CPPPO: Compilation of Fluid/Particle Post-Processing Routines

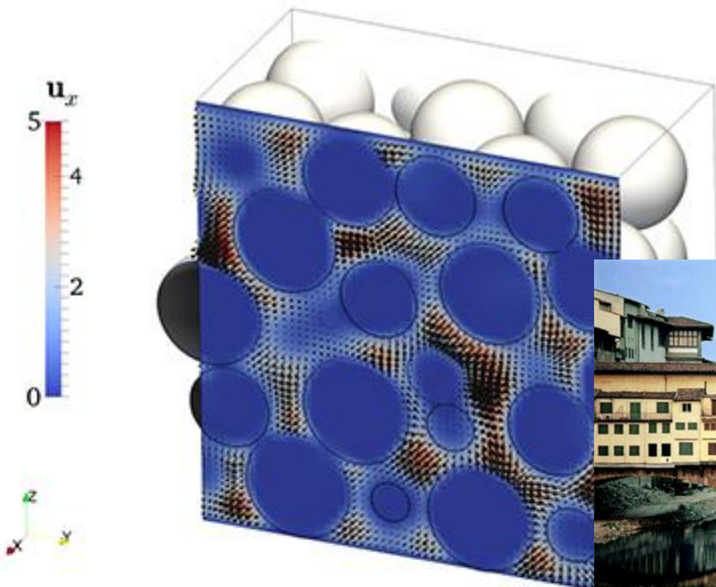
Graz University of Technology

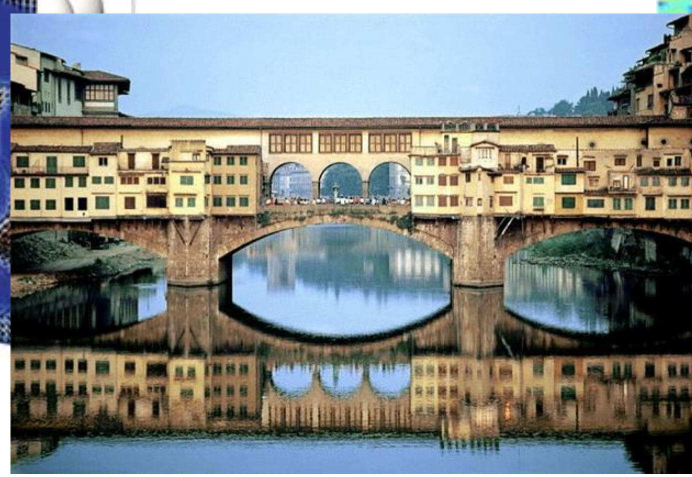
MSc. Federico Municchi
Ass. Prof. Dipl. Ing. Dr. Tech. Stefan Radl
Dipl. Ing. Dr. Tech. Christoph Goniva

July 8 2015, OpenFOAM® User Meeting PFAU X
Graz, IPPT

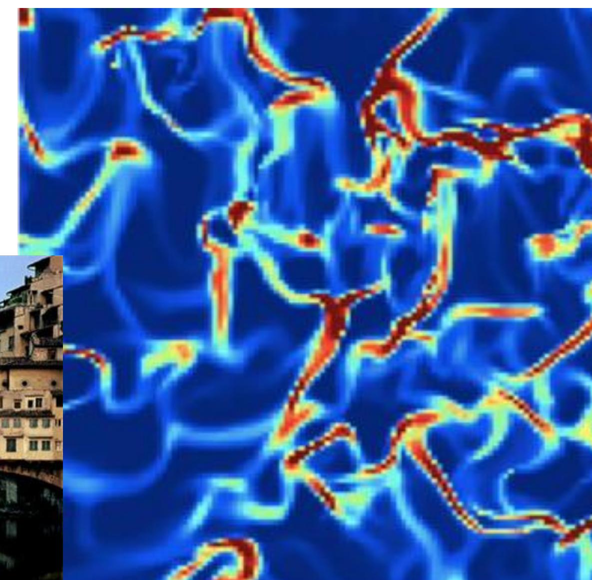
What is CPPPO?

- CPPPO is a C++ library of **parallel** data processing functions.
- It is a tool for "**scale bridging**", i.e., developing closures for coarse mesh models by filtering fine mesh data.





smokingdesigners.com



W. Holloway, PhD Thesis, 2012.

What is CPPPO?

- **Direct Numerical Simulation** has been **widely accepted** since it can provide **information that are not accessible in other ways**.

- *J. Van der Hoef, M.A., van Sint Annaland, M., Deen, N.G., Kuipers, Numerical simulation of dense gassolid fluidized beds: a* ***multiscale modeling*** *strategy, Annu. Rev. Fluid Mech. 40 (2008) 47–70.*
- *M. van der Hoef, M. van Sint Annaland, J. Kuipers, Computational fluid dynamics for dense gas-solid fluidized beds: a* ***multiscale modeling*** *strategy, China Particuology 3 (1-2) (2005) 69–77.*
- *N. G. Deen, S. H. L. Kriebitzsch, M. a. van der Hoef, J. a. M. Kuipers,* ***Direct numerical simulation*** *of flow and heat transfer in dense fluid-particle systems, Chemical Engineering Science 81 (2012) 329–344.*
- *Z.-G. Feng, S. G. Musong,* ***Direct numerical simulation*** *of heat and mass transfer of spheres in a fluidized bed, Powder Technology 262 (2014) 62–70.*
- *H. Tavassoli, S. Kriebitzsch, M. van der Hoef, E. Peters, J. Kuipers,* ***Direct numerical simulation*** *of particulate flow with heat transfer, International Journal of Multiphase Flow 57 (2013) 29–37.*
- *A. A. Zaidi, T. Tsuji, T. Tanaka, A* ***new relation of drag force*** *for high Stokes number monodisperse spheres* ***by direct numerical simulation****, Advanced Powder Technology 25 (6) (2014) 1860–1871.*
- *S. Tenneti, B. Sun, R. Garg, S. Subramaniam, Role of fluid heating in dense gas-solid flow* ***as revealed by particle-resolved direct numerical simulation****, International Journal of Heat and Mass Transfer 58 (1-2)(2013) 471–479.*

Main features

- It performs **on-line filtering** of data from particle/fluid flow simulators **running in parallel**.
- It provides a **number of tools** for drawing samples and performing statistical analysis.
- Can be linked to existing simulation software. Linking to **OpenFOAM®** is currently available as well as **CSV** interfaces for **ANSYS FLUENT®** or **Neptune CFD®**.
- It features **run-time specification of data operations** (no more coding to do fancy post-processing!).

Main structure

C3PO core

- Cell/Particle selection
- Field and particle operations
- I/O (Json/hdf5)
- MPI

C3PO interface

- Data conversion to C3PO format
- Storage of new fields
- I/O interface

CPPPO run

- For every **user-defined filter**:
  - **Filtering Operations**
    Perform field averaging over the fluid domain
  - **Sampling Operations**
    Draw samples using original or filtered fields
  - **Binning Operations**
    Organize sampled values in bins and performs statistics

User interface – c3po.input

```
operation   filtering   Favre0          #Favre averaging
operation   filtering   Favre 1         #Favre averaging

operation   sampling mySample
operation   binning     myBin

selector   cellIJK          myCell0      #selectors for cell index calculation#
selector   cellIJK          myCell1
selector   cellUnstruct     myCell2
selector   cellUnstruct     myCell3
selector   cellUnstruct     myCell4
selector   cellUnstruct     myCell5
selector   cellUnstruct     myCell6
selector   cellUnstruct     myCell7
selector   cellUnstruct     myCell8
selector   cellUnstruct     myCell9

selector   filter     0.75     #filters named as their filter-size
selector   filter     1.00
selector   filter     1.25
selector   filter     1.50
selector   filter     1.75
selector   filter     2.00
selector   filter     2.25
selector   filter     2.50
selector   filter     2.75
selector   filter     3.00
```

The *c3po.input* script allows the user to easily call the functionalities of CPPPO.

User interface – c3po.json

```
{
  "mainSettings":
  {
      "doFiltering": true,
      "doSampling":  false,
      "doBinning":   false,
      "dumpFormat": "json",
      "verbose": false,
      "FieldsToRegister":
      {
      "Vectorfields": " U ",
      "Scalarfields": " voidfraction "
      },
       "filterBCs":
      {
      "x":"periodic",
      "y":"periodic",
      "z":"periodic"
      },

      "storageWriteFields"     :  false,
      "storageWriteParticles" :  true,
       "interfaceWriteFields" : true

  },

…
```

The Json (JavaScript Object Notation) format is used for most of the input scripts in CPPPO. These files allows to define the details of models and operations.

User interface – c3po.json

```
"0.75":        {
            "CoordSys": 1,
            "r": 0.75
          },
 "1.00":    {
            "CoordSys": 0,
            "x": 1.0,
            "y": 1.0,
           "z": 1.0

          },

…

  "2.00":    {
            "CoordSys": 1,
            "r": 2.0
```

Filters are available in spherical and cartesian coordinate systems.

User interface – c3po.json

```
“mySample": {
        "type"             : "general",
        "marker"           : "voidfraction_Favre0",
        "VFieldsToSample" : "velocity_x_Favre0",
        "component"    : 0,
        "sampleCount"      : -1,
        "save2Disk"        : true,
        "save2Bin"         : true,
        "lagrangian"       : false,
        "overwrite"        : true
      },

“myBin": {
        "bincount"         : 20,
        "binUpBorder"      : 0.70,
        "binLowBorder"     : 0,
        "overwrite"        : true
     },
```

For every sampling operation one or more binning operations are necessary.

User interface – c3po.json

```
"Favre0":
          {
           "type":"FavreRunningVariance",
           "VectorfieldsToFilter": "U",
           "ScalarfieldsToFilter": " ",
           "phaseFractionField": "voidfraction",
           "VectorfieldsForVarianceName1": "U",
           "VectorfieldsForVarianceComputeOffDiagonal": [false],
           "ScalarfieldsToFilter": " ",
           "ScalarfieldsForVectorScalarMixedVariance"  : " off",
           "ScalarfieldsForVarianceName1" : "",
           "ScalarfieldsForVarianceName2" : "",
           "lagrangian": true
          }

"Favre1":
          {
           "type":"Favre",
           "VectorfieldsToFilter": "U",
           "ScalarfieldsToFilter": " ",
           "phaseFractionField": "voidfraction",
           "lagrangian": false
          }

}
```

Filtering operations may require many input fields in case of variance/covariance calculation.

Filltering operations can be run both in Eulerian or Lagrangian mode.

Spatial Filtering Operations

The filtering process can be considered a subset of the general operation:

$$\overline{\psi}(\mathbf{x},t) = K * \psi = \int K(\mathbf{x}-\mathbf{z},t-t')\psi(\mathbf{z},t')d\mathbf{z}dt'$$

Where $K(\mathbf{x}-\mathbf{z},t-t')$ is the convoultion Kernel that defines the filtering methodology.

One of the most common Kernels is the top-Hat Kernel (box filter):

$$K(\mathbf{x}-\mathbf{z},t-t') = \delta(t-t')\prod_{i=1}^{3}\frac{\mathcal{H}\left(\frac{\Delta_i}{2}-|x_i-z_i|\right)}{\Delta_i}$$

$\mathcal{H}$: Heaviside step function

Δ_i : spatial filter cut-off lenght on the i direction

Favre Average

Favre variables are mass-weighted variables. A Favre averaged variable is defined as:

$$\widetilde{\psi} = \frac{\overline{\rho\psi}}{\overline{\rho}}$$

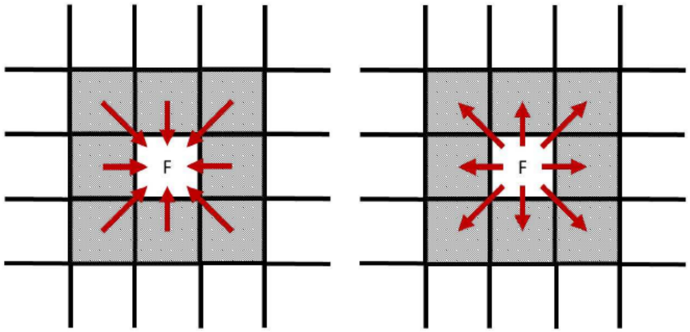
In the case of an incompressible multiphase flow:

$$\widetilde{\psi}_p(\mathbf{x},t) = \frac{\int K(\mathbf{x}-\mathbf{z},t-t')\phi_p(\mathbf{z},t')\psi_p(\mathbf{z},t')d\mathbf{z}dt'}{\int K(\mathbf{x}-\mathbf{z},t-t')\phi_p(\mathbf{z},t')d\mathbf{z}dt'}$$

Where the subscript *p* indicates a phase variable and ϕ_p is the phase volume fraction.

Numerical implementation

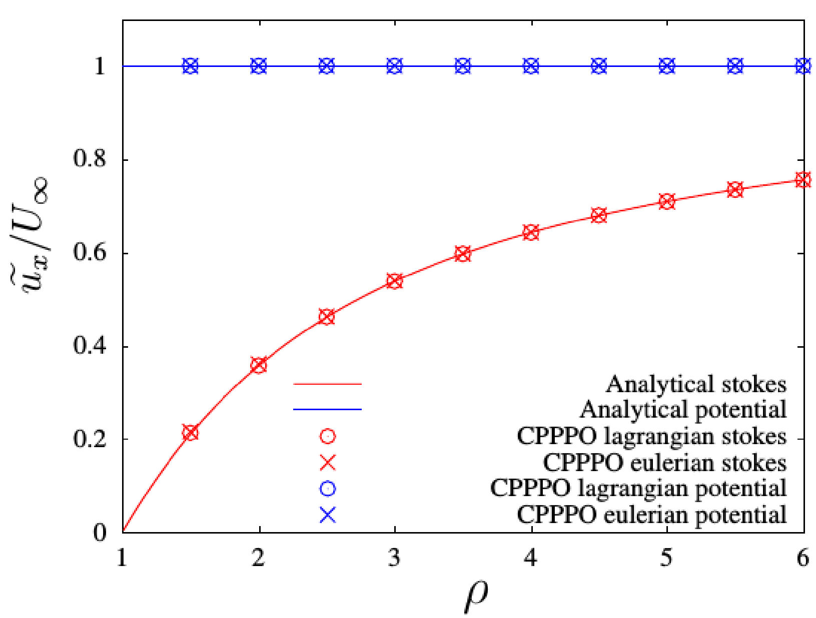
Convergent and divergent filtering algorithms

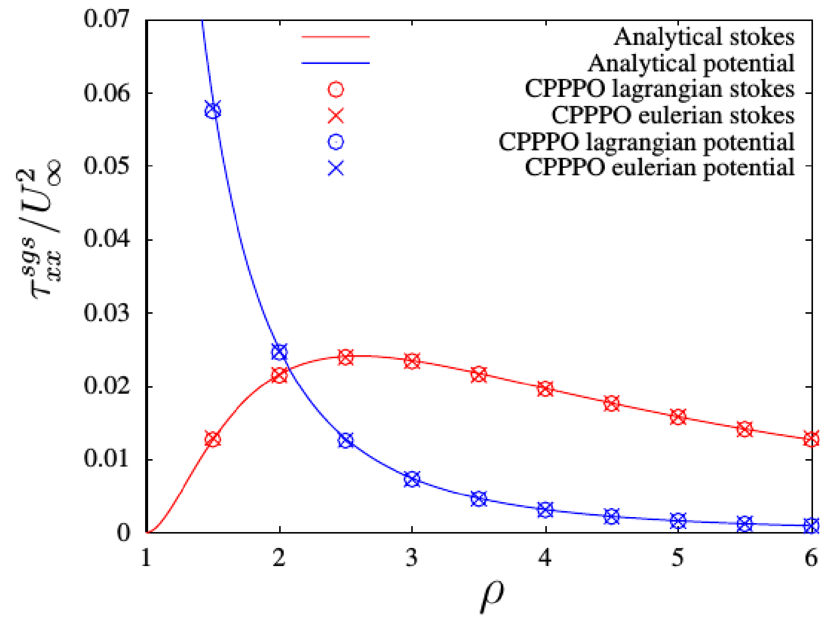


The convergent algorithm is used for **Lagrangian filtering** (i.e., at particle/probe positions) while the divergent for **Eulerian filtering** (i.e., the whole domain is filtered).

Verification shows excellent agreement with analytical results

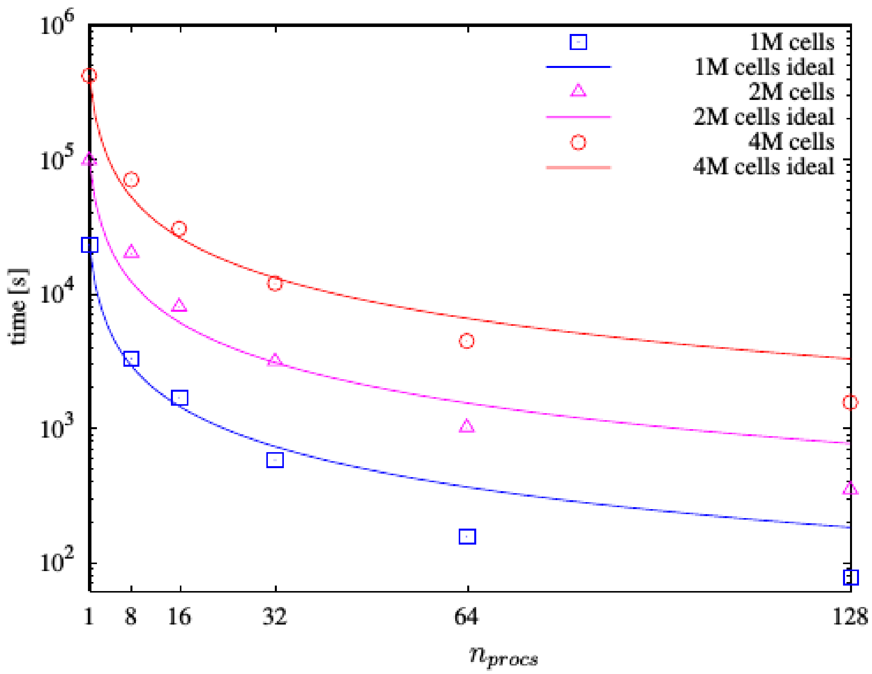
- CPPPO routines are able to correctly calculate field Favre average and variance.
- The accuracy of Eulerian and Lagrangian filters is nearly the same.

