BARRACUDA VIRTUAL REACTOR®

16-GPU Release Guide

September 2013

CPFD Software LLC
10899 Montgomery Blvd. NE, Suite A
Albuquerque, NM 87111
+1.505.275.3849
www.cpfd-software.com

Contents

General Overview

- This Release Guide presents an overview of the advanced features in Barracuda VR 16-GPU.

- These major features include:
  – Solver efficiency improvements to reduce calculation time
  – Parallel computations on a GPU card – ‘desktop supercomputer’
  – Blended acceleration model for interpenetration contact force
  – User-defined drag models
  – Support for chemical reaction conversion terms
  – A discrete particle shrinking core model for chemistry
  – A Barracuda plot manager
  – Other GUI and usability improvements

GPU Parallel Acceleration

- Graphics processing unit (GPU) on video cards or dedicated GPU cards are designed for very fast parallel calculations.

- GPU cards are increasingly being added to desktop computers to create “desktop supercomputers” for computational fluid dynamics.

- Barracuda VR 16-GPU contains significant solver efficiency improvements including parallelization on an onboard NVIDIA GPU card (Initially Linux systems only, separate license required).
  - Note that this is a first release of a GPU-enabled solver. Additional speed-ups are anticipated for future releases.

- This results in a significant acceleration of the calculation rate for simulations.

GPU Parallel Acceleration Speed-Up Examples

- CPFD Software has run timing tests on many different types of systems that are typical of a variety of Barracuda applications. The timing results show significant speed-up across all problems.

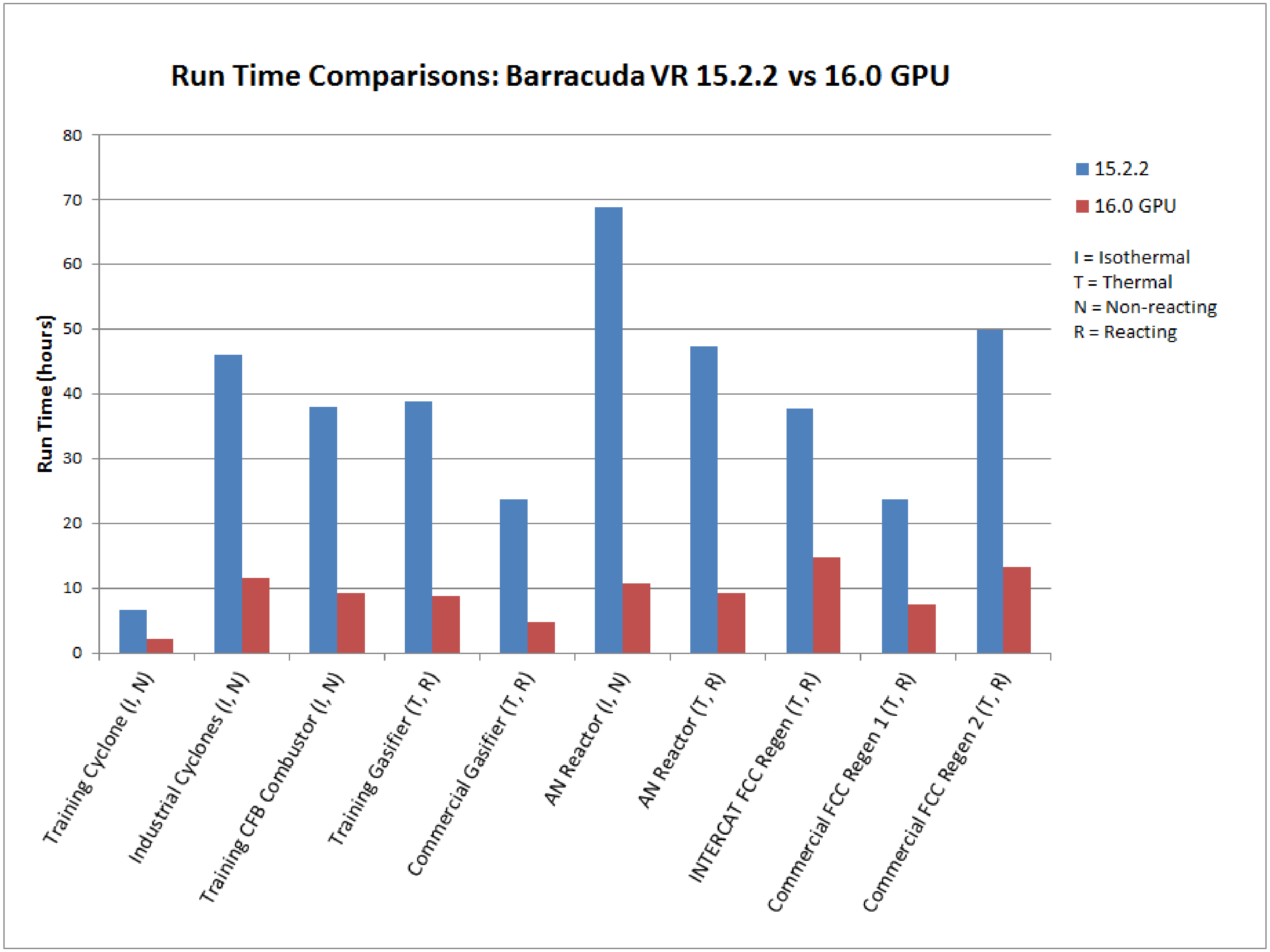
- It is important to keep the following in mind when discussing problem speed-up statistics:

    - The speed-up is problem-dependent, and a function of model size, computational particle count, and problem physics. For example, a large reacting system will likely have different speed-up than a small isothermal system. Problems with more computational particles may have a greater speedup than similar problems with fewer computational particles. Very small problems may not see any speed-up.

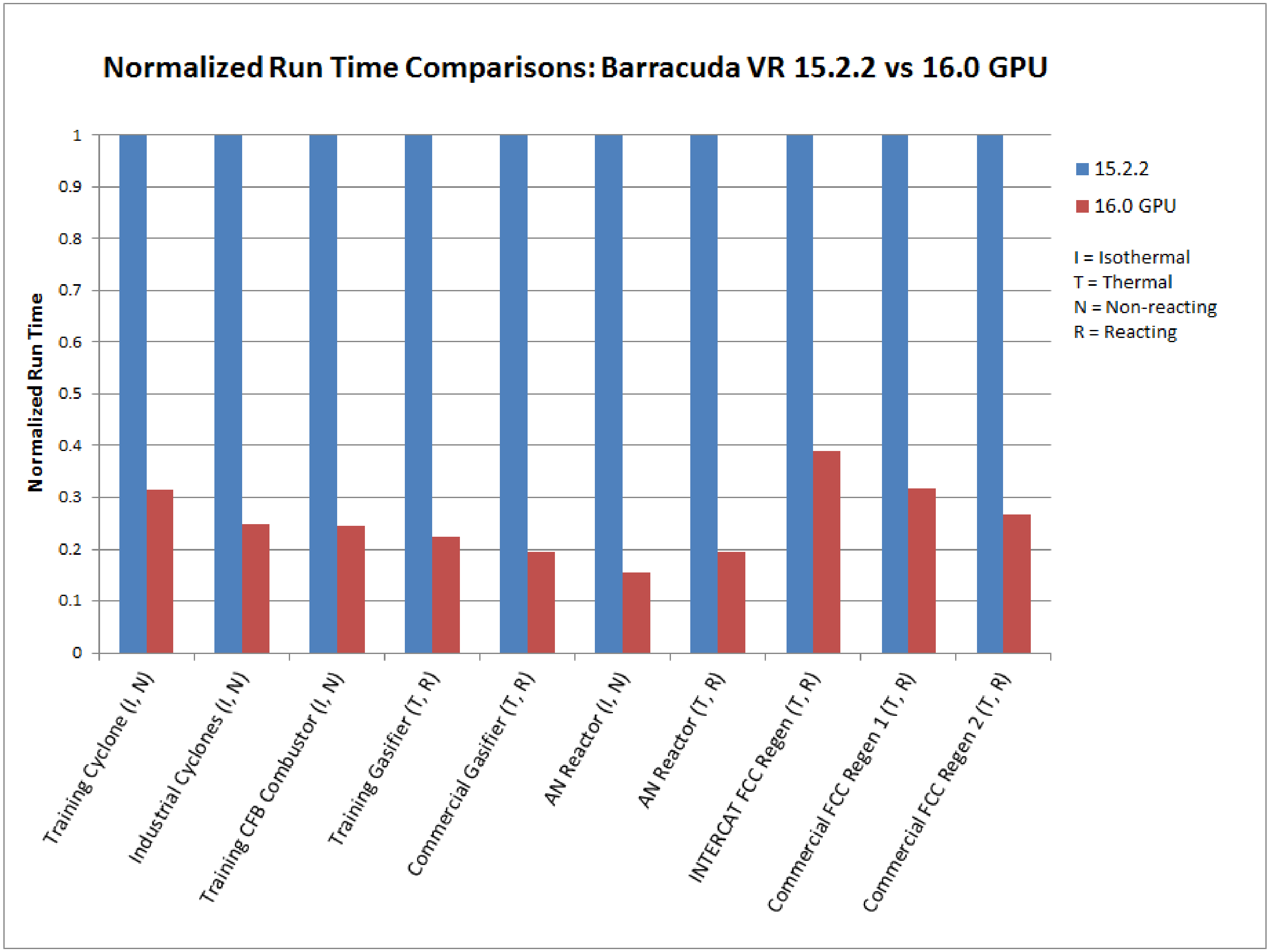
    - The speed-up depends on the speed of both CPU and GPU.

- The plots on the following slides are all based on the same data. The different speed-up descriptions are included here because each is useful in different circumstances.

