NanoSim

# Closures for discrete suspension flow models: new insight from particle-resolved simulations and spatial data filtering


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

***Federico Municchi,***
***Stefan Radl***

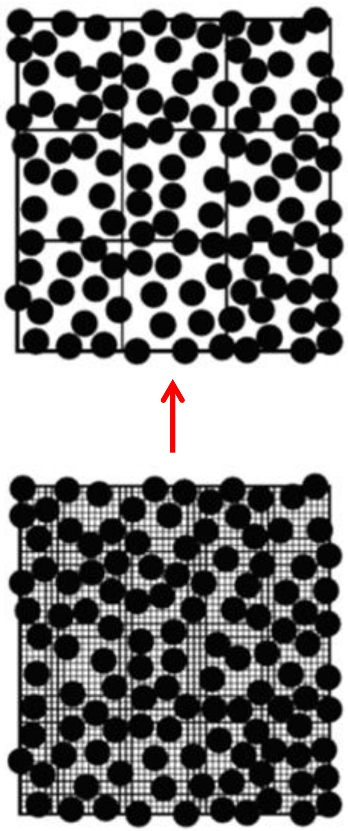


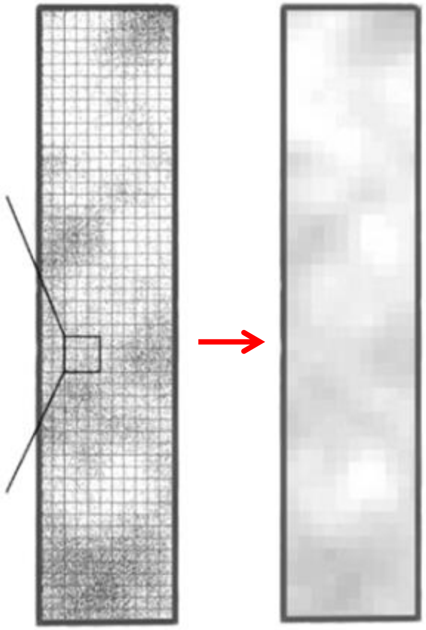
**June 27 2017, International Conference on Numerical Methods for Multiphase Flows, ICNMMF-III, Tokyo, Japan**

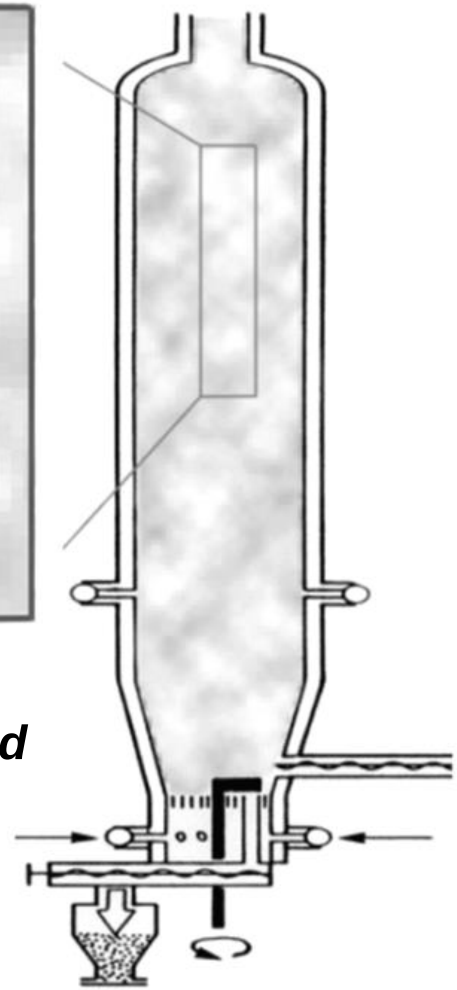
- **Coarse graining of fluid-particle systems**
- **Particle Resolved Simulations**
- **CPPPO and data processing**
- **Bi-disperse and wall bounded suspensions**
- **Conclusions**

## *Particle-Unresolved Formulation*

## *Two Fluids Formulation*







## *Particle-Resolved Formulation*

***PU-EL***

*Volume averaging operator is applied to the governing equations of the continuous phase*

*The new formulation allows the simulation of larger domains*

***TF-EE***

*Volume averaging operator is applied to the governing equations of both continuous and dispersed phases*

*Figures adapted from: Van Der Hoef et al. , Advances in chemical engineering (2006)*

## Volume Averaging Operator

$$\mathcal{F} * v(t,\boldsymbol{x}) = \frac{1}{V_f}\iiint_{\Omega_f} \phi(t,\boldsymbol{x}) v(t,\boldsymbol{x}) d^3\boldsymbol{x}$$

*$\phi(t,\boldsymbol{x})$ is the phase indicator*

$v(t,x)$ should satisfy a partial differential equation:

