
IC_P_star(1)     = 0.0
IC_THETA_M(1,1)  = 0.0

! Boundary Conditions Section
! 1. Distributor flow
BC_X_w(1)        = 0.0
BC_X_e(1)        = 0.3
BC_Y_s(1)        = 0.0

```


BC_Y_n(1) = 0.0
BC_Z_b(1) = 0.0
BC_Z_t(1) = 0.1
BC_TYPE(1) = 'MI'

BC_EP_g(1) = 1.0
BC_U_g(1) = 0.0
BC_V_g(1) = 20.0
BC_W_g(1) = 0.0
BC_P_g(1) = 0.0

! 2. Exit

BC_X_w(2) = 0.0
BC_X_e(2) = 0.3
BC_Y_s(2) = 2.0
BC_Y_n(2) = 2.0
BC_Z_b(2) = 0.0
BC_Z_t(2) = 0.1
BC_TYPE(2) = 'PO'
BC_P_g(2) = 0.0

! x- and z-periodic

CYCLIC_X = .T.
CYCLIC_Z = .T.

! DEM parameters

SOLIDS_MODEL = 'DEM'
DES_INTERP_ON = .F.
DES_EXPLICITLY_COUPLED = .T.
DES_DIFFUSE_WIDTH = 0.06
NFACTOR = 0
PARTICLES = 5181
KN = 10.0
MEW = 0.0d0
DES_EN_INPUT = 0.8
DES_ETAT_FAC = 0.5
KN_W = 10.0
MEW_W = 0.0d0
DES_EN_WALL_INPUT = 0.8
DES_ETAT_W_FAC = 0.5d0

DES_NEIGHBOR_SEARCH = 4
NEIGHBOR_SEARCH_N = 999
GENER_PART_CONFIG = .F.
PRINT_DES_DATA = .F.

! DES_OUTPUT_TYPE = 'TECPLOT'

! Output Control

RESID_STRING = 'P0' 'U0' 'V0' 'W0'
FULL_LOG = .t.

!rm OUT_DT = 20000

RES_DT = 20000

! SPX_DT = 0.01 0.01 0.01 100. 100. 100. 100. 100. 100. 100.

SPX_DT = 11*20000.

```
CALL_USR = .T.
```

```
! Decomposition in I, J, and K directions for parallel run
!runtime NODESI = 2 NODESJ = 8 NODESK = 1
!runtime NODESI = 1 NODESJ = 1 NODESK = 1
!
```

A.3 PARTICLE GENERATOR SOURCE FOR BURSTING BUBBLE CASE

```
!*****
!
! DES input generator
! Sample generator to generate particle positions and outputs in DES readable format
! Please use this as a basis and perform any additional customizations as
! needed and this module does not come with any guarantees
!
! The particles generated by this code are greater than or equal to specified np and
! MFIX will pick up the correct particles according to value of np specified in the code
!
! TO DO:
! 1) Fix so that only np particle information is created
! 2) In MFIX read the entire input file and if the number of particles does not correspond to
!    the input deck, than flag an error.
!
! Author: Jay Boyalakuntla (May-12-06)
! Modified: S. Pannala (Nov-21-06)
! Modified: wdf (6-28-2017)
!
!*****
!
!
! PROGRAM DES_Particle_Genrator
!
!
! IMPLICIT NONE
!
! DOUBLE PRECISION, PARAMETER :: PI = 4.0d0*ATAN(1.0d0)
!
! INTEGER :: seedno
! INTEGER :: accept, touch, ia
! INTEGER :: ii, jj, kk, ijk
! INTEGER :: npdense, npdilute, np
!
! DOUBLE PRECISION :: xp, yp, zp
! DOUBLE PRECISION :: xl, yl, zl
! DOUBLE PRECISION :: diameter, radius, density
! DOUBLE PRECISION :: rbub, ysurf, x0, y0, z0
! DOUBLE PRECISION :: phidilute, Vdilute, phidense, Vdense
! DOUBLE PRECISION :: ru1, ru2, ru3, ru4
! DOUBLE PRECISION :: dist, mindist
! DOUBLE PRECISION :: T_0, temp
!
! DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: x, y, z
! DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: u, v, w
! DOUBLE PRECISION, DIMENSION(:), ALLOCATABLE :: gridx, gridy, gridz
```