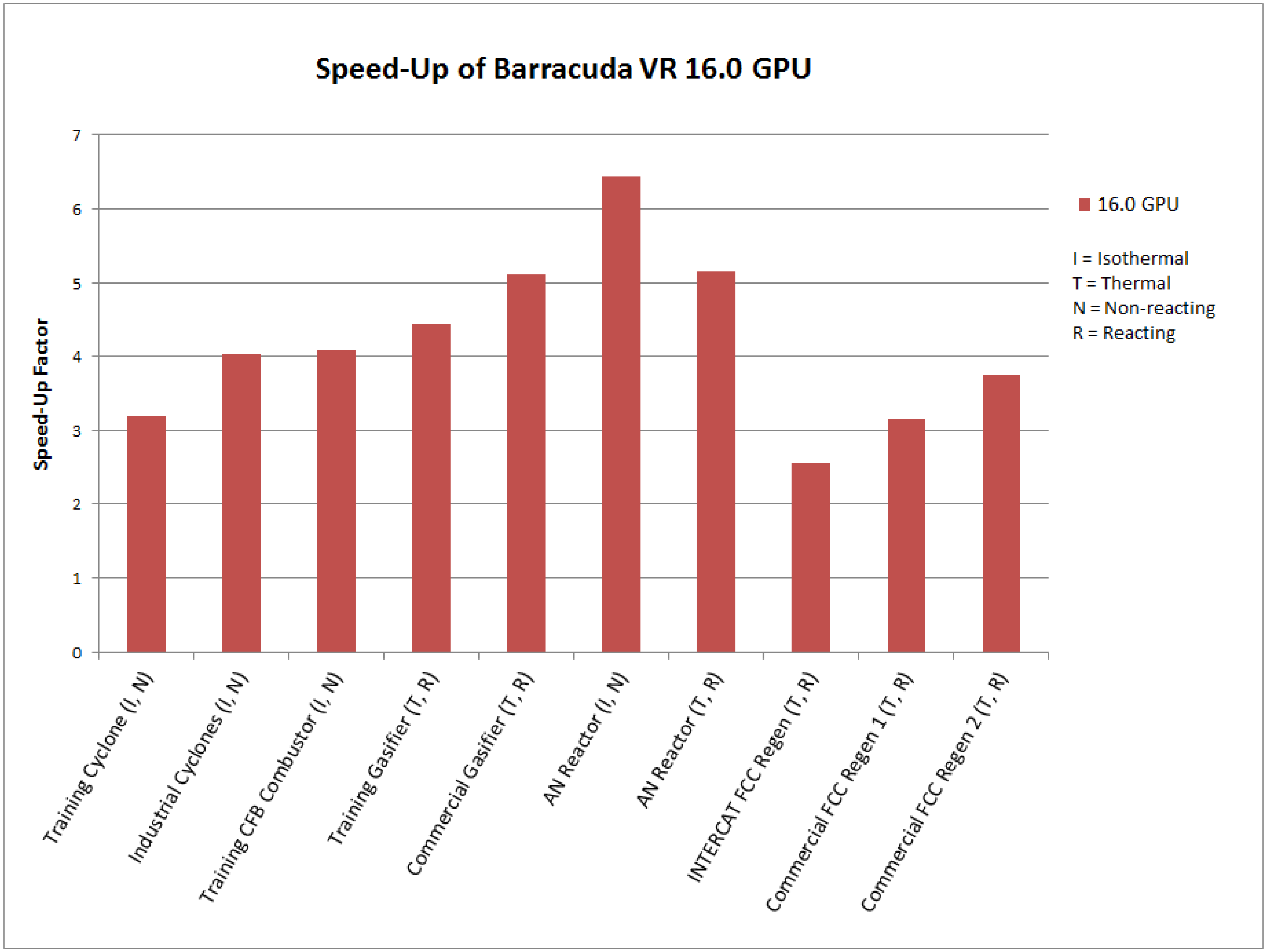
Absolute Run-Time Results



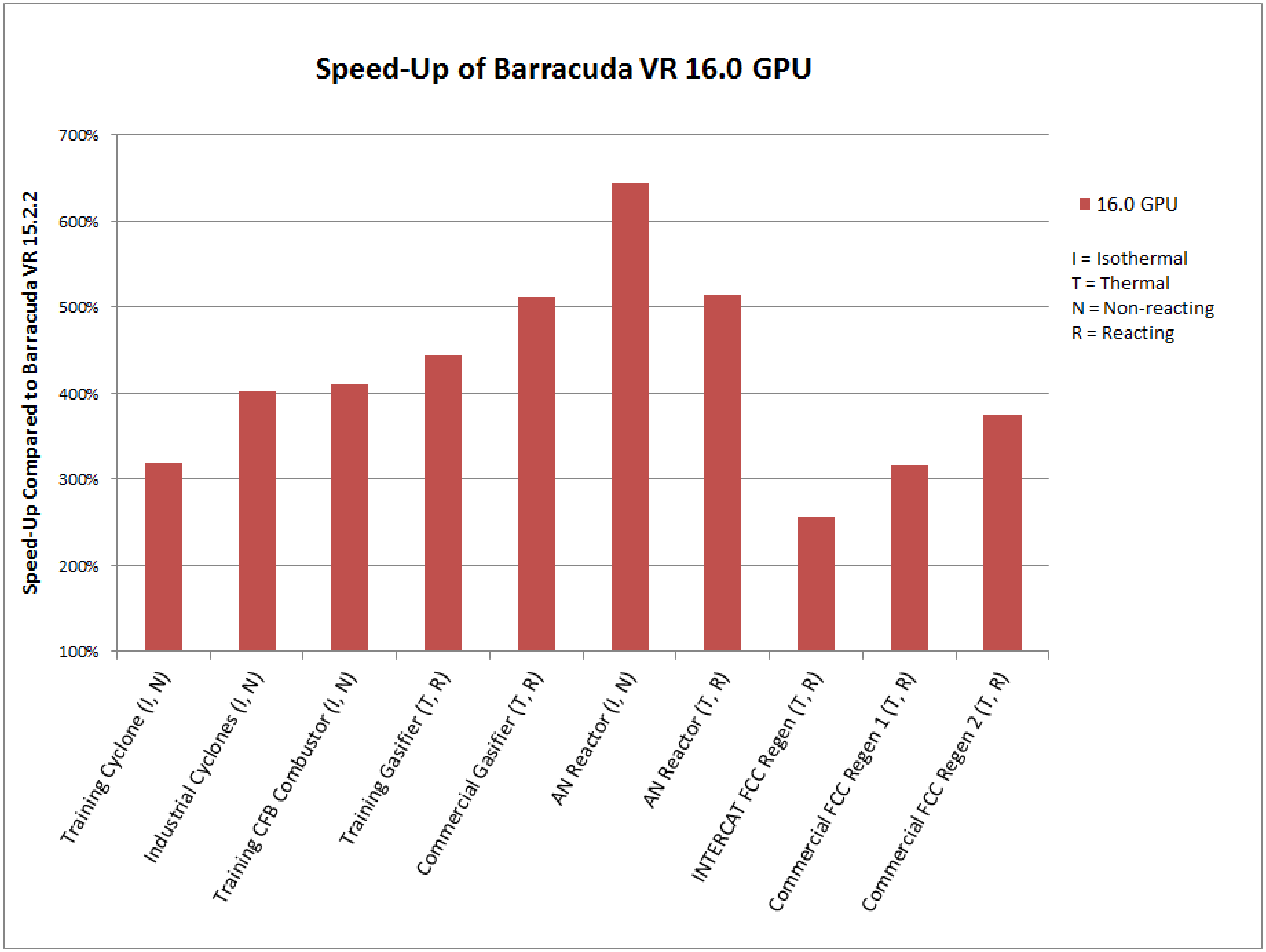
Normalized Run-Time Results



Speed-up Factor



Speed-up Percentage



Controlling GPU Device Usage from the Command-Line

- From the command line, GPU acceleration is enabled by default. The user will be prompted to select a GPU when multiple are available

- The **`-d[0-3]`** flag can be used to run the calculation on a specific device

  `cpfd.x myproject.prj -d2` *(will run on CUDA device with id 2)*

- The **`-dauto`** flag can be used to run the calculation on the first available GPU with enough memory

  `cpfd.x myproject.prj -dauto`

GPU Hardware Considerations

- For the GPU acceleration to be effective, the simulation must fit on the GPU card's on-board memory. For reference, a calculation with 300,000 cells and 7 million particles nearly requires 4 GB of memory for basic calculation (no chemistry, no thermal). If GPU memory is exhausted during a calculation, Barracuda VR 16-GPU performance will likely be significantly hindered.

- Recommended GPU cards: NVIDIA GeForce GTX TITAN, NVIDIA Tesla K20, or better with CUDA Compute Capability of 3.0 and at least 5 GB of GPU RAM.

- Minimum GPU requirements: NVIDIA GPU with CUDA Compute Capability of 2.0 and 4 GB of GPU RAM. NVIDIA driver is required for GPU acceleration. CPFD has prepared an NVIDIA Driver Install Guide to assist customers.

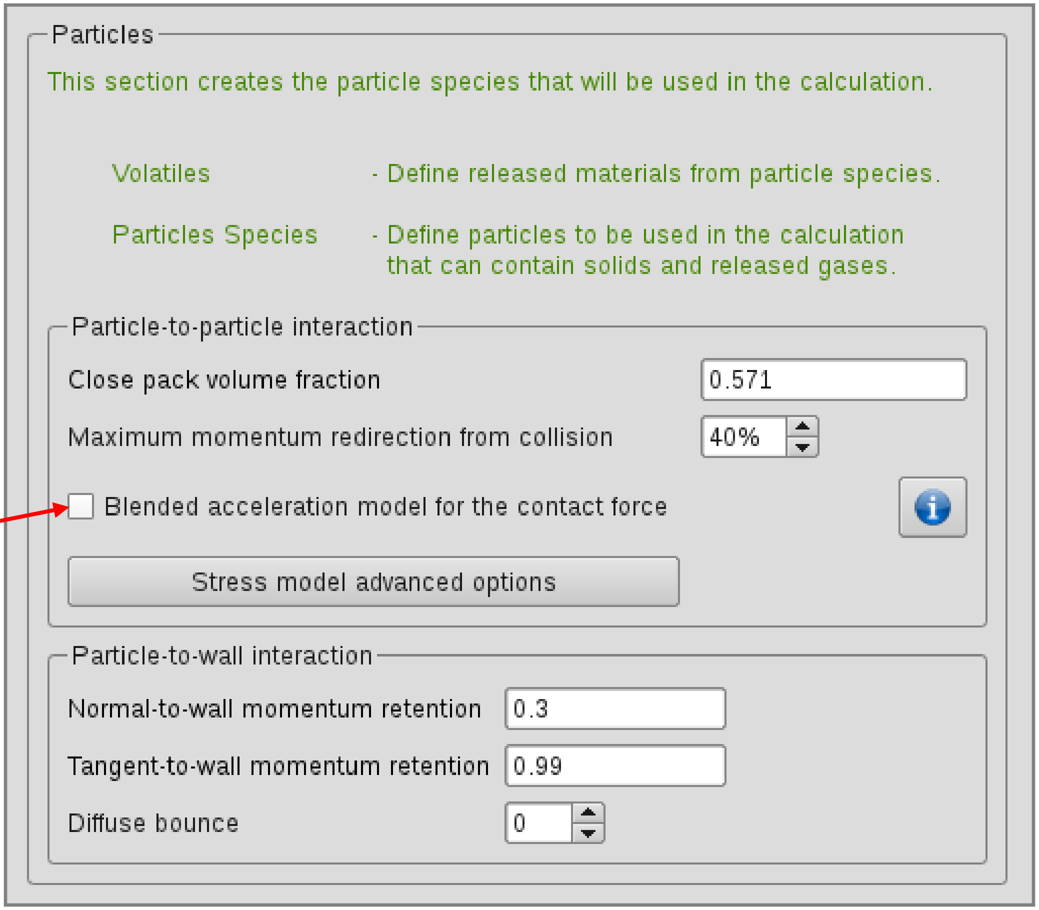
- CPFD offers turn-key workstations fully configured for Barracuda VR 16-GPU with GPU acceleration. Contact sales@cpfd-software.com for more details.

Multiple Calculations and GPUs

- Barracuda VR 16-GPU localizes calculations to a single GPU. i.e., a single calculation will not span multiple GPUs.
- The execution of multiple calculations on separate GPUs is <u>recommended</u> for users on a computer with multiple GPU cards. The increase in calculation time is similar to multiple serial calculations in this case.
- While it is possible to run multiple calculations on the same GPU, it is <u>not recommended</u> as both simulations will be significantly slower.