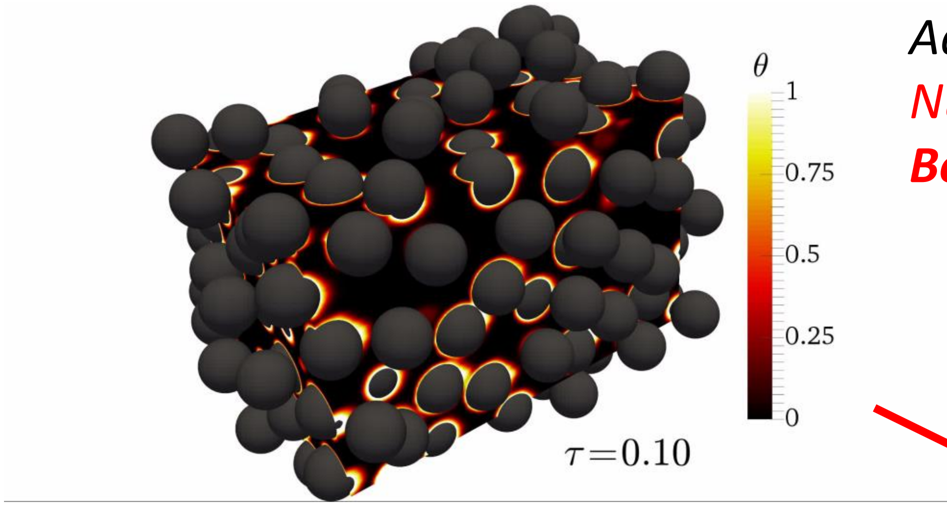
$$\mathcal{P}^k v(t,\boldsymbol{x}) = \Phi(t,\boldsymbol{x})$$

## …and the Closure Problem

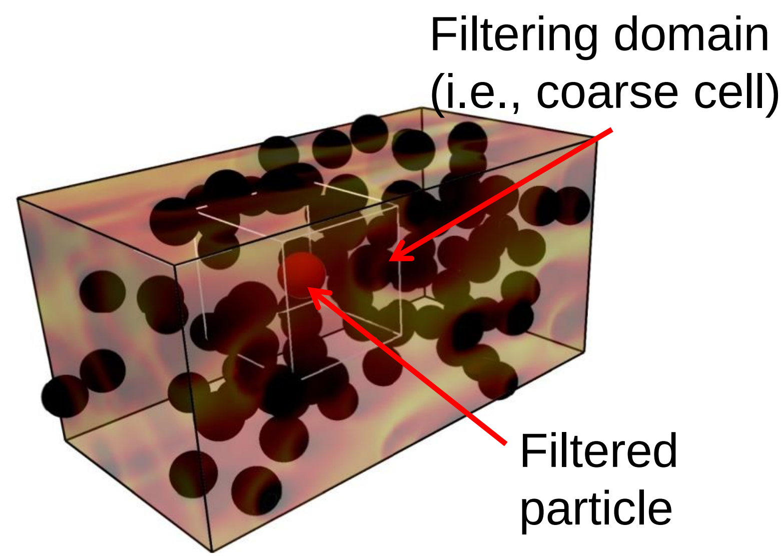
Volume averaging gives rise to terms that need modeling:

- Non-linearity of $\mathcal{P}^k$ and coupled equations (momentum, heat, etc.) require the knowledge of variance and covariance in addition to the average
- Non-commutation of $\mathcal{F}$ and $\mathcal{P}^k$ brings new terms into the equations for the filtered fields (drag force, interphase heat transfer, etc.) that require the knowledge of the original fields

# Coarse-graining: workflow



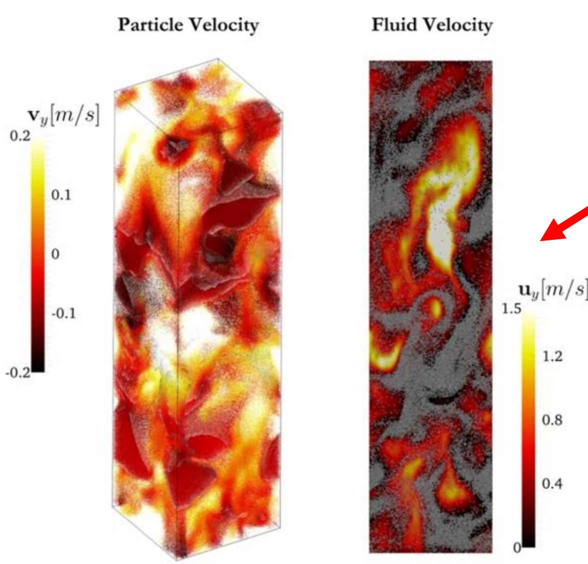

*Advanced algorithms for Direct Numerical Simulation (DNS): **Immersed Boundary Methods***




*Municchi et al., computer physics communications (2016)*

Data filtering and statistical analysis (closure development):
**CPPPO**

*Closures are used in Euler-Lagrange simulations: **Consistency***




*Radl and Sundaresan, CES (2014)*

- Coarse graining of fluid-particle systems
- **Particle Resolved Simulations**
- CPPPO and data processing
- Bi-disperse and wall bounded suspensions
- Conclusions

# Immersed Boundary and Fictitious Domain methods

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i^{IB}$$

$$\frac{\partial \theta}{\partial t} + \frac{\partial u_i \theta}{\partial x_i} = \frac{1}{Pe}\frac{\partial^2 \theta}{\partial x_i \partial x_i} + Q^{IB}$$

*And boundary conditions...*

Take into account the presence of rigid bodies inside the fluid domain

**Immersed Boundary**
The forcing term imposes the Dirichlet boundary condition at the immersed body surface*.

**Fictitious Domain**
The forcing term imposes a rigidity condition inside the immersed body**.

*Peskin C., Journal of computational physics (1971)

