CATEGORY: COMPUTATIONAL FLUID DYNAMICS - CFD08
POSTER P6107
GPU TECHNOLOGY CONFERENCE

CONTACT NAME
Andrew Larson: andrew.larson@cpfd-software.com

User-Defined Drag Models on the GPU

Andrew D. Larson, James M. Parker

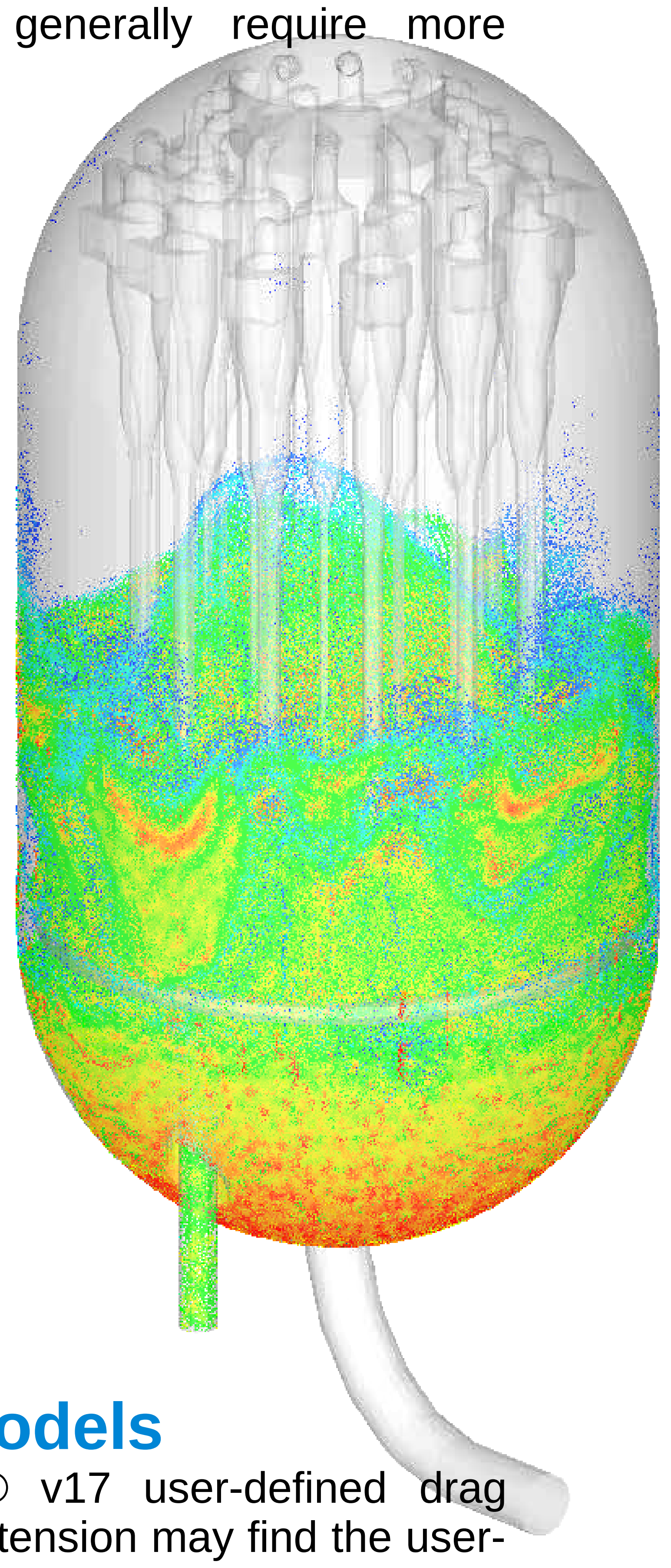
cpfd COMPUTATIONAL PARTICLE FLUID DYNAMICS
BARRACUDA VIRTUAL REACTOR

Overview

An approach to supporting user-defined drag models on the GPU is presented. Performance comparisons between compiled predefined models (PDM) and user-defined models (UDM) that are evaluated at runtime demonstrate the feasibility of allowing user-defined models on the GPU. In computational fluid dynamics simulations of fluidized beds, the particle drag force correlation significantly influences all aspects of the fluidization behavior including flow regime, bed density, bubbling characteristics, and entrainment. Some well-known models can be included at compile time but end users generally require more flexibility.