```

LOGICAL, DIMENSION(:,:,:), ALLOCATABLE :: deltrue
!
!io  OPEN(unit=10, file="Pgen.in", status='old')
    OPEN(unit=20, file="particle_input.dat", status='replace')
!
    WRITE(*,*) ' '
    WRITE(*,*) ' Enter the seed no '
    WRITE(*,*) ' '
    READ(*,*) seedno
    diameter = 0.01d0
    radius = diameter/2.0d0
    density = 1.0d0
    xl = DBLE(30)*diameter
    zl = DBLE(10)*diameter
    yl = DBLE(50)*diameter
    phidilute = 0.01d0
    phidense = 0.30d0
    ysurf = xl
    rbub = DBLE(5)*diameter
    x0 = xl/2.0d0
    y0 = ysurf - rbub
    z0 = zl/2.0d0
    T_0 = 0.25d0

!      calc the number of particles in each region
    Vdense = (0.0962497 - PI*rbub**2)*zl !area under gaussian from wolframalpha.com
    Vdilute = xl*yl*zl - Vdense
    npdense = INT(phidense*Vdense/(PI/6.0d0*diameter**3))
    npdilute = INT(phidilute*Vdilute/(PI/6.0d0*diameter**3))
    np = npdense + npdilute
    WRITE(*,*) 'np dilute, dense, tot = ', npdilute, npdense, np

!      alloc
    ALLOCATE(x(np),y(np),z(np),u(np),v(np),w(np))

!      set random generator
    CALL RANDOM_SEED
    DO jj = 1, seedno - 1
        DO ii = 1, np
            CALL RANDOM_NUMBER(ru1)
        END DO
    END DO

!      generate dense region first
    DO ii = 1, npdense
100  CONTINUE
        CALL random_particle(radius,xp,yp,zp,xl,yl,zl)
!      in freeboard or bubble?
        IF ((yp .GT. ysurf+rbub*DEXP(-(xp-x0)**2/(2.0d0*rbub**2))) &
        .OR. ((xp-x0)**2 + (yp-y0)**2 .LT. rbub**2)) GOTO 100 !failed, in dilute region, try again
!      touching other particles?
        dist = 0.0d0
        DO jj = 1, ii-1
            dist = DSQRT((xp-x(jj))**2 + (yp-y(jj))**2 +&
            (zp-z(jj))**2)
            IF (dist .LE. diameter + 1.0d-6) GOTO 100 !failed, touching another particle, try again

```

```

        END DO
!       passed, keep particle
        x(ii) = xp
        y(ii) = yp
        z(ii) = zp
        WRITE(*,*) 'particle ', ii, ' seeded'
    END DO

!       generate dense region first
    DO ii = npdense+1, np
101    CONTINUE
        CALL random_particle(radius,xp,yp,zp,xl,yl,zl)
!       in freeboard or bubble?
        IF ((yp .LT. ysurf+rbub*DEXP(-(xp-x0)**2/(2.0d0*rbub**2))) &
        .AND. ((xp-x0)**2 + (yp-y0)**2 .GT. rbub**2)) GOTO 101 !failed, not in dilute region, try again
!       touching other particles?
        dist = 0.0d0
        DO jj = 1, ii-1
            dist = DSQRT((xp-x(jj))**2 + (yp-y(jj))**2 +&
                (zp-z(jj))**2)
            IF (dist .LE. diameter + 1.0d-6) GOTO 101 !failed, touching another particle, try again
        END DO
!       passed, keep particle
        x(ii) = xp
        y(ii) = yp
        z(ii) = zp
        WRITE(*,*) 'particle ', ii, ' seeded'
    END DO
    WRITE(*,*) np, 'particles generated successfully'