**Smagulov S., Preprint CS SO USSR, N 68 (1979)

## *How to calculate the forcing terms at the surface?*

$$\mathcal{P}^k \theta(t, \boldsymbol{x}) = \Phi(t, \boldsymbol{x})$$

*Partial differential equation of order $k$*

$$\sum_{n=0}^{k-1} \alpha_n [\mathbf{n}(t, \boldsymbol{x}) \cdot \boldsymbol{\nabla}]^n \theta\,(t, \boldsymbol{x}) = \gamma$$

*General boundary condition at the particle surface*

*Then the forcing term is calculated from:*

$$f^{IB}(t, \boldsymbol{x}) = \left(\mathcal{P}^k \theta(t, \boldsymbol{x}) - \Phi(t, \boldsymbol{x})\right) + \left(\theta(t, \boldsymbol{x}) - \theta_i(t, \boldsymbol{x})\right)$$

$$\theta_i(t, \boldsymbol{x}) = \left\{\sum_{n=0}^{k-1} \alpha_n [\mathbf{n}(t, \boldsymbol{x}) \cdot \boldsymbol{\nabla}]^n \theta\,(t, \boldsymbol{x})\right\}^{-1} \gamma$$

*Is solution to the boundary condition*

*The boundary operator needs to be inverted!*

*Expanding $\theta$ in Taylor series in the surface normal direction, we obtain a system of equations for each boundary node $\boldsymbol{z_i}$*

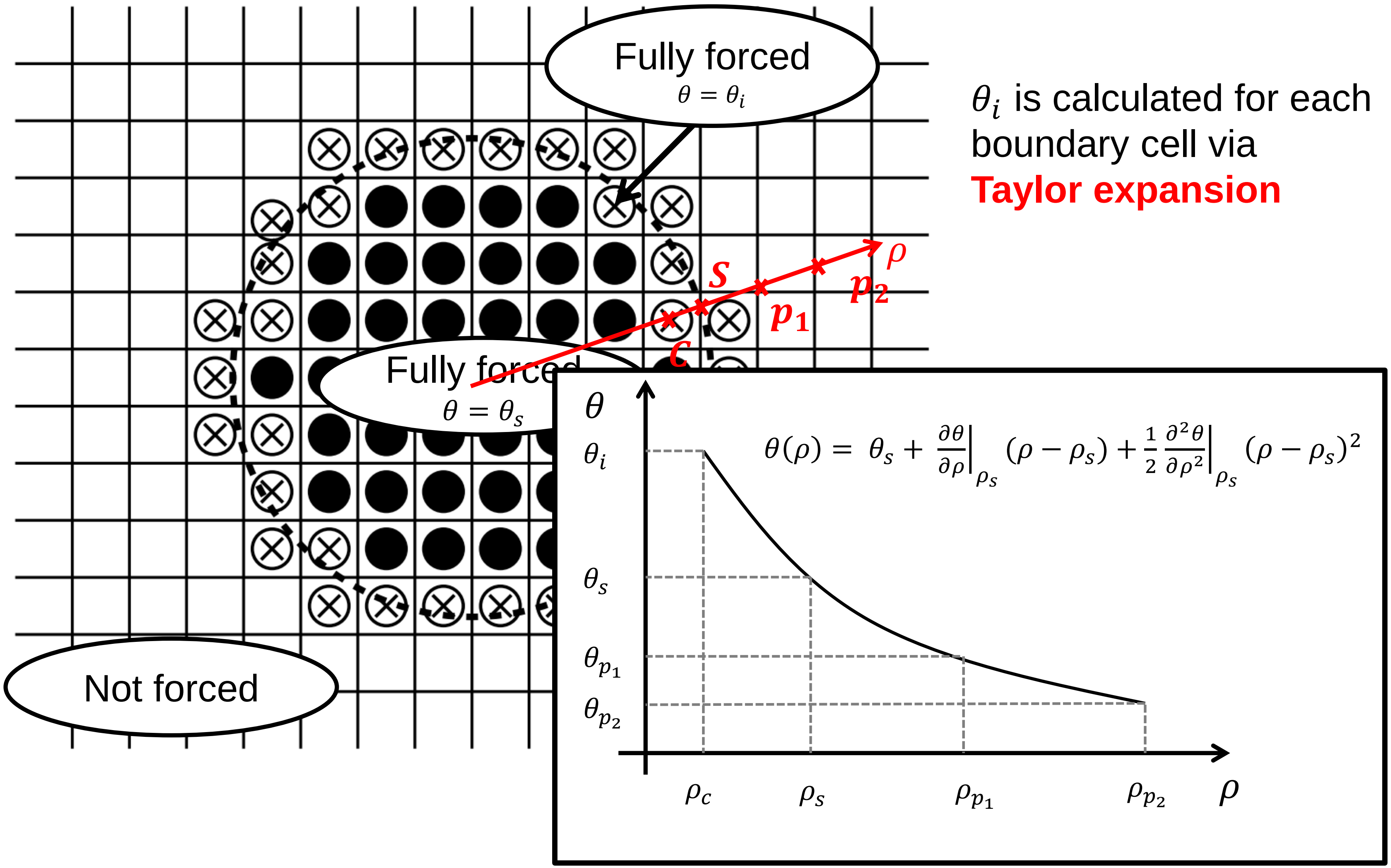
$$\mathbf{M} = \begin{bmatrix} 1 & \Delta s & \frac{\Delta s^2}{2} & \cdots & \frac{\Delta s^N}{N!} \\ 1 & 2\Delta s & \frac{(2\Delta s)^2}{2} & \cdots & \frac{(2\Delta s)^N}{N!} \\ 1 & 4\Delta s & \frac{(4\Delta s)^2}{2} & \cdots & \frac{(4\Delta s)^N}{N!} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_0 & \alpha_1 & \alpha_2 & \cdots & \alpha_N \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} \psi_i(t, \mathbf{z}_i) \\ \left.\frac{\partial \psi}{\partial s}\right|_{\mathbf{z}_i} \\ \left.\frac{\partial^2 \psi}{\partial s^2}\right|_{\mathbf{z}_i} \\ \vdots \\ \left.\frac{\partial^N \psi}{\partial s^N}\right|_{\mathbf{z}_i} \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} \phi(t, \Delta s) \\ \phi(t, 2\Delta s) \\ \phi(t, 3\Delta s) \\ \vdots \\ \gamma_i(t, \mathbf{z}_i) \end{bmatrix}$$