Motivation

Many industrialized processes use a fluidized bed to achieve an even and effective contacting between solid (particle) and fluid phases. When modeling fluidized systems, the calculation of the particle drag force is the primary source of uncertainty in the particle-fluid hydrodynamics. As such, the choice of drag correlation can significantly influence predictions. While drag models currently exist in literature (e.g. Ergun, Wen-Yu), this is an active area of research. New methods for calculating drag continue to evolve (e.g. EMMS, Beetstra). Compiling previously unknown models into Barracuda VR at run time is impossible. The significance and variability of drag models requires a more flexible approach.



User-Defined Drag Models

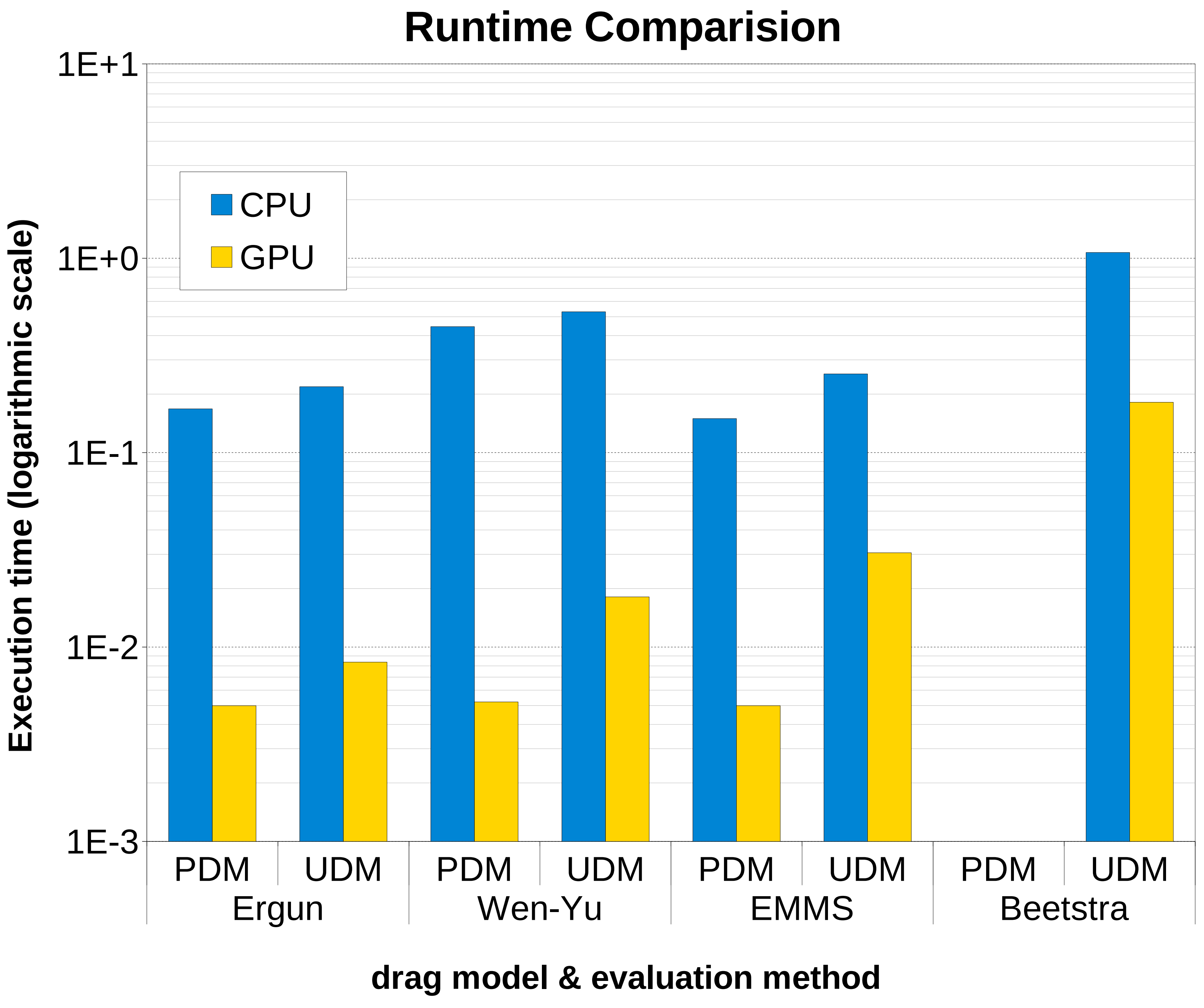
Beginning with Barracuda VR® v17 user-defined drag models are supported. Further extension may find the user-defined functionality appear in chemistry or thermal expressions. At runtime, a user can specify their own drag model for use in simulation. Expression syntax is Microsoft(R) Excel style IF-ELSE statements. The language supports arithmetic, simple branching, comparison, provides common values from simulation and other commonly found operations:

- +, -, *, /, ^
- IF-ELSE
- <, <=, >, >=, ==
- fluid volume fraction, Reynolds number, density, etc.
- *pow, max, min, sin, cos, log, exp, abs*

Expressions are converted to Reverse Polish Notation and evaluated using an software implemented stack in serial on CPU or in parallel on GPU.