!       calc min separation distance
    WRITE(*,*) 'double checking min separation >= diameter'
    mindist = xl
    DO ii = 1, np
        DO jj = 1, ii-1
            dist = DSQRT((x(ii)-x(jj))**2 + (y(ii)-y(jj))**2 +&
                (z(ii)-z(jj))**2)
            mindist = MIN(mindist,dist)
        END DO
    END DO
    mindist = mindist/diameter
    WRITE(*,*) 'min separation distance normalized by dp is', mindist

!       generate random particle velocities
    DO ii = 1, np
        CALL RANDOM_NUMBER(ru1)
        CALL RANDOM_NUMBER(ru2)
        CALL RANDOM_NUMBER(ru3)
        CALL RANDOM_NUMBER(ru4)
!       random velocities uniform -> standard normal via Box-Muller
        u(ii) = DSQRT(-2.0d0*DLOG(DBLE(ru1)))*COS(2.0d0*PI*ru2)
        v(ii) = DSQRT(-2.0d0*DLOG(DBLE(ru1)))*SIN(2.0d0*PI*ru2)
        w(ii) = DSQRT(-2.0d0*DLOG(DBLE(ru3)))*COS(2.0d0*PI*ru4)
    END DO

```

```

!      Calc mean granular temperature
temp = 0.0d0
DO ii = 1, np
    temp = temp + (u(ii)**2 + v(ii)**2 + w(ii)**2)/3.0d0
END DO
temp = temp/DBLE(np)

!      Adjust velocities so that the mean granular temperature is equal
!      to the desired initial granular temperature from the input
u = u*DSQRT(T_0/temp)
v = v*DSQRT(T_0/temp)
w = w*DSQRT(T_0/temp)

!      Print to particle_input.dat
! 12  FORMAT (8(d10.4,2x))
12  FORMAT (8(e24.16,2x))
! 12  FORMAT (3(d24.16,2x),2(d12.4,2x),3(d24.16,2x))
DO ii = 1, np
    WRITE(20,12) x(ii), y(ii), z(ii), radius, density, u(ii), v(ii), w(ii)
END DO

STOP

!DEALLOCATE(x,y,z,u,v,w)
END PROGRAM

!
!
!~~~~~
! Random particle
!~~~~~
!
SUBROUTINE random_particle(rad, xp1, yp1, zp1, xl1, yl1, zl1)
!-----
! Dummy arguments
!-----
    DOUBLE PRECISION, INTENT(IN) :: rad, xl1, yl1, zl1
    DOUBLE PRECISION, INTENT(OUT) :: xp1, yp1, zp1
!-----
! Local variables
!-----
    INTEGER :: i, Nfail
    DOUBLE PRECISION :: rad1
    DOUBLE PRECISION :: pxy(3)

    Nfail = 100000
    DO i = 1, Nfail
        call random_number(pxy)
        xp1 = dble(pxy(1))*xl1
        yp1 = dble(pxy(2))*yl1
        zp1 = dble(pxy(3))*zl1
        rad1 = (1.0d0 + 1.0d-2)*rad
        IF((xp1.GE.rad1).AND.(xp1.LE.xl1-rad1).AND.(yp1.GE.rad1)&
            .AND.(yp1.LE.yl1-rad1).AND.(zp1.GE.rad1)&
            .AND.(zp1.LE.zl1-rad1)) EXIT
    END DO

```

END DO

IF(i .GT. Nfail) THEN

WRITE(*,*) 'error in subroutine random_particle'

WRITE(*,*) 'not able to place particle, i>Nfail'

STOP

ENDIF

RETURN

END SUBROUTINE random_particle

,



Sean Plasynski

Executive Director
Technology Development & Integration
Center
National Energy Technology Laboratory
U.S. Department of Energy

Jared Ciferno

Associate Director
Oil and Gas
Technology Development & Integration
Center
National Energy Technology Laboratory
U.S. Department of Energy

Elena Melchert

Director
Division of Upstream Oil and Gas
Research
U.S. Department of Energy

Cynthia Powell

Executive Director
Research and Innovation Center
National Energy Technology Laboratory
U.S. Department of Energy