$\mathbf{t}$ *contains interpolated values*

*And the derivatives are obtained from:* $\mathbf{d} = \mathbf{M}^{-1}\mathbf{t}$

*Derivatives are then used to calculate $\theta_i$ at the boundary cell node*

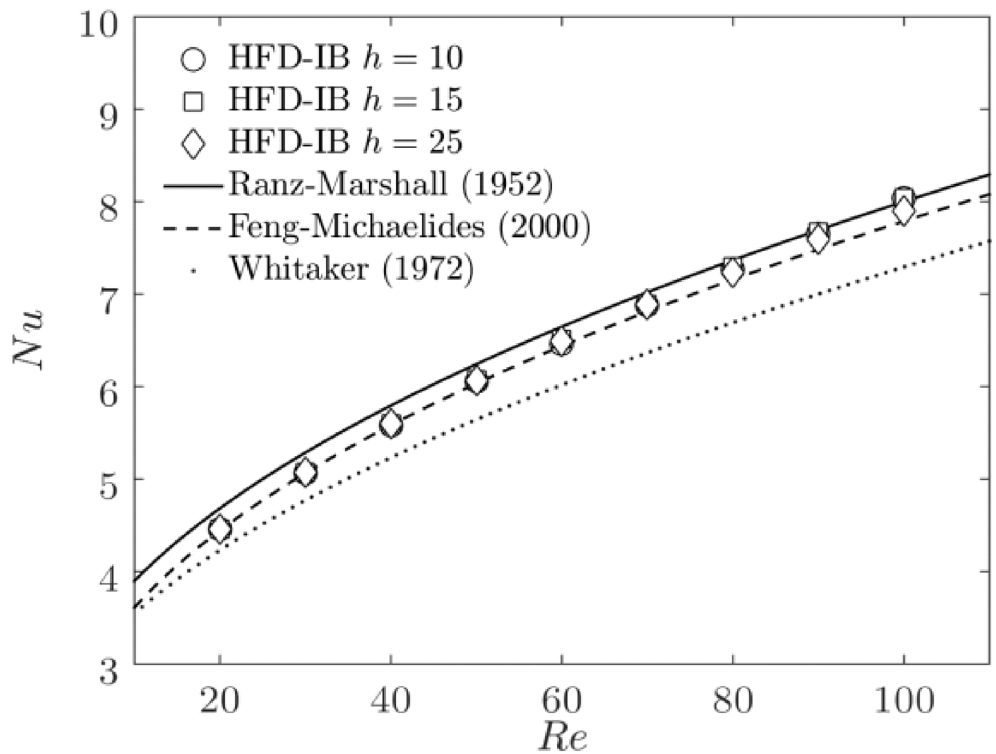
## *Hybrid Fictitious Domain Immersed Boundary Method**




$\theta_i$ is calculated for each boundary cell via **Taylor expansion**

*Municchi and Radl, International Journal of Heat and Mass Transfer (2017)

# Verification - Forced convection around a sphere




*Municchi and Radl, International Journal of Heat and Mass Transfer (2017)*

*Excellent agreement with existing correlations*

*Weak and irregular mesh dependence*

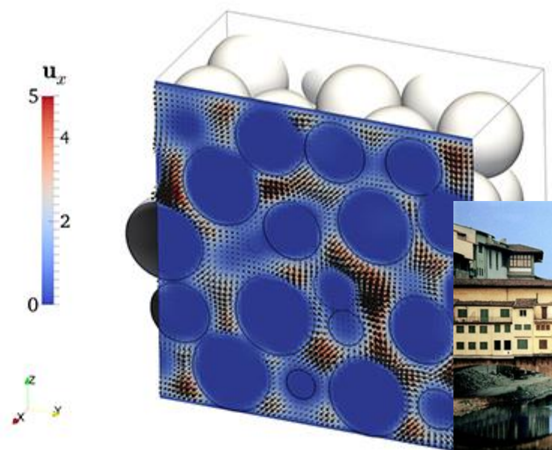
*Very accurate even in coarse grids*

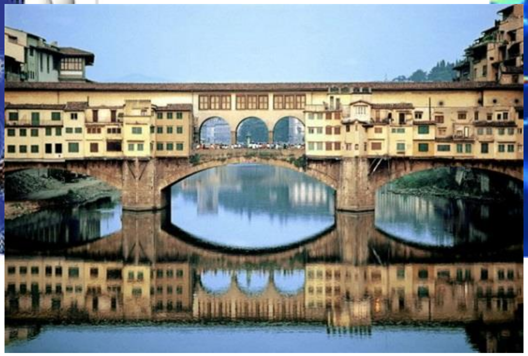
- Coarse graining of fluid-particle systems
- Particle Resolved Simulations
- **CPPPO and data processing**
- Bi-disperse and wall bounded suspensions
- Conclusions

