NanoSim

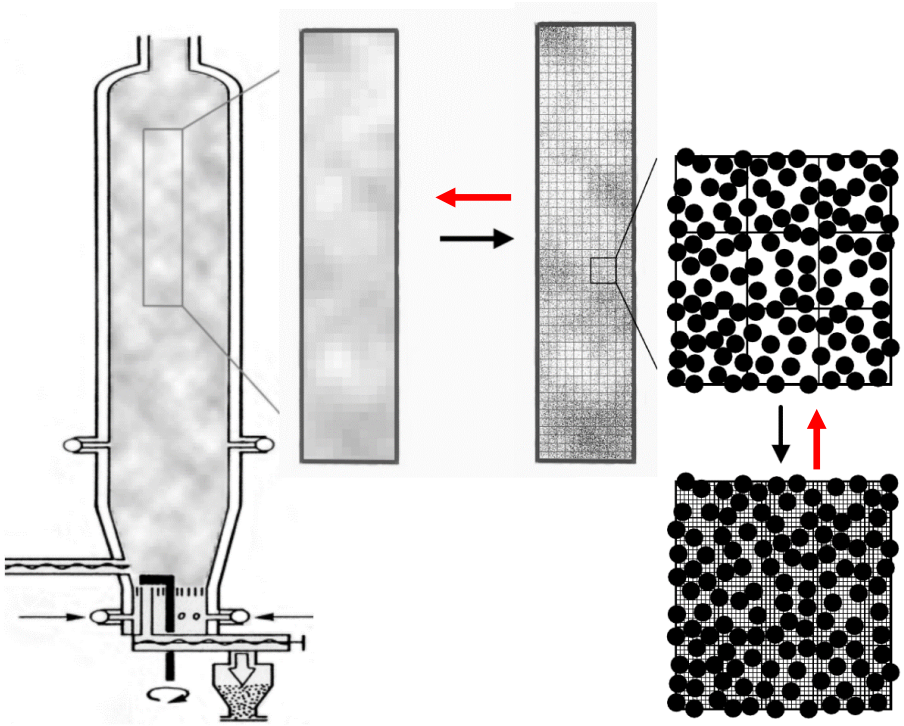
Architecture and application of the data filtering library ”CPPPO” to transport phenomena in dense gas-particle flows

TU Graz, Institute of Process and Particle Engineering , Graz, Austria

Federico Municchi,
Stefan Radl

July 5 2016, PDESoft, Warwick, UK

Multi-scale approach and coarse graining



To solve the transport equations on (affordable) coarse grids we need to take into account transport phenomena occurring at sub-grid scales by mean of closure models («**material relations**»[1])

Closure models can be derived by **filtering «resolved» simulations**

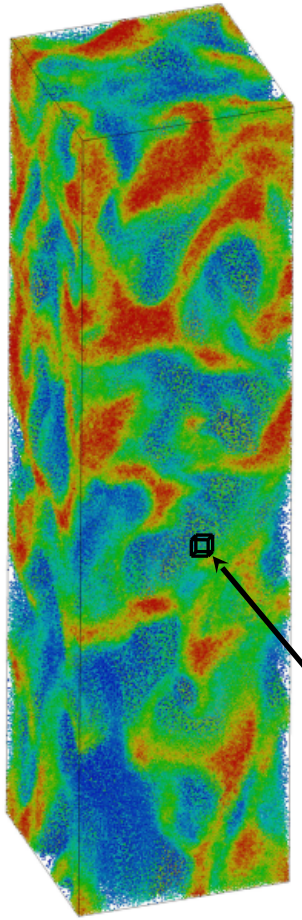
Van Der Hoef et al. 2006, Multiscale modeling of Gas-Fluidized beds, Advances in chemical engineering.