Performance

Naive implementations will produce sub-standard results. Even for predefined models, running a kernel per species, reducing branching, gives needed performance that a naive implementation would not provide. On the GPU, shared memory is a limited resource and used to evaluate a user-defined drag expression. For simple drag expressions, assigning shared memory per thread works well. As expression complexity increases, serializing threads to reduce total shared memory per block gives noticeable improvement in kernel execution time. For complicated user-defined models, serializing 2 threads to use the same shared memory stack space reduces runtime by at least 10%. Without GPU acceleration, particle drag calculations in typical industrial-scale simulations (isothermal, non-reacting) can account for 2.5-7.5% of run time.



Drag Models

Drag models are normalized by Stokes Drag Law and are a function of Reynolds number (Re), voidage (ε), and particle volume fraction (θ).

Ergun (1952)

$$F = \frac{150}{18}\frac{(1-\varepsilon)}{\varepsilon^2} + \frac{1.75}{18}\frac{\mathrm{Re}}{\varepsilon^2}$$

Wen and Yu (1966)

$$F = \begin{cases} \left(1 + 0.15\mathrm{Re}^{0.687}\right)\varepsilon^{-3.65} & \mathrm{Re} \le 1000 \\ \dfrac{0.44}{24}\mathrm{Re}\,\varepsilon^{-3.65} & \mathrm{Re} > 1000 \end{cases}$$

EMMS (Yang et al., 2003)

$$F = \begin{cases} \dfrac{150}{18}\dfrac{(1-\varepsilon)}{\varepsilon^2} + \dfrac{1.75}{18}\dfrac{\mathrm{Re}}{\varepsilon^2} & \varepsilon < 0.74 \\ F_{sp}\cdot\omega & \varepsilon \ge 0.74 \end{cases}$$

$$F_{sp} = \begin{cases} 1 + 0.15\mathrm{Re}^{0.687} & \mathrm{Re} \le 1000 \\ \dfrac{0.44}{24}\mathrm{Re} & \mathrm{Re} > 1000 \end{cases}$$

$$\omega = \begin{cases} -0.576 + \dfrac{0.0214}{4(\varepsilon - 0.7463)^2 + 0.0044} & 0.74 \le \varepsilon < 0.82 \\ -0.0101 + \dfrac{0.0038}{4(\varepsilon - 0.7789)^2 + 0.004} & 0.82 \le \varepsilon \le 0.97 \\ -31.8295 + 32.8295\varepsilon & \varepsilon > 0.97 \end{cases}$$

Beetstra et al. (2007)

$$F = 10\theta\varepsilon^{-2} + \varepsilon^2\left(1 + 1.5\sqrt{\theta}\right) + \frac{0.413\mathrm{Re}}{24\varepsilon}\left(\frac{\varepsilon^{-1} + 3\varepsilon\theta + 8.4\mathrm{Re}^{-.343}}{1 + 10^{3\theta}\mathrm{Re}^{-0.5-2\theta}}\right)$$

Conclusions

Adding the capability for user-defined functions to be evaluated at runtime on the GPU is feasible. Although performance degrades with function complexity, overall, the performance is useful for providing flexibility in a commercial software product. Effort has been given to increasing performance of this kernel, but overall, performance is acceptable. Looking at PDM vs UDM comparisons, CPU results indicate that more effort optimizing the GPU implementation could yield similar trends significantly improving performance for larger, more complicated user-defined functions.

References

1. Beetstra, R., van der Hoef, M., and Kuipers, J. (2007). Drag Force of Intermediate Reynolds Number Flow Past Mono- and Bidisperse Arrays of Spheres. *AIChE Journal.* 53(2): 489-501.
2. Ergun, S. (1952). Fluid flow through packed columns. *Chemical Engineering Progress*. 48: 89-94.
3. Wen, C. and Yu, Y. (1966). Mechanics of Fluidization. *Chemical Engineering Progress Symposium Series.* 62: 100-111.
4. Yang, N, Wang, W., Ge, W., and Li, J. CFD simulation of concurrent-up gas-solid flow in circulating fluidized beds with structure dependent drag coefficient. *Chemical Engineering Journal.* 96: 71-80.