# *Compilation of fluid/Particle Post Processing rOutines*

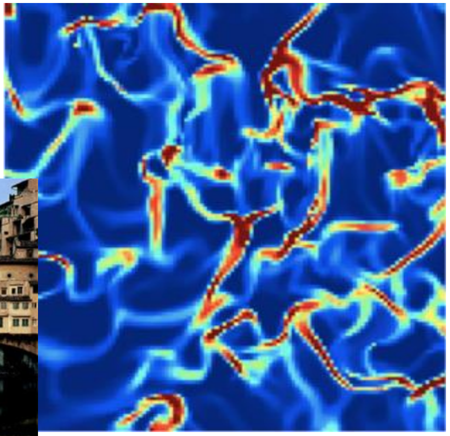
CPPPO is a C++ library of **parallel** data processing functions.

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



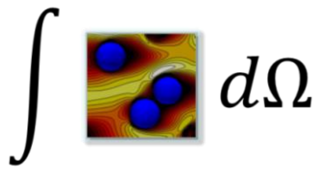



smokingdesigners.com

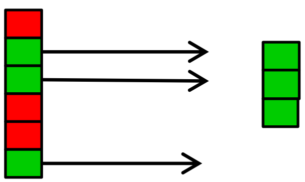


W. Holloway, PhD Thesis, 2012.

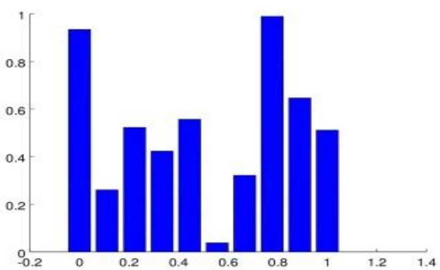
A typical CPPPO run consists of **three sets of operation** performed **on the fly** (i.e., while the solver is running).




Filtering of fluid and particle data, including **variance calculation**

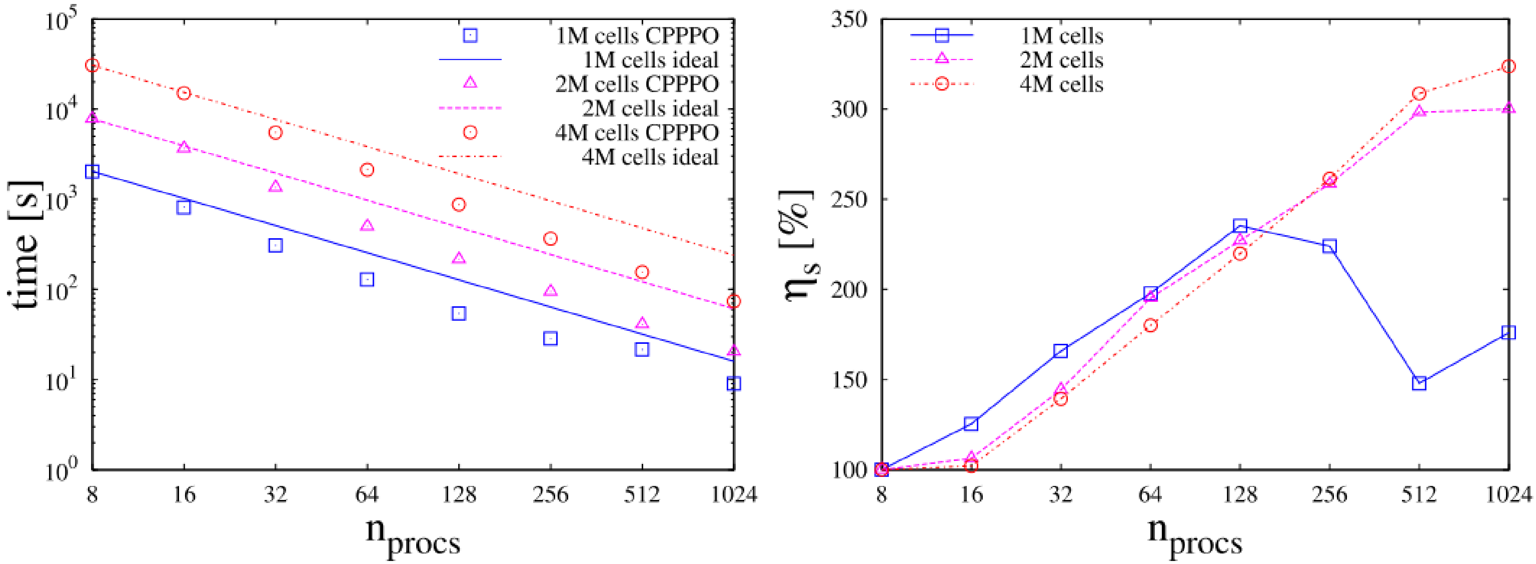


Sampling of filtered data and their derivatives with **statistical biasing** (e.g. limiters)