Blended Acceleration Model for Interpenetration Contact Force

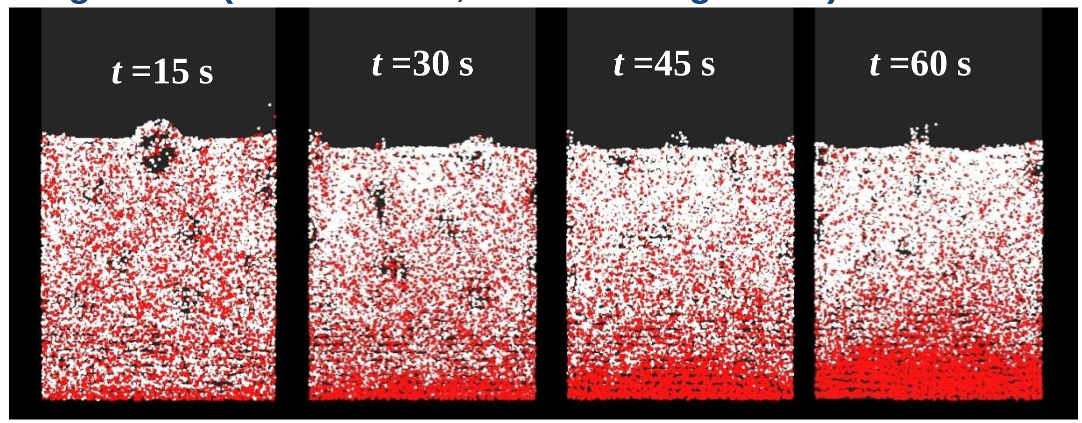
- Barracuda VR 16.0 now includes a blended acceleration model for interpenetration contact force.
- This model better captures restrictions to particle segregation that naturally result from particle-particle contacting in a bed of closely spaced particles.
- The model applies to segregation due to differences in both particle size and particle density.
- This model can be enabled on the **Particles** window.

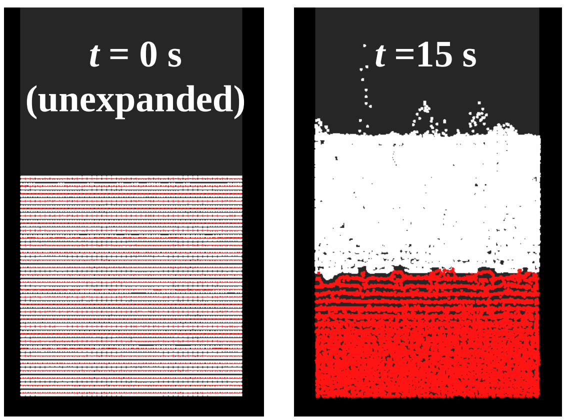


Use of Blended Acceleration Model

A simulation of particle mixing in a pseudo 2D bed shows a qualitatively better segregation behavior than observed without the model

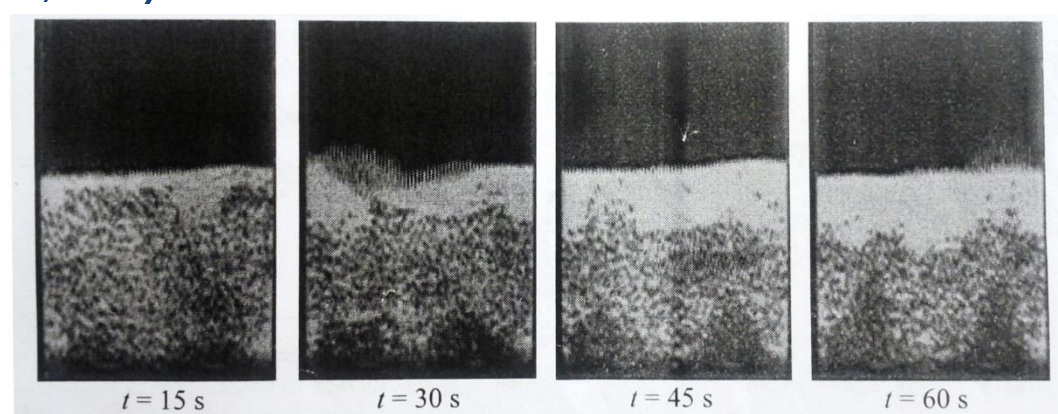
Barracuda VR 16.0 blended acceleration & Koch-Hill-Ladd drag model (user defined, see following slides)





When blended acceleration model is turned off, excessive segregation is observed

Experimental operation of pseudo 2D bed (Goldschmidt et. al, 2003)



Experimental results: Goldschmidt, M., Link, J., Mellema, S., and Kuipers, J. (2003) Digital image analysis measurements of bed expansion and segregation dynamics in dense gas-fluidised beds. *Powder Technology*, 138:135-139.

User Defined Drag Models

- Users can specify their own drag model as a function of **Reynolds number**, **fluid** and **particle volume fractions**, **particle diameter**, **Sauter mean diameter**, **fluid viscosity**, **relative particle velocity**, **fluid density**, **particle density**, and **particle sphericity**.

- The user-defined drag model consists of a custom expression which is multiplied by Stokes drag. Normalizing a drag model by the Stokes drag allows all drag models to be specified accurately and succinctly.

$$\vec{F}_{\text{drag}} = \vec{F}_{\text{Stokes}} \times \underbrace{f_{\text{custom}}}_{\text{User-defined expression}} \qquad \vec{F}_{\text{Stokes}} = 3\pi\mu_f d_p \left(\vec{u}_p - \vec{u}_f\right)$$

- The user-defined multiplier is a dimensionless expression that can be as complex as needed and may contain variables, constants, functions, and if-statements.

- Both System and User-defined drag models are optimized to take advantage of GPU parallelization.

Drag Model Expression Input

- **Available functions**
  - ABS(val1)
  - COS(val1)
  - EXP(val1)
  - IF(condExpr, valTrue, valFalse)
  - LN(val1)
  - LOG(val1)
  - MAX(val1, val2)
  - MIN(val1, val2)
  - SIN(val1)

- **Available Variables**

  Re: Reynold's number
  volfracF: fluid volume fraction
  volfracP: particle volume fraction
  sphericityP: particle sphericity
  densityP: particle density
  densityF: fluid density
  viscF: fluid viscosity
  dVelPF: magnitude of particle velocity relative to fluid velocity

- **Available Operators**

  + - * / ^
  < > <= >= ==
  && ||

- IF statements are written in a format similar to that used in Microsoft® Excel:
  IF(conditional expression, value if true, value if false)

- For example, IF statement is used to create a Wen-Yu drag model expression:
  IF(Re<1000,1+0.15*Re^0.687,0.44*Re/24)*volfracF^-2.65

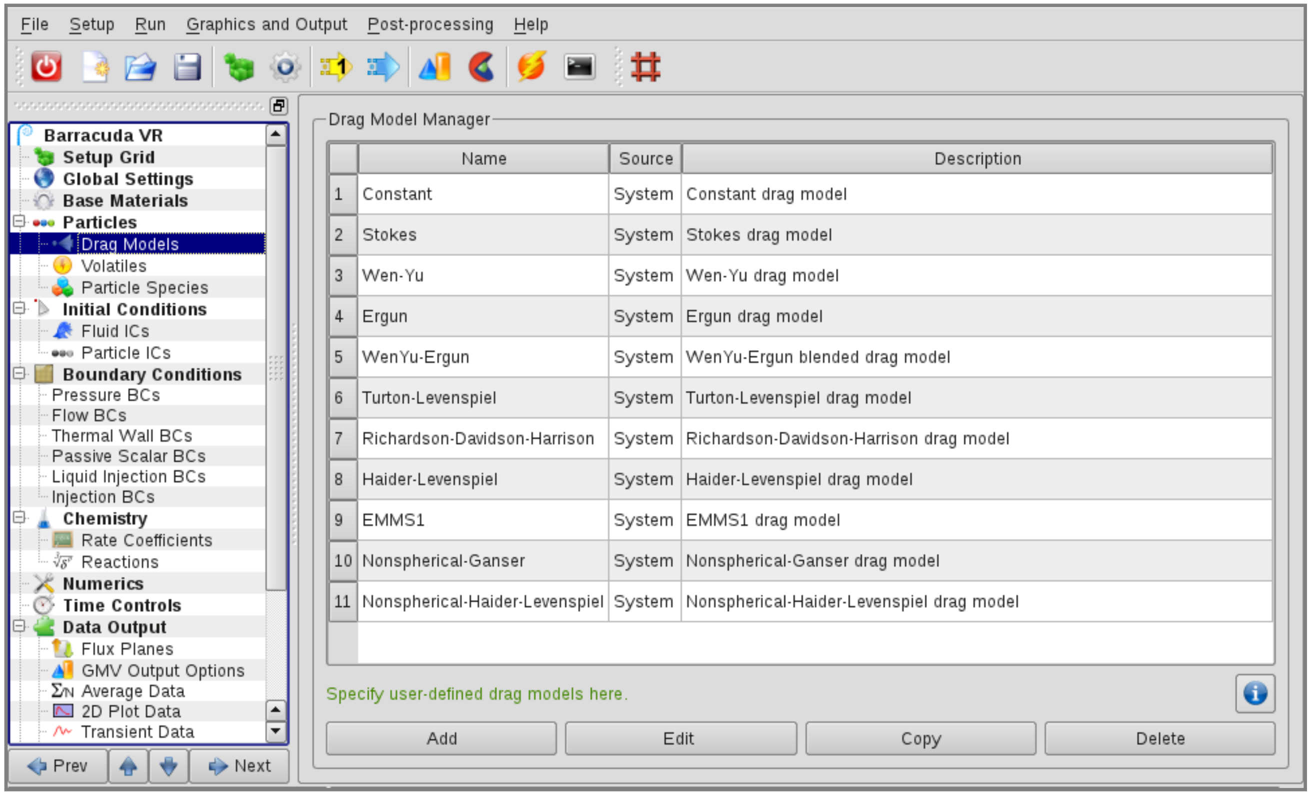
  conditional **if true** **if false**

Managing Drag Models

- Drag model manager is a new item in the GUI tree, which allows the user to:
  - View built-in “System” drag models
  - Create user-defined drag models

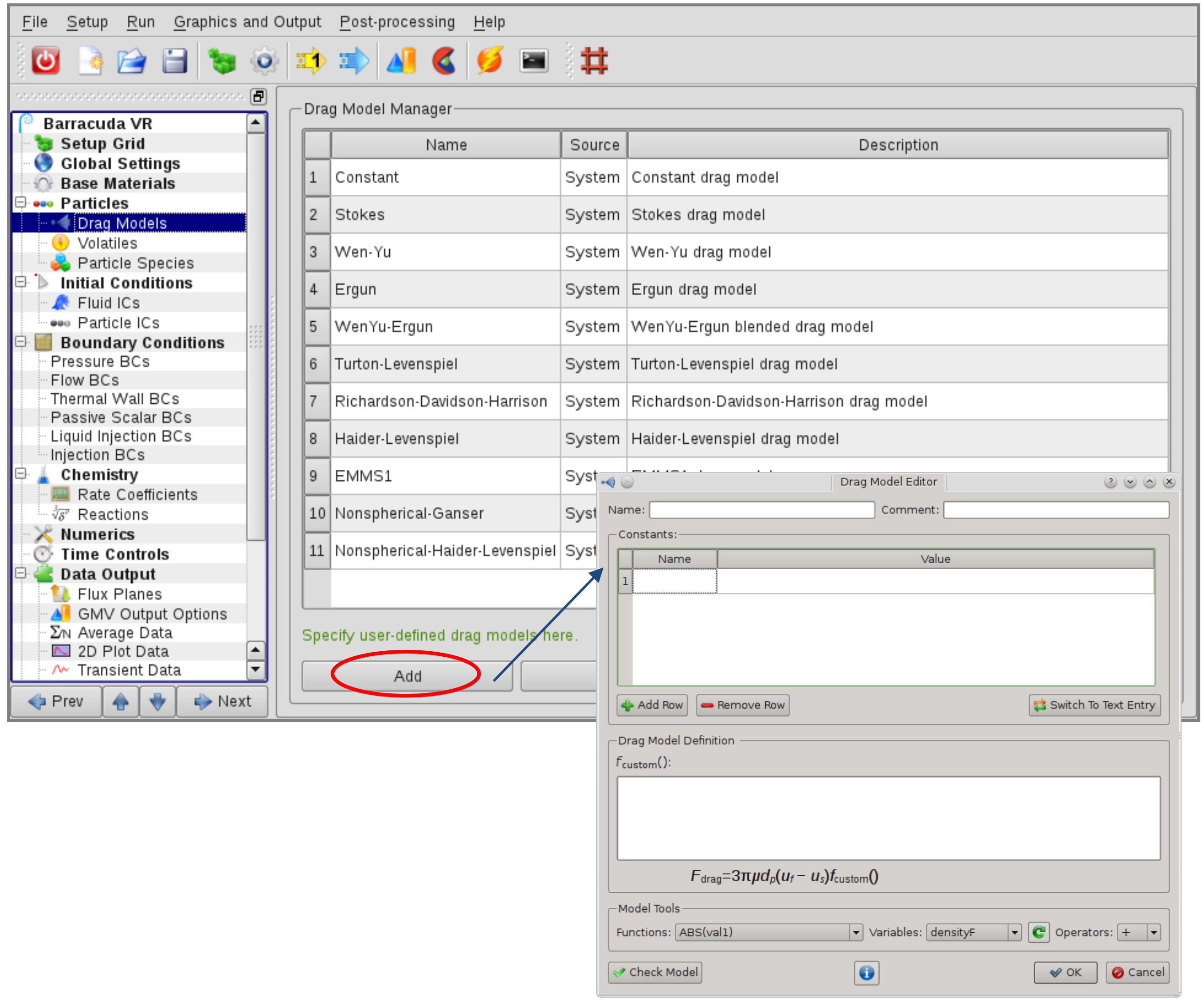
- As in previous versions of Barracuda, drag models are applied to a particle species in the **Particle Species** window.

- “System” drag models can be copied and then edited as user-defined drag models. This provides a convenient way to define a custom drag model that is based on an existing “System” drag model.



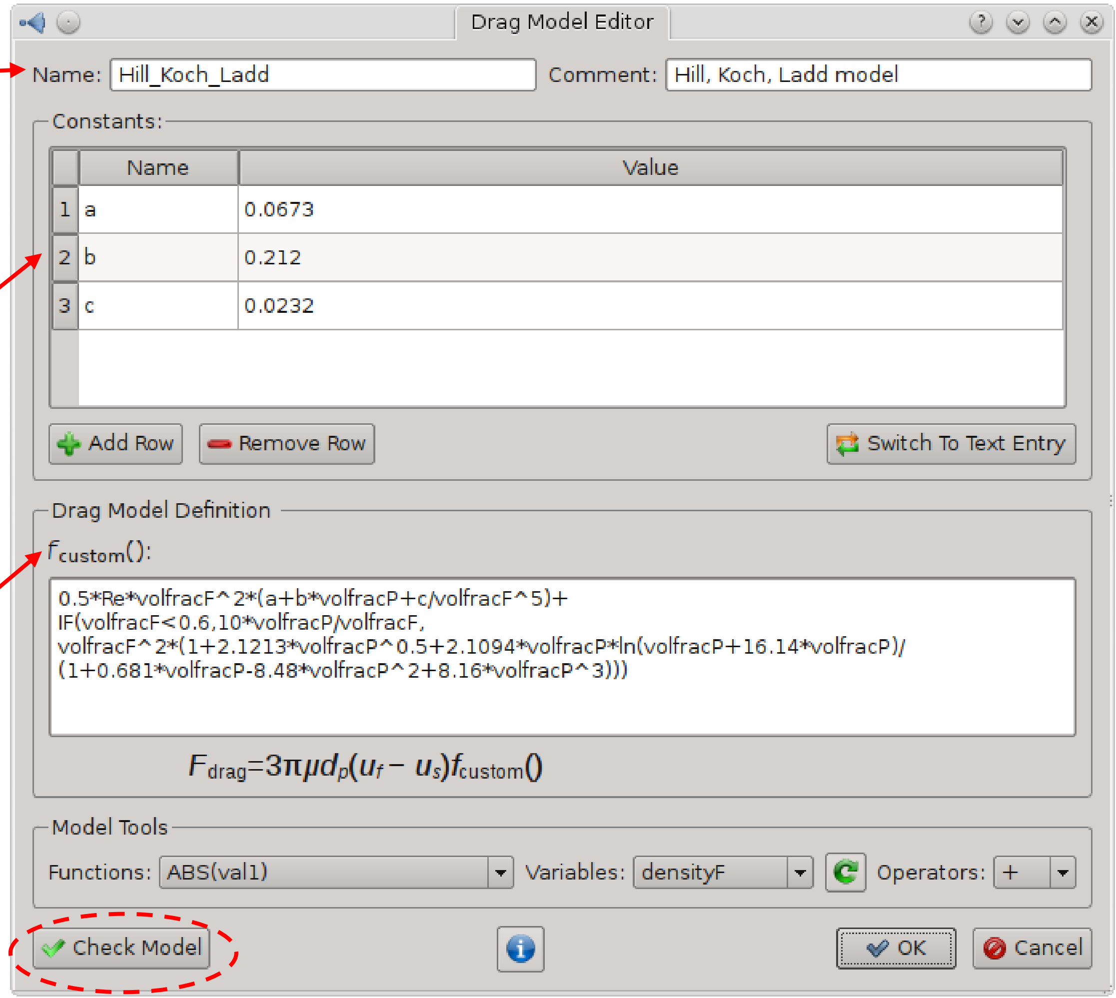
Adding Drag Models

- Users may define their own drag model by clicking **Add**, which displays the drag model editor window.
- This dialog provides the interface for creating a custom drag model.



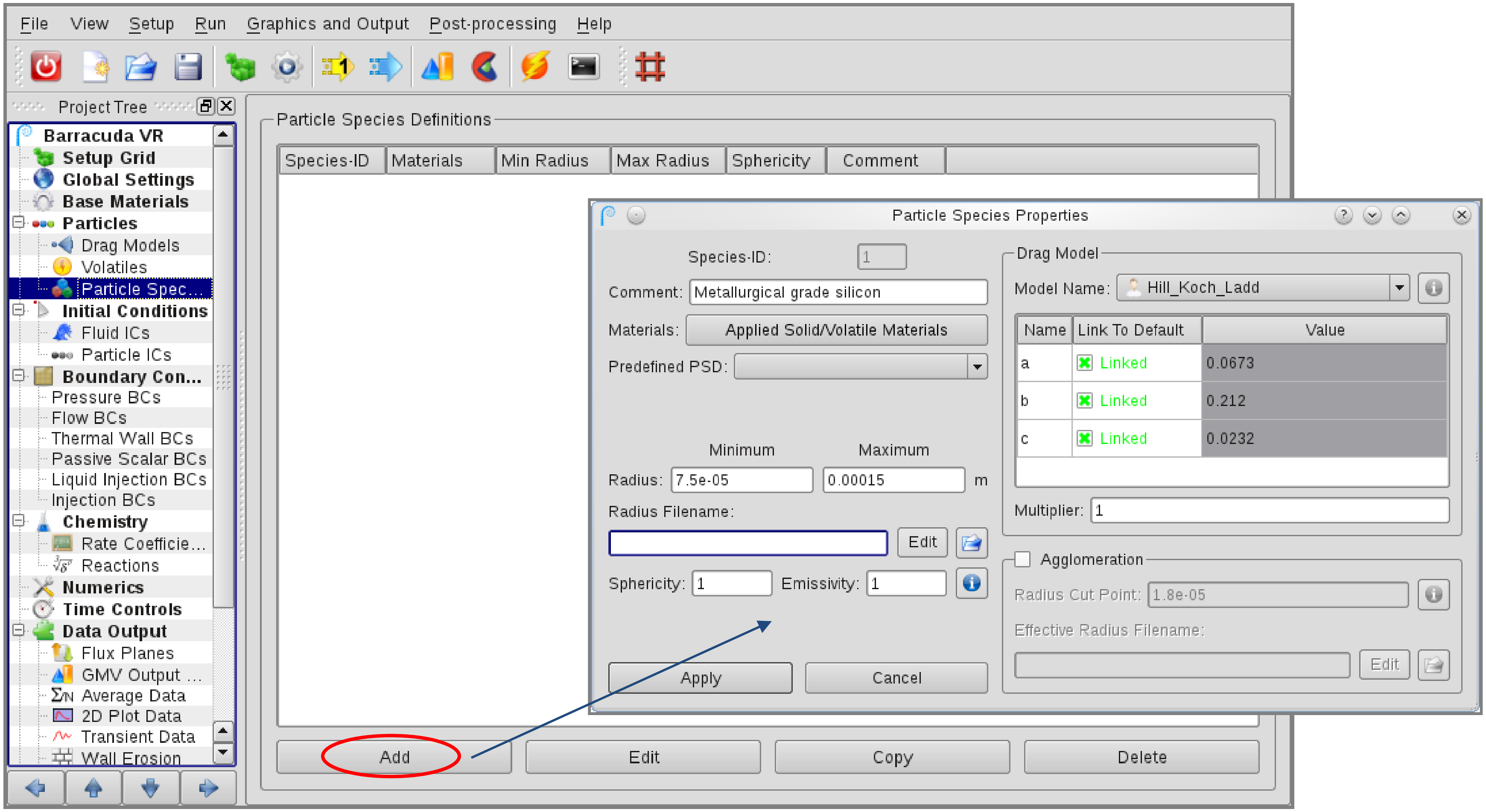
Drag Model Editor Window

- A new drag model must be given an unique name, which will be used to apply the model to a particle species.
- Constants can be defined and used in the drag model expression.
- The drag model expression is entered in the f_{custom}() field and once complete can be verified by clicking on **Check Model.**