[1]https://ec.europa.eu/research/industrial_technologies/pdf/review_of_materials_modelling_iv.pdf

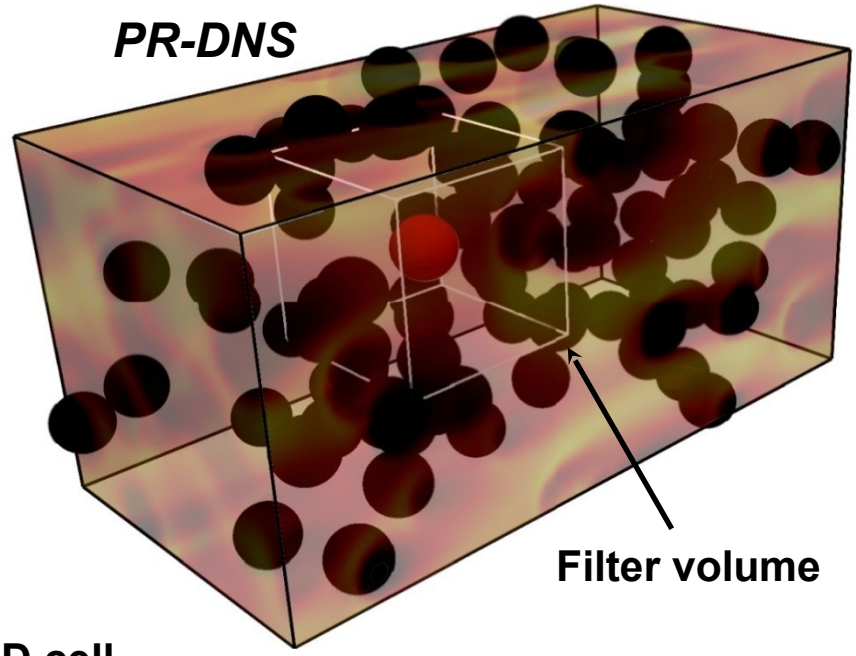
Coarse graining

Euler-Lagrange

Filtering operations should represent the **implicit filtering** due to grid coarsening.



PR-DNS



The CFD cell contains the **same amount of information** of the filter volume.

Filter volume ↔ CFD cell

Radl et al., ECCOMAS 2012

What is CPPPO?

- CPPPO is an **open-source** C++ library of **parallel** data processing functions.
- CPPPO is a tool for "**offline scale bridging**", i.e., developing closures for coarse mesh models by **filtering** fine mesh data.
- CPPPO is a **stand-alone** library that can be **linked** to almost any simulator.
- CPPPO allows **sampling** an **binning** of simulation data "**on-the-fly**".
- Input specified using Json files, **no coding required**.

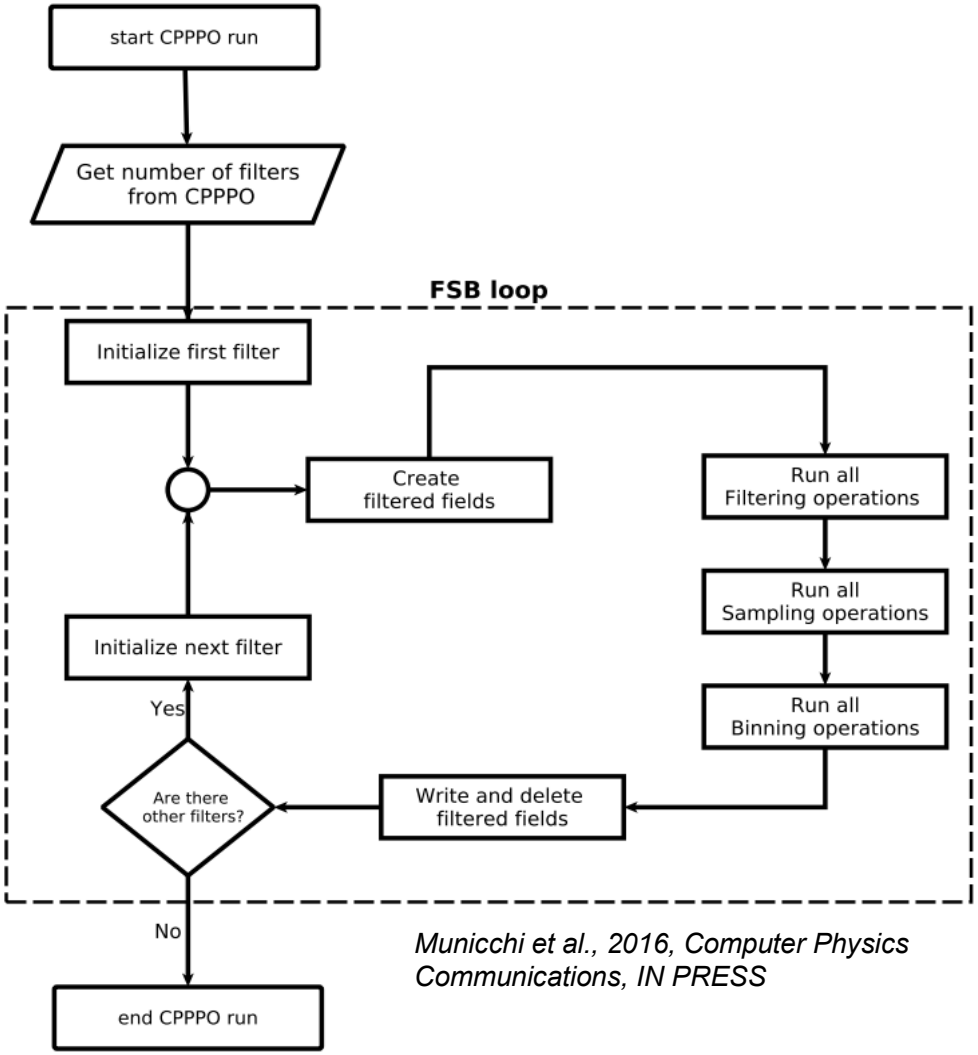
CPPPO - Compilation of fluid/Particle Post Processing rOutines