Binning of sampled data using **running statistics**

# *Parallel scalability analysis*



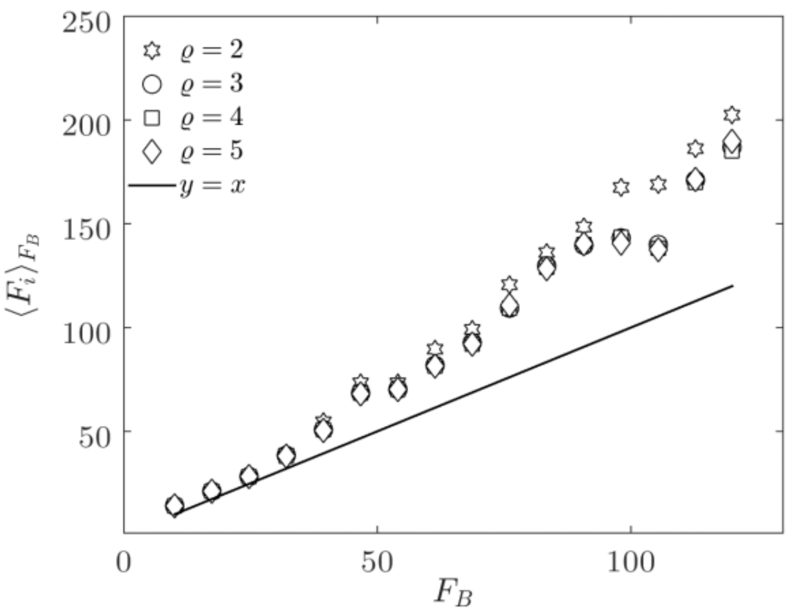

*Municchi et al., Computer Physics Communications (2016)*

Volume averaging algorithms implemented in CPPPO are very efficient in parallel architectures.

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

- Coarse graining of fluid-particle systems
- Particle Resolved Simulations
- CPPPO and data processing
- **Bi-disperse and wall bounded suspensions**
- Conclusions

# Drag Coefficient in Bi-dispersed Suspensions




$$\varrho = \frac{L_{filt}}{d_p}$$

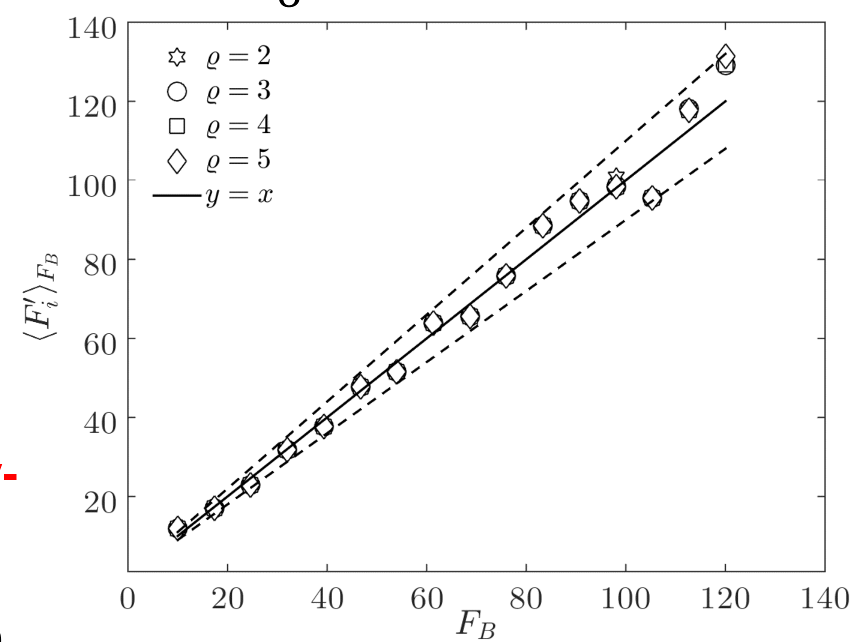
Significant difference with respect to Beetstra*!

Direct subctraction of the filtered pressure gradient force:

$$\boldsymbol{f}_i = \boldsymbol{f}_{i,total} - \frac{\pi}{6} d_i^3 \nabla p_i^{\varrho}$$

Beetstra's approach:

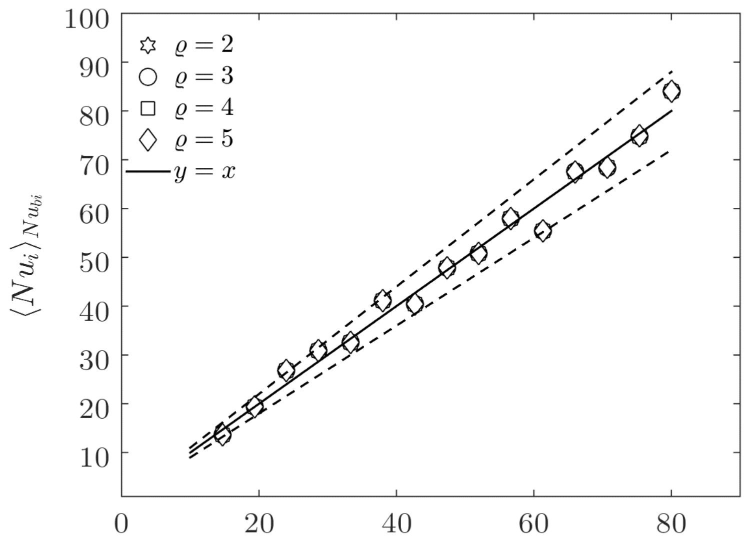
$$\boldsymbol{f}_i = \frac{\boldsymbol{f}_{i,total}}{1 - \phi_p}$$