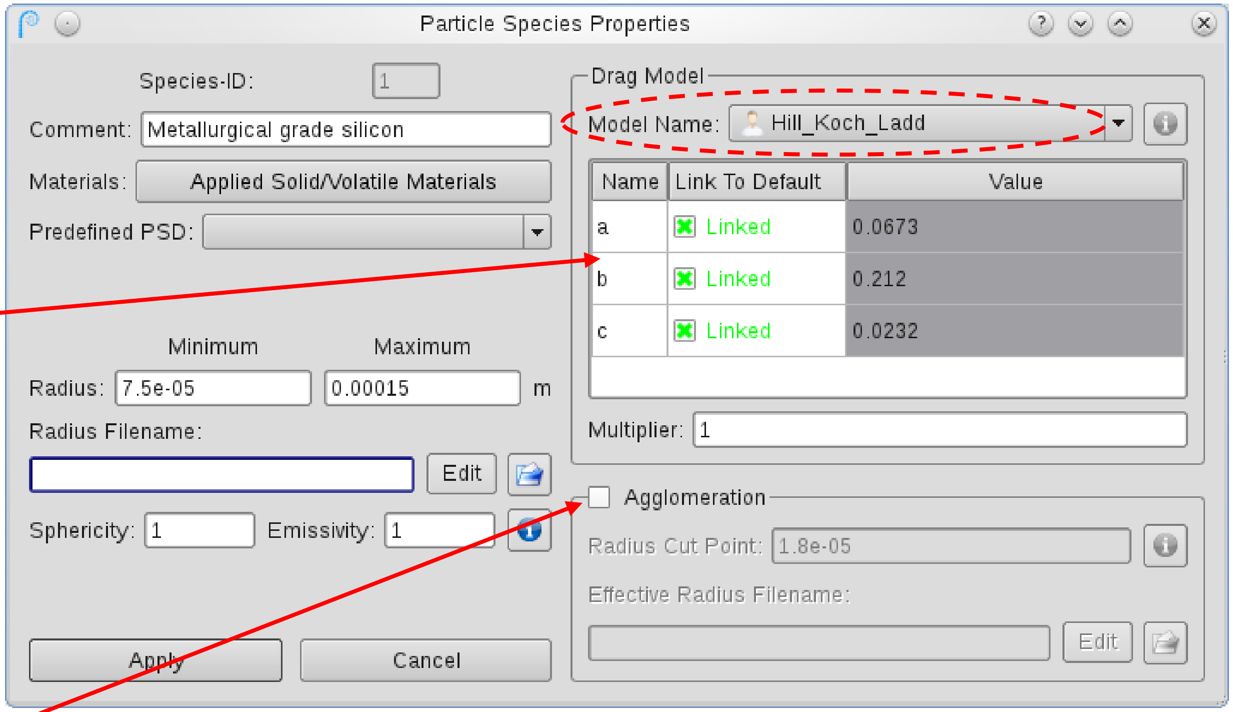
New Particle Species Editor

- Barracuda VR 16.0 contains an improved particle species dialog.



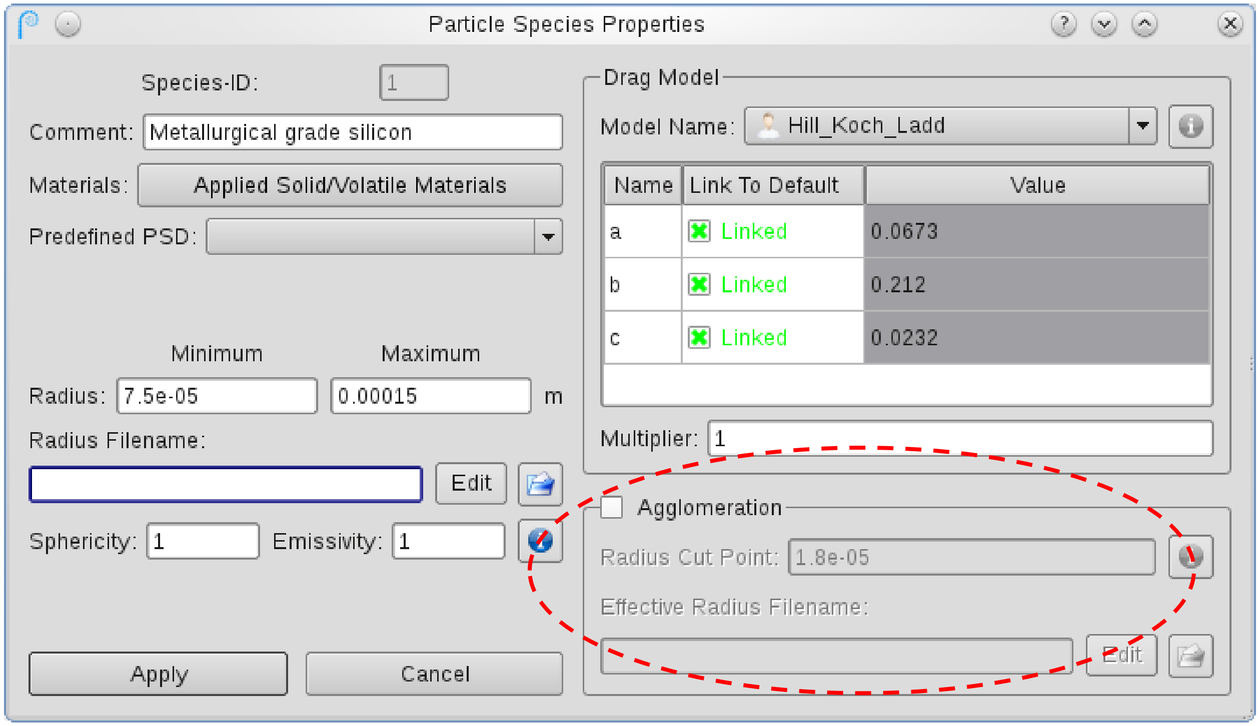
Applying a Drag Model

- Drag models are applied to individual particle species and drag model constants can be viewed and edited.
- Any changes to constants apply only to the current particle species.
- Definitions of constants can be viewed in the drag model manager.
- Agglomeration models can be applied to any drag model including user defined models.



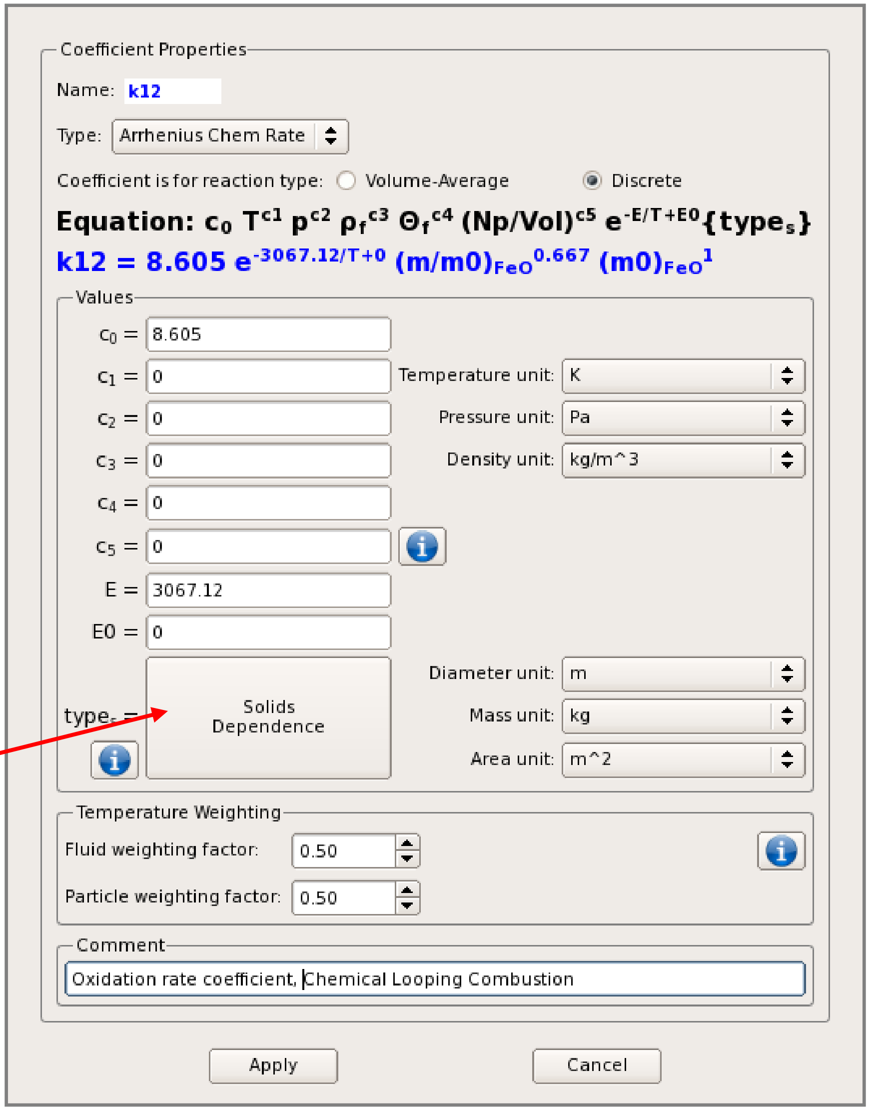
Applying Agglomeration Model

- An agglomeration model can be applied to particle species for **any** drag model, including user defined models.
- Interface on particle species dialog allows:
  - Agglomeration to be enabled
  - Specification of radius cut point
  - Creation or editing of an effective radius file

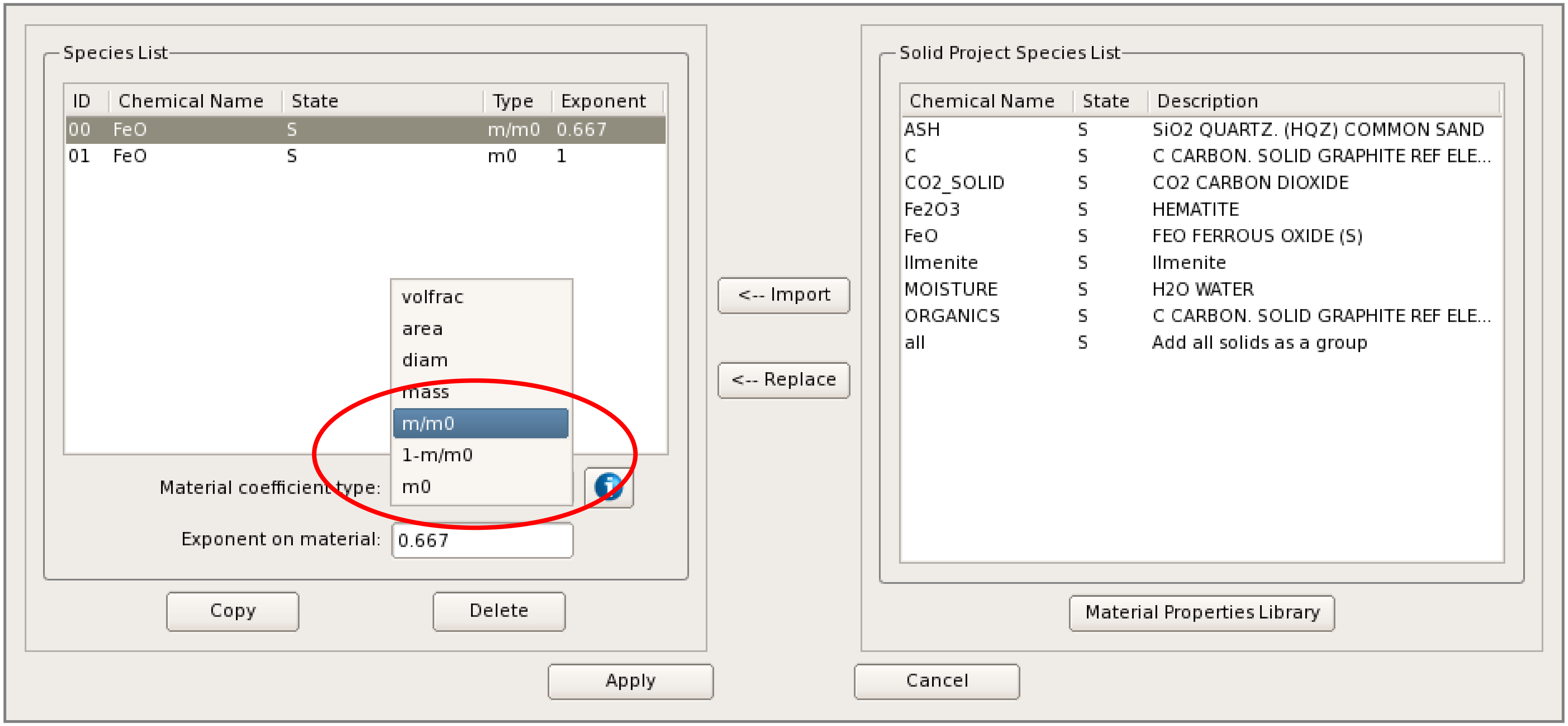


Conversion Terms in Chemistry

- Barracuda VR 16 now includes new solids dependence terms for ease of use when adding chemical reactions based on conversion.
- $\mathbf{m_0}$ is the initial mass of a particle *or* particle component.
- **m** is the current mass of a particle or particle component.
- New solids dependence terms are included in **Solids Dependence** Dialog:
  - m/m0
  - 1- m/m0
  - m0



Conversion Terms in Chemistry



Using Conversion Terms

- Kinetics of ilmenite oxidation reactions use conversion (X) in definition:

$$4\mathrm{FeO} + \mathrm{O_2} \rightarrow 2\mathrm{Fe_2O_3}$$

$$\frac{dX_o}{dt} = (1 - X_o)^{2/3} \frac{3bk_0}{\rho_m r_g} \exp(-E/RT)[\mathrm{O_2}] \quad \text{where} \quad X_o = 1 - \frac{m_{\mathrm{FeO}}}{m_{0,\mathrm{FeO}}}$$

- Substituting conversion definition and rearranging, yields a form that is readily entered in Barracuda VR 16.0 using new solids terms ***m0*** and ***m/m0***

$$4\mathrm{FeO} + \mathrm{O_2} \rightarrow 2\mathrm{Fe_2O_3}$$

$$\frac{d\,m_{\mathrm{FeO}}}{dt} = \underbrace{m_{0,\mathrm{FeO}} \left(\frac{m_{\mathrm{FeO}}}{m_{0,\mathrm{FeO}}} \right)^{2/3} \frac{3bk_0}{\rho_m r_g} \exp(-E/RT)[\mathrm{O_2}]}_{\text{Rate coefficient}}$$

Rate coefficient

Particle Age Factor

- Models may require the feed or initialization of particles that are have already undergone a reaction. Therefore, the composition used for particle initialization would be *different* from the initial particle composition used for calculating conversion in reactions.

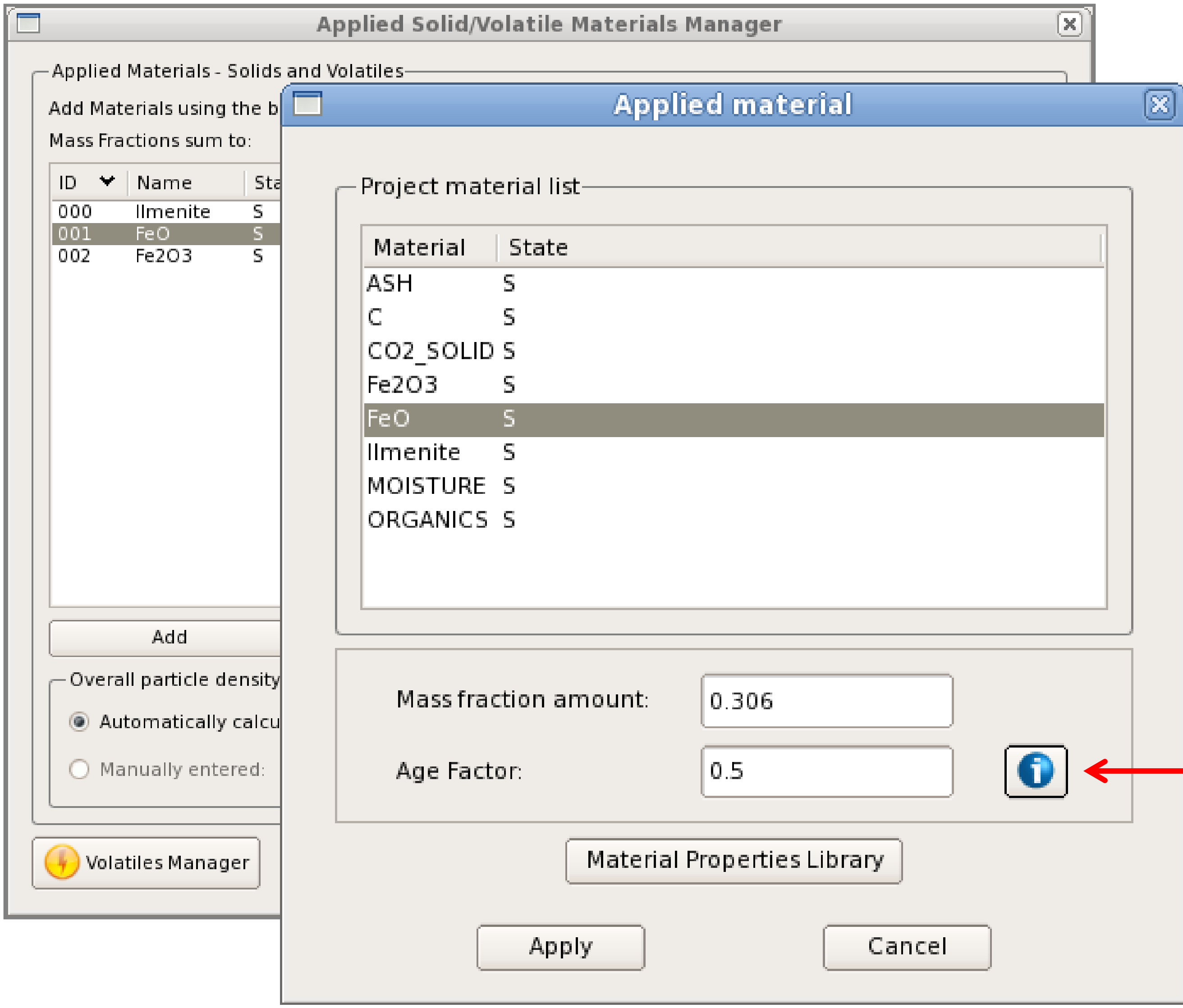
- Barracuda VR 16.0 now contains an **Age Factor** in the particle species dialog to relate the *feed* mass of particle components to the *initial* mass of particle components:

Initial mass of a component = Age factor X Feed mass of a component

- **Feed composition**: the composition of a particle when it enters the model domain.

- **Initial composition**: the composition of a particle when it is considered new or "fresh" for reaction chemistry calculations.

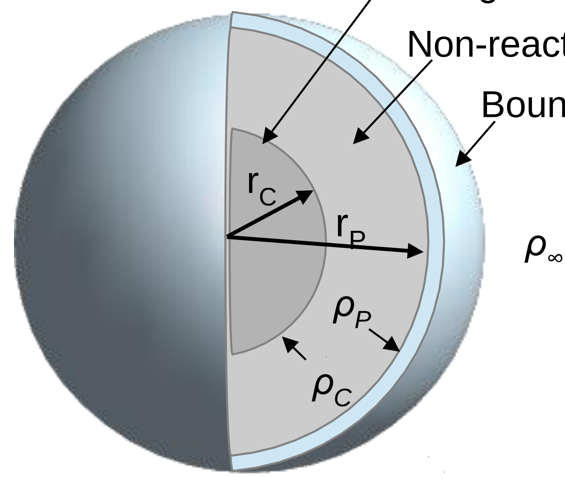
Particle Age Factor



Age factor can have values in the range: 0 < Age Factor <= 1

Barracuda Shrinking Core Reaction model

- Barracuda VR 16.0 contains a new Shrinking Core Model for more accurate modeling of some types of reactions. For example, analysis of partially reacted carbon particles show an ash region surrounding an unreacted core (Yagi and Kunii, 1955; Levenspiel,1972).
- The Barracuda shrinking-core-model is on a per particle basis and assumes that the solid material in a particle reacts in the presence of a gas species. The rate of reaction is controlled by :
  - the first-order reaction rate;
  - the transport of the gas through the non-reacting material to the core; and
  - the transport of gas through the boundary layer



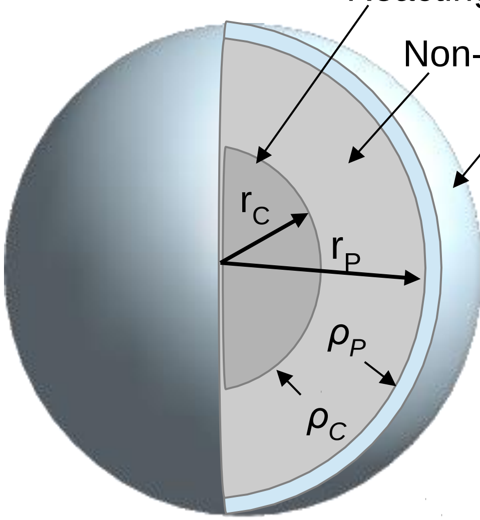
- Each particle has its own history and a “fresh” particle will have more reaction than an “old” particle.

Levenspiel, 0., 1972, Chemical Reaction Engineering, John Wiley &Sons, New York.

Yagi, S., Kunii, D., 1955. Studies on combustion of carbon particles in flames and fluidized beds. In: Fifth Symposium (International) on Combustion, Reinhold, New York, pp. 231–244

Barracuda Shrinking Core Reaction model

- For the shrinking core model, the gas concentration at the reacting core is



$$\rho_c = \frac{\rho_\infty}{k_R\left(\frac{1}{k_R}+\frac{1}{k_D}+\frac{1}{k_B}\right)}$$

$$k_D = \frac{4\pi D_m}{r_p^{-1} - r_c^{-1}} \qquad k_B = h_m A_p$$

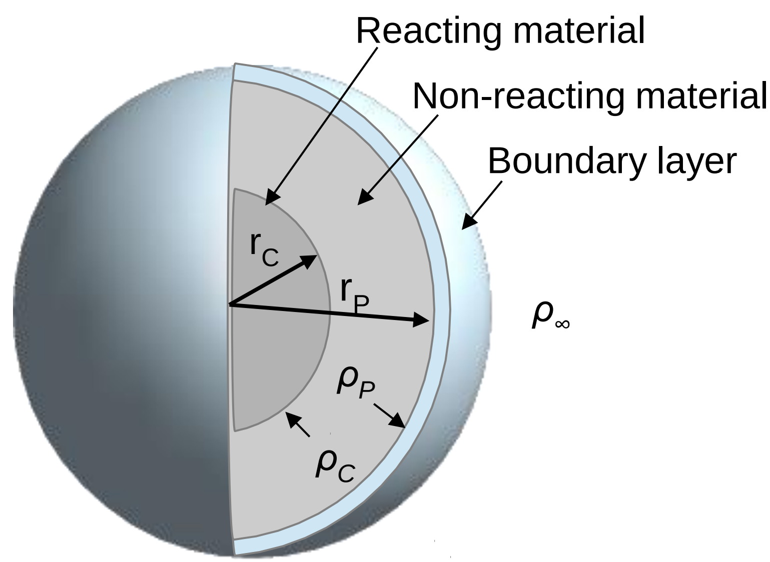
ρ_c	The gas mass concentration at the reacting solid core
ρ_∞	The gas mass concentration in the bulk gas
k_R	The first order reaction rate
k_B	Boundary layer mass transfer coefficient
k_D	Diffusion through non-reacting material

Barracuda Shrinking Core Reaction model

- Parameters in the shrinking core model are

$$r_c = \left[\frac{3m_s}{4\pi \bar{\rho}_s}\right]^{1/3} \qquad Re = \frac{2r_p \rho |u_p - u_f|}{\mu}$$

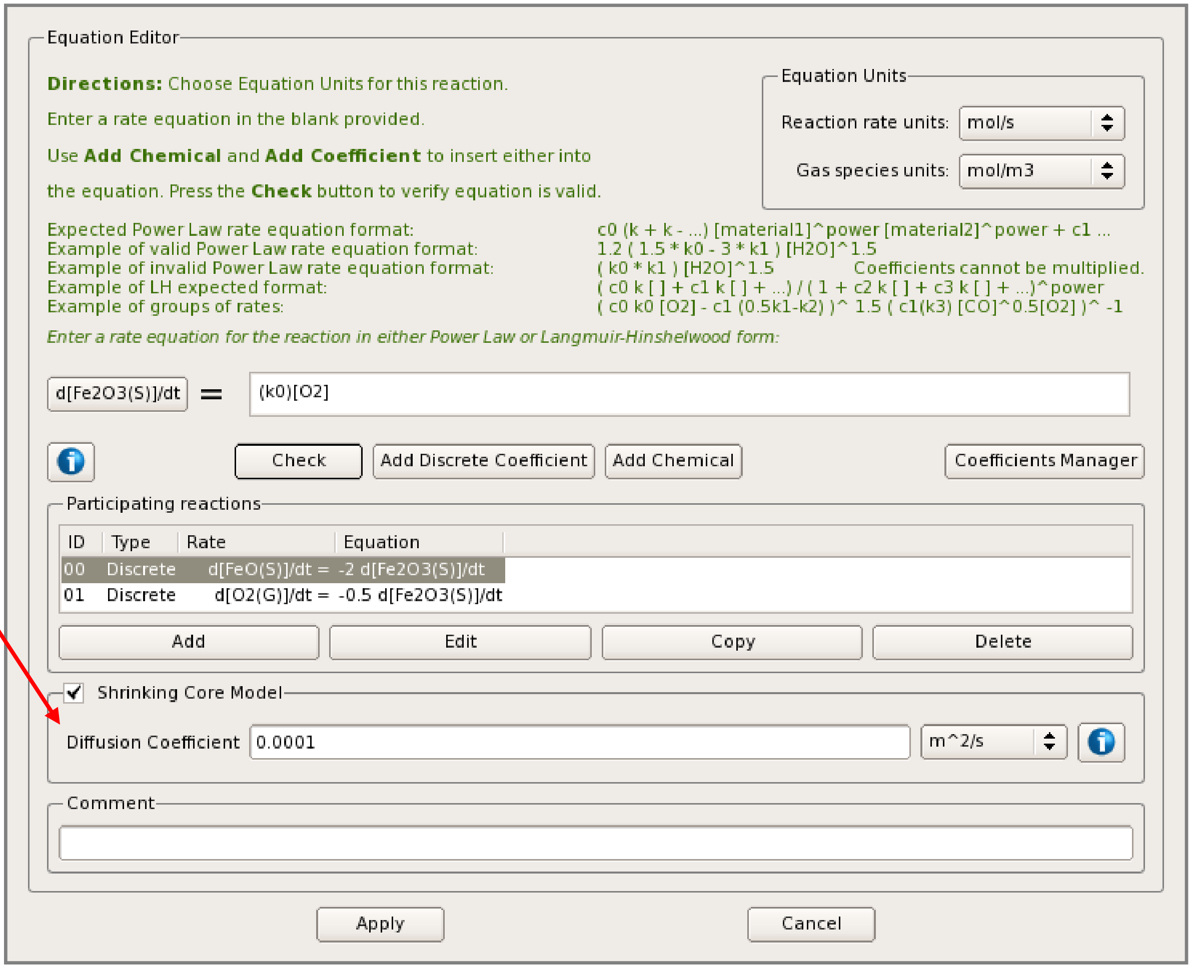
$$\frac{h_m 2r_p}{D_{m,g}} = \mathrm{Sh} = 2 + 0.6\mathrm{Re}^{1/2}\mathrm{Sc}^{1/3} \qquad Sc = \frac{\mu_f}{\rho_f D_{m,g}}$$



r_c	Radius of reacting core
r_p	Radius of particle
ρ_c	Density of reacting material
m_s	Mass of reacting solid in particle
D_m	Mass diffusion coefficient for non-reacting material
Sh	Sherwood number
h_m	Mass transfer coefficient through boundary layer
A_p	Surface area of particle

Using the Shrinking Core Model

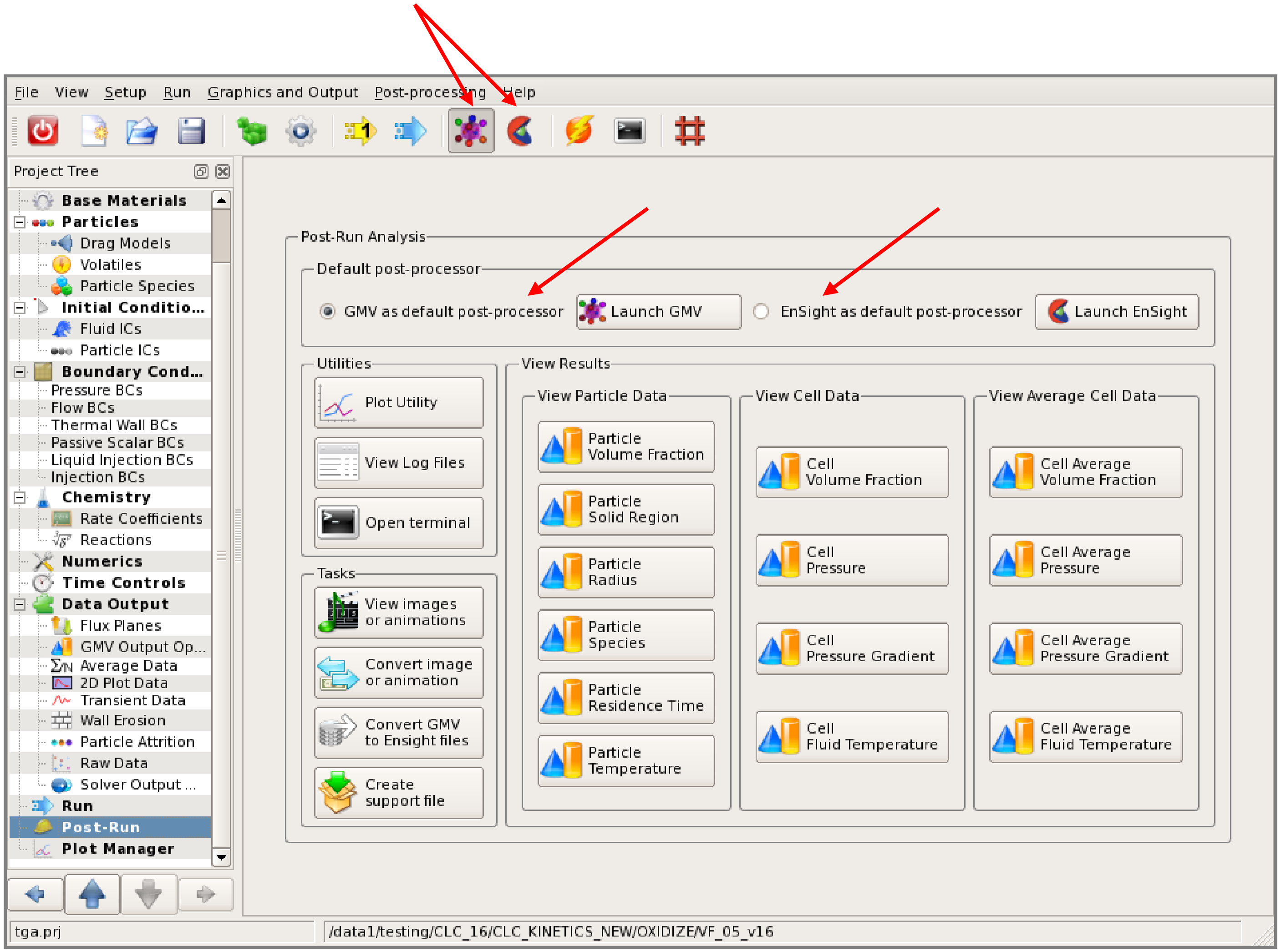
- The *Shrinking Core Model* can be enabled for any discrete particle reaction in the reactions window.
- The diffusion coefficient of the gas through the non-reacting solid must be specified.
- Strictly speaking, the shrinking core analytic solution is for a first order reaction. However, Barracuda does not enforce a first order reaction, and the shrinking core model may be used with any reaction order. It is the user's responsibility to verify that the model is appropriate in such cases.



Post-processor Toggling

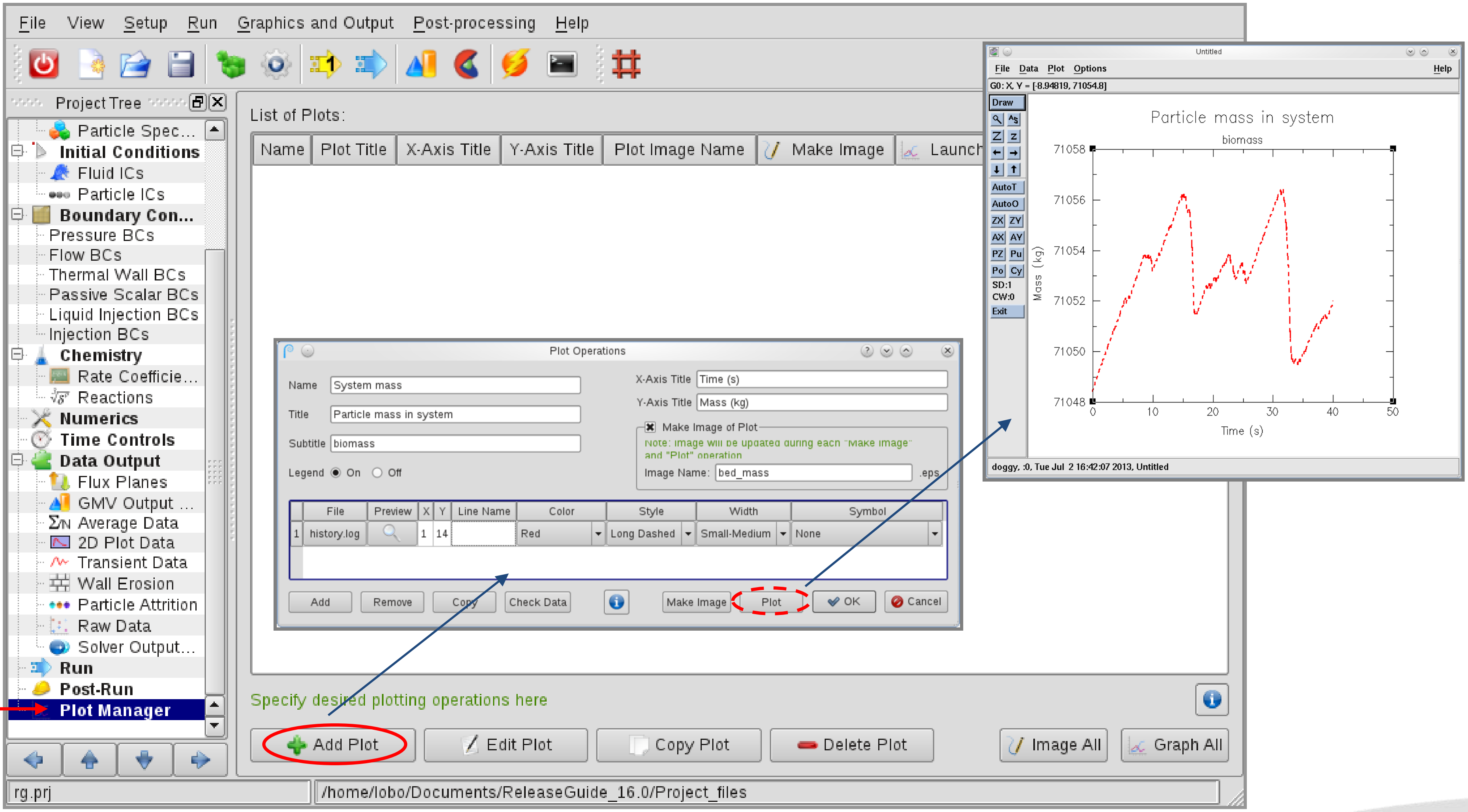
- Toggling between using GMV or Ensight for post-processing is done from the **Post-Run** window.

- Alternatively, the shortcut bar contains a post-processor toggle control as well.

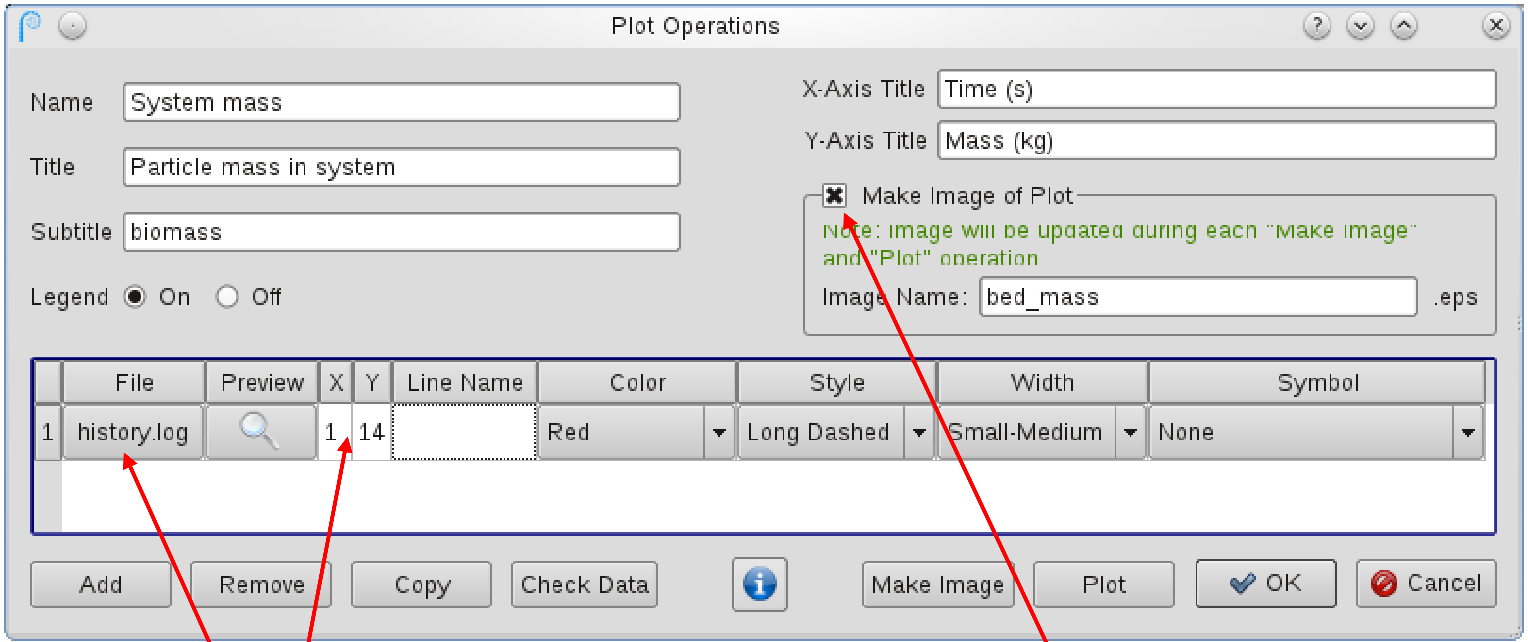


Plot Manager

- The plot manager is a new item in the GUI tree, which provides a convenient interface for creating and managing 2D plots.



Plot Manager

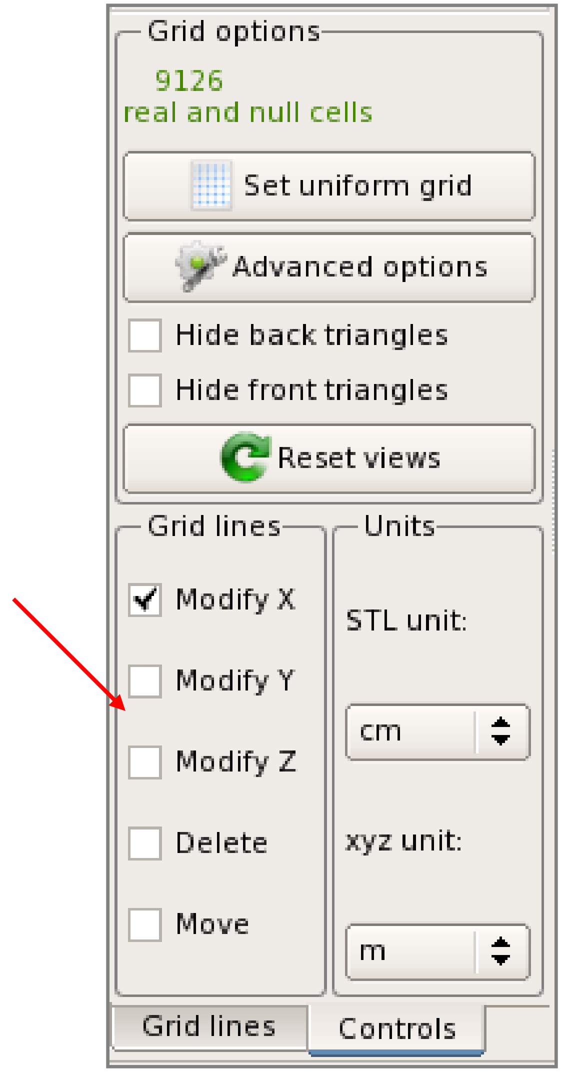


A single plot can have multiple sources of data given by the file name and columns

Make image will automatically save the plot to the name given as an .eps file

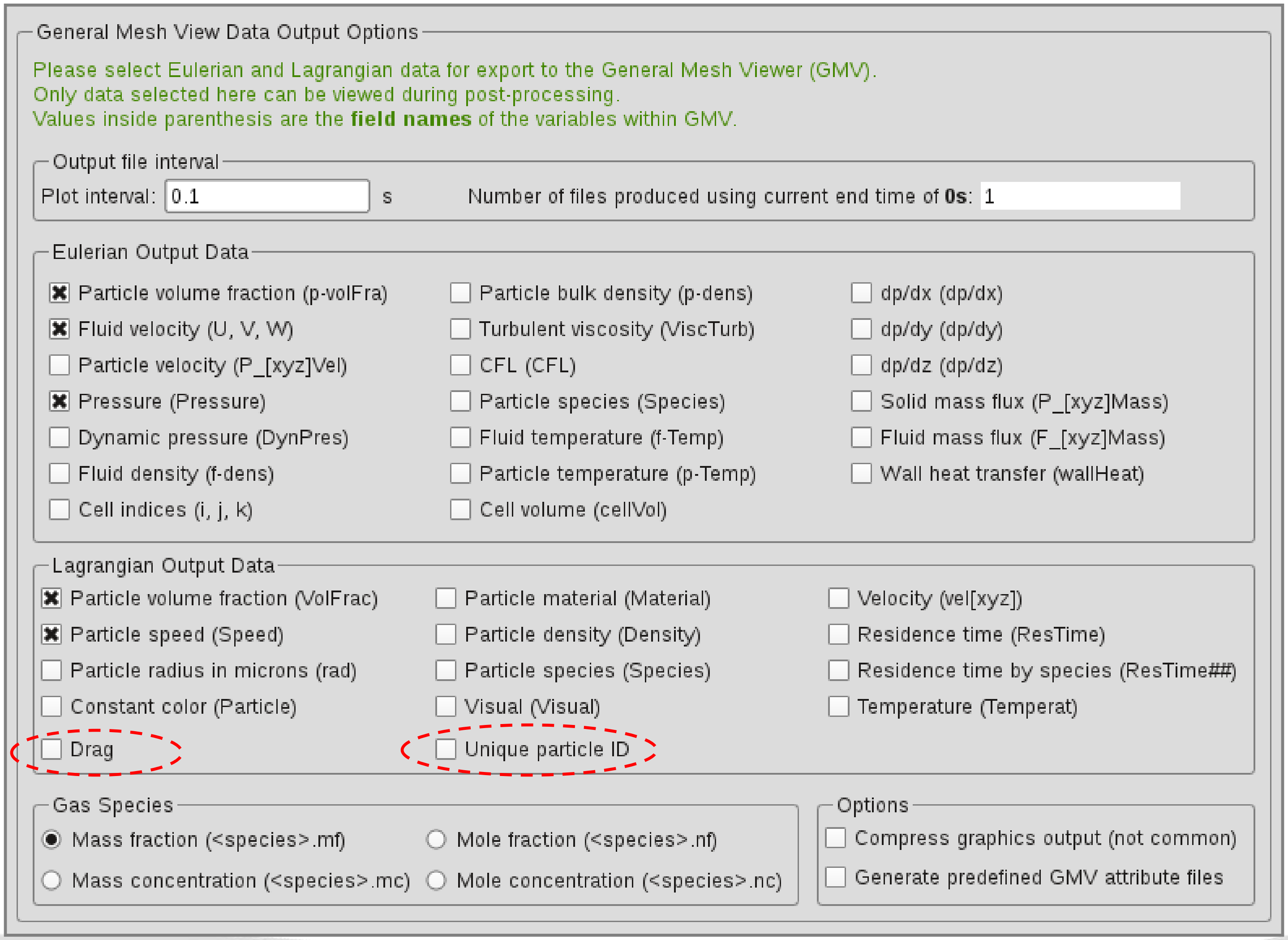
Setup Grid Keyboard Shortcuts

- In Barracuda VR 16.0, quickly switch between gridding tasks using keyboard shortcuts:
  - Modify X = “x”
  - Modify Y = “y”
  - Modify Z = “z”
  - Delete = “d”
  - Move = “m”
  - Use “Escape” to deselect the current mode



New GMV Output Data

Lagrangian variables for **Unique particle ID** and **Drag** have been added



Installation and Support

Please do not hesitate to email or call with installation and support issues

Email: support@cpfd-software.com

Phone: +1-505-275-3849

Coming soon! All new customer support site at www.cpfd-software.com including:

- Knowledge base articles
- Barracuda VR Users Manual
- Downloads