CPPPO is a C++ library of **parallel** data processing functions.

It is a tool for "**offline scale bridging**", i.e., developing closures for coarse mesh models by **filtering** fine mesh data.

Radl et al., Chemical Engineering Science 117 (2014) 416–425

Main structure



Municchi et al., 2016, Computer Physics Communications, IN PRESS

FSB loop:

- ***Filtering***: Compute averaged fields with custom Kernels.
- ***Sampling***: Select a subset of data.
- ***Binning***: Collapse the selected subset of data to obtain statistical information.

Cell-particle-region selectors

Filtering operations are performed togheter with **Selectors.**

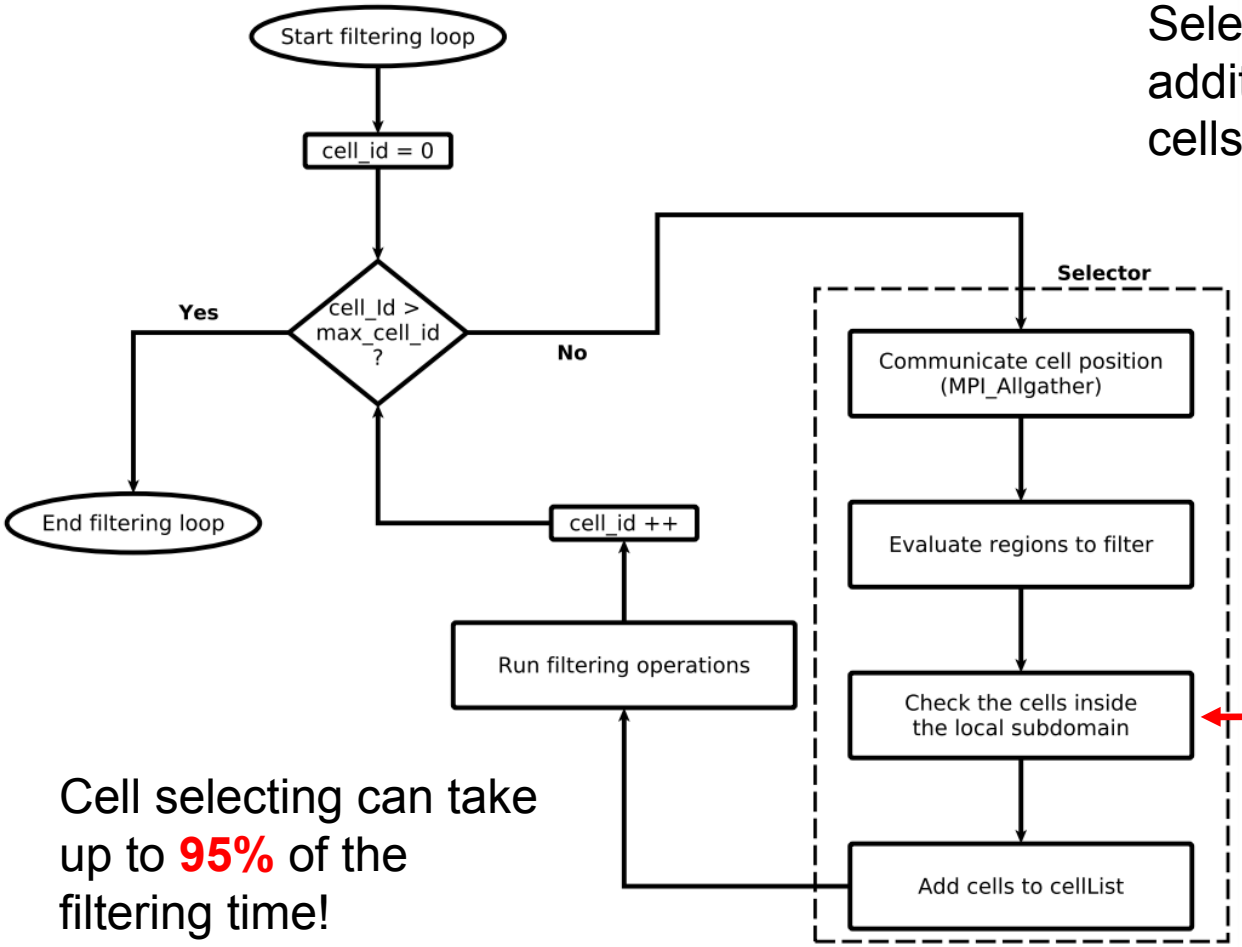
For each filtered cell, the selector evaluates the cells within the filter domain.

CPPPO features two selectors for filtering:

- **Cartesian** : for fully structured grids
- **Unstructured**: for general meshes

In addition, CPPPO features a **cell region selector** to evaluate zones of interest (e.g., bubbles) in the fluid domain.

The filtering loop



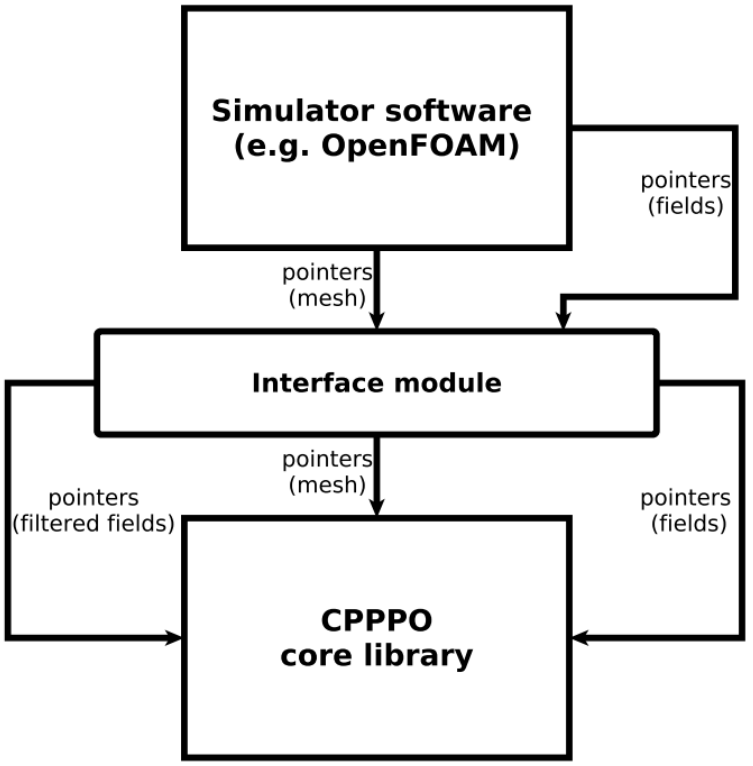
Selectors require additional looping over cells.

Requires sub-looping

Cell selecting can take up to **95%** of the filtering time!

Municchi et al., 2016, Computer Physics Communications, IN PRESS

Coupling to simulator



Municchi et al., 2016, Computer Physics Communications, IN PRESS

The **CPPPO core library** is linked to the simulator software by mean of an **interface module**.

CPPPO only needs pointers to field and mesh quantities.

Additional heap memory is allocated in the interface class to create the filtered fields.

Data requirements for CPPPO

- A set of **nodes**, each one identified by three spatial components.
- A set of scalars representing the **spatial volume** around each node.
- A set of scalars representing **field values** at each node.

These quantities are passed to the core library by mean of the interface library.

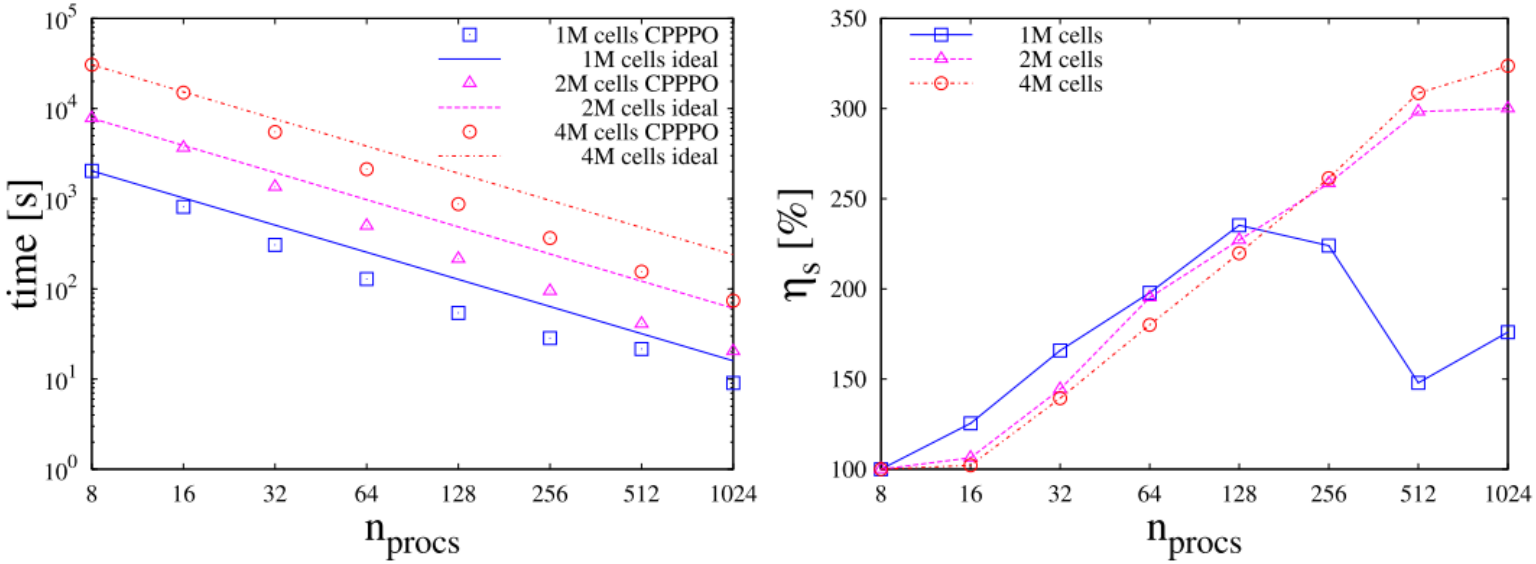
Nodes + volumes = cells

Parallelization requirements for CPPPO

- CPPPO is parallelized using **MPI**
- The parallelization is performed using a **domain decomposition technique**.
- CPPPO relies on MPI **collective operations** for most of the communications

This approach to parallelization is compatible with most of the currently available simulator software.

Parallel scalability

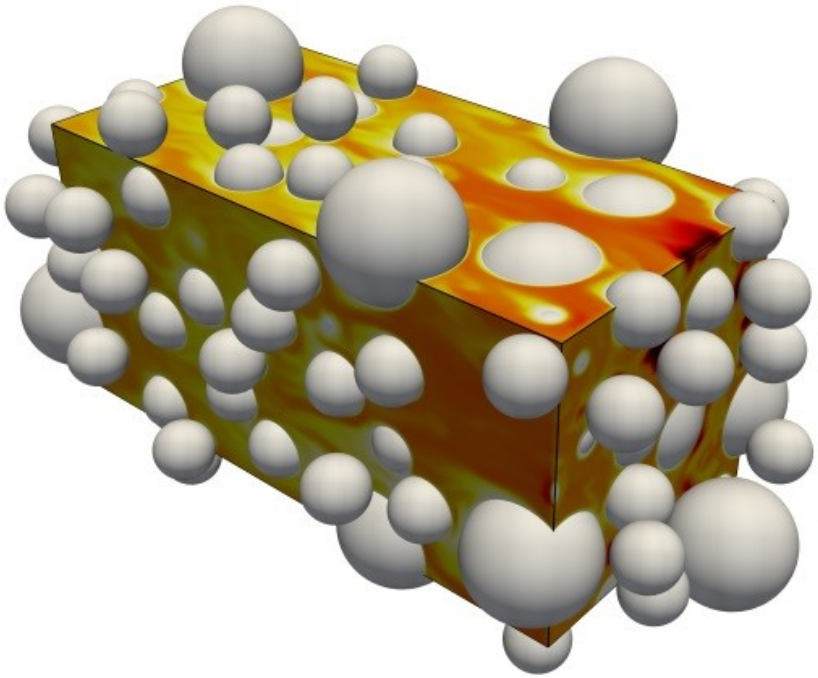


Municchi et al., 2016, Computer Physics Communications, IN PRESS

Strong parallel efficiency (η_s) **well above 100%** on the VSC-3 (*Vienna Scientific Cluster* http://vsc.ac.at/)

The total time is **a small fraction** of the total computational time (**less than 2%** for flow and heat transfer in a particle bed)

Heat and mass transfer in bi-disperse suspensions



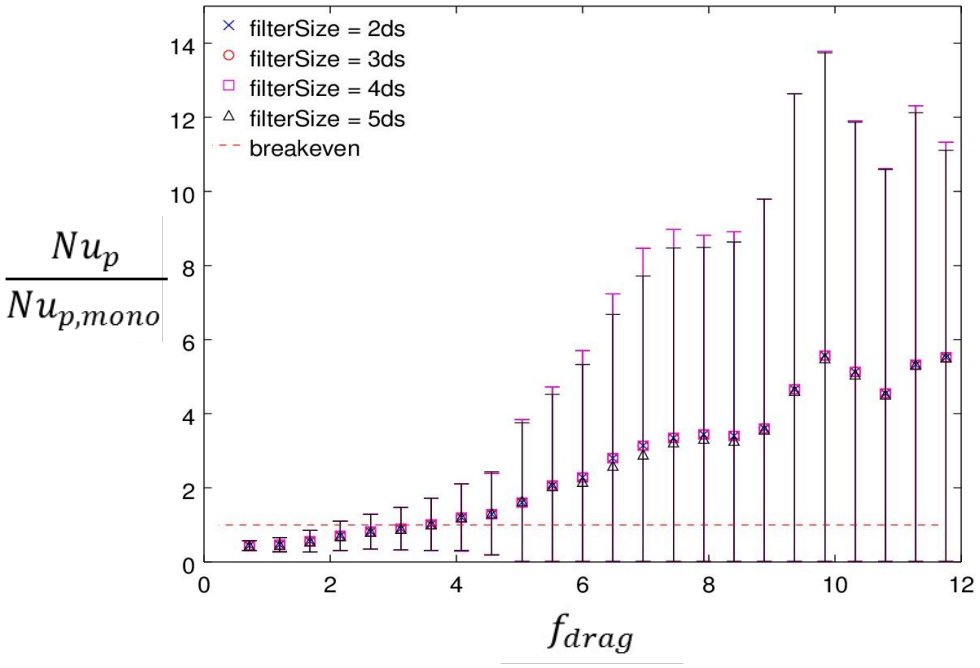
Particle-based Nusselt number

$$Nu_p = \frac{h_p \, d_p}{\lambda_f}$$

Should be expressed as a function of homogeneized (i.e., filtered) quantities.

Filtering at each particle (i.e., **Lagrangian**) is required.

Preliminary results



There exist an **analogy** between the average particle based Nusselt number and the drag force.

These results were obtained by post processing more than 150 large simulations ($\approx 10^7$ cells).

A correlation could be expressed in the convenient form:

$$Nu_p = F(Nu_{p,mono}, f_{drag})$$

- CPPPO is a **flexible**, **powerful** and **open-source** tool for processing large amount of simulation data.
- CPPPO's fast and user friendly post processing allows to **quickly find trends in large data sets**.
- Heat and mass transfer in bi-dispersed suspensions can be related to momentum transfer by mean of an «**analogy ansatz**».

Future developments

- Extending CPPPO by mean of an interface to post-process **many simulations at the same time** (or results from a first post-processing with CPPPO).
- Implementation of more sophisticated algorithms (i.e., **multi-level MC**) for collecting data from different simulations.

Architecture and application of the data filtering library ”CPPPO” to transport phenomena in dense gas-particle flows

Find CPPPO at:
http://www.tugraz.at/en/institute/ippt/downloads-software/

TU Graz, Institute of Process and Particle Engineering , Graz, Austria

Federico Municchi,
Stefan Radl

July 5 2016, PDESoft, Warwick, UK