**Beetstra's drag does not remove the pressure gradient contribution in poly-dispersed suspensions!**

**Beetstra et al., IFAC proceedings Volumes( 2009)*

# Heat transfer in Bi-dispersed Suspensions

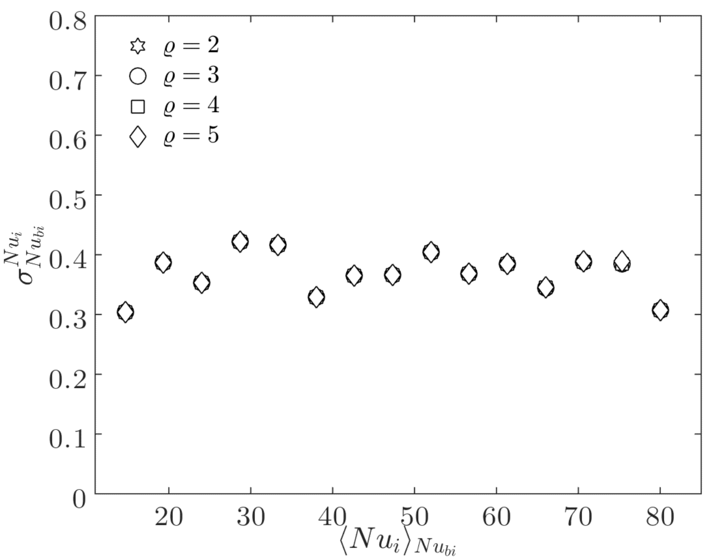


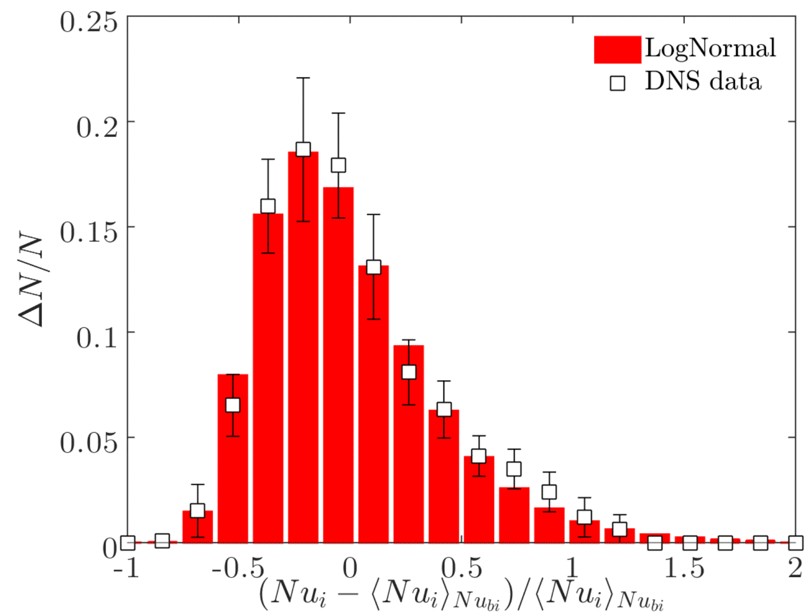

Correlation with the drag force

$$Nu_{i,bi} = Pr^{\frac{1}{3}}\left(12.2 + 0.312 F_{i,corr}\right)$$

Relative standard deviation is almost constant

Log-Normal distribution



# *Wall effects – drag force*

$\phi_p$ =0.1

The drag is **larger at the wall** and **larger for $\varrho$ low.**

$\phi_p$ =0.2

**Oscillations** close to the wall and weak dependence on $\varrho$.

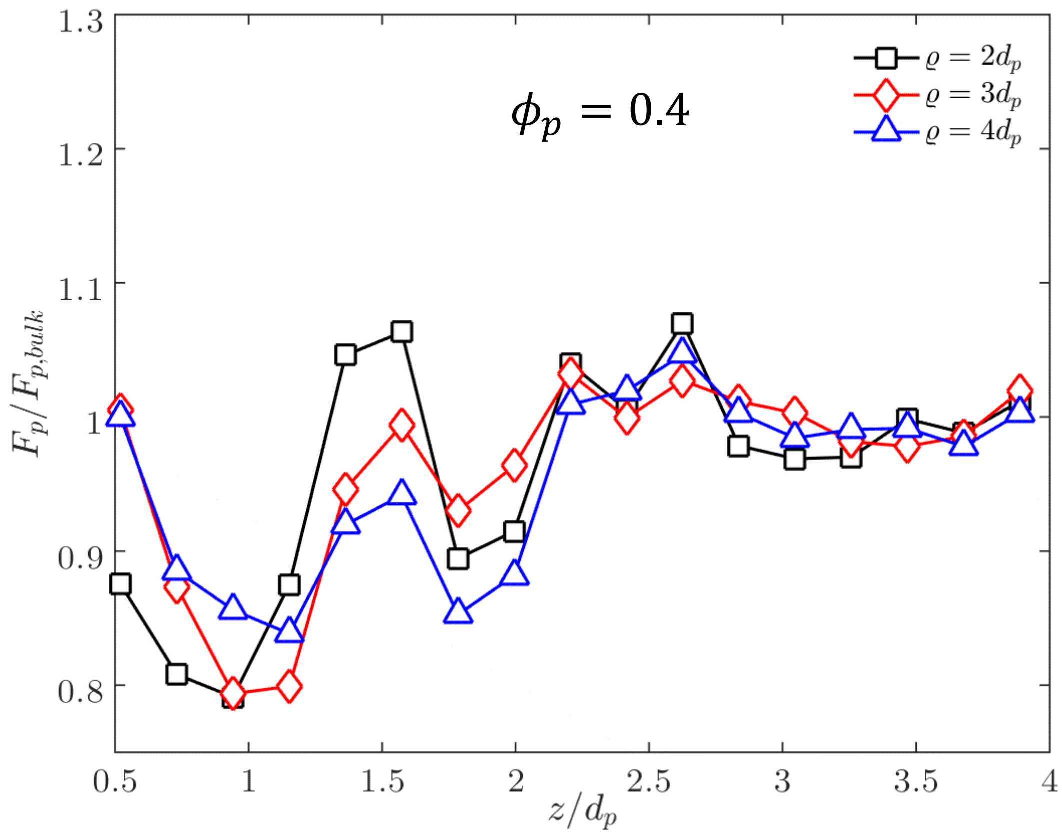
$\phi_p$ =0.3

Minimum at $z = d_p$ and **no dependence** on $\varrho$ .

$\phi_p$ =0.4

The drag is **smaller at the wall** and **smaller for $\varrho$ low.**

**Oscillations** propagate up to $z = 2.5d_p$




The drag at the wall tends to **decrease with increasing $\phi_p$** .

# Wall effects – Nusselt number

$\phi_p$ =0.1

No large wall effects.

$\phi_p$ =0.2

Increasing Nusselt number.

Inverted dependence on $\varrho$.

$\phi_p$ =0.3

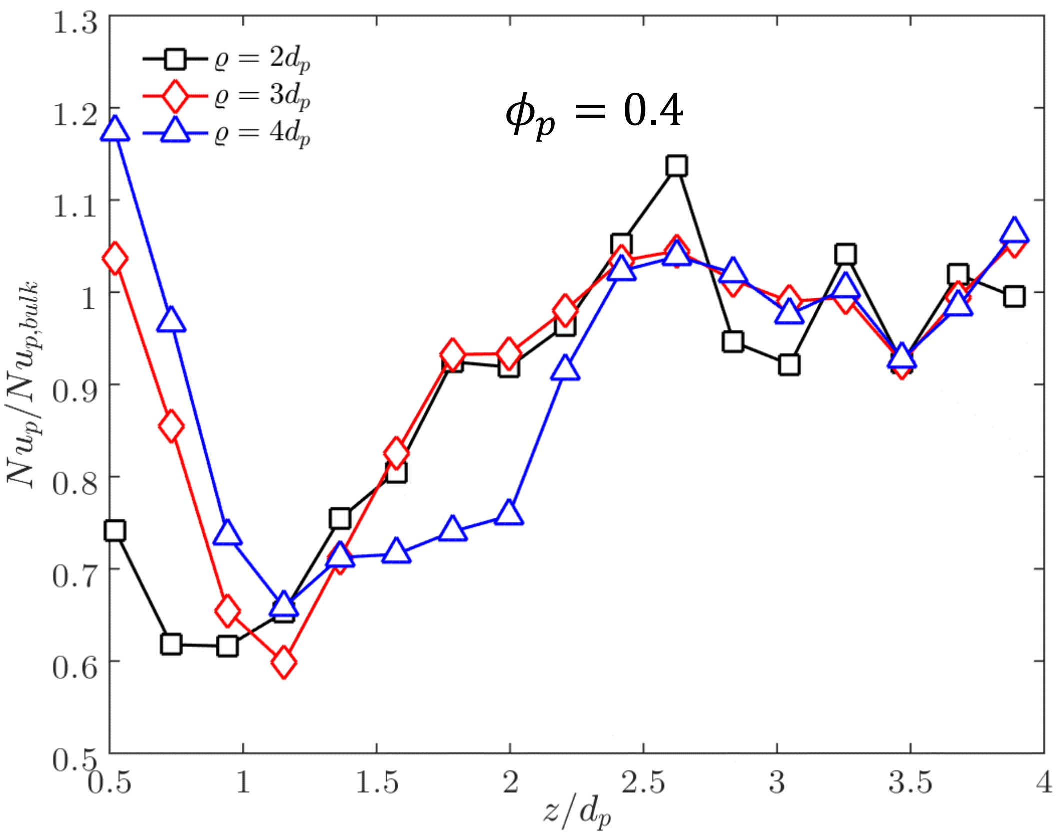
Dependence on $\varrho$ .

**Minimum** at $z = d_p$.

$\phi_p$ =0.4

Clear dependence on $\varrho$.

Stronger minimum and effects up to $z = 2.5d_p$




**Similar** to the drag coefficient **but** **increases at the wall with increasing $\phi_p$**

- Coarse graining of fluid-particle systems
- Particle Resolved Simulations
- CPPPO and data processing
- Bi-disperse and wall bounded suspensions
- **Conclusions**

# Conclusions

- *DNS solver (HFD-IB)*
  - *Our formulation of the HFD-IB allows the imposition of general boundary conditions.*
  - *High accuracy was demonstrated for Dirichlet boundary conditions.*

- *Data analysis (CPPPO)*
  - *Automated extraction of relevant data from simulations.*
  - *High parallel efficiency allows use for massively parallel applications.*

- *Closures*
  - *Consistency with coarse-grained models.*
  - *Filter size dependence arises in heterogeneous systems.*

# Closures for discrete suspension flow models: new insight from particle-resolved simulations and spatial data filtering

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

*Federico Municchi*,
*Stefan Radl*

June 27 2017, International Conference on Numerical Methods for Multiphase Flows, ICNMMF-III, Tokyo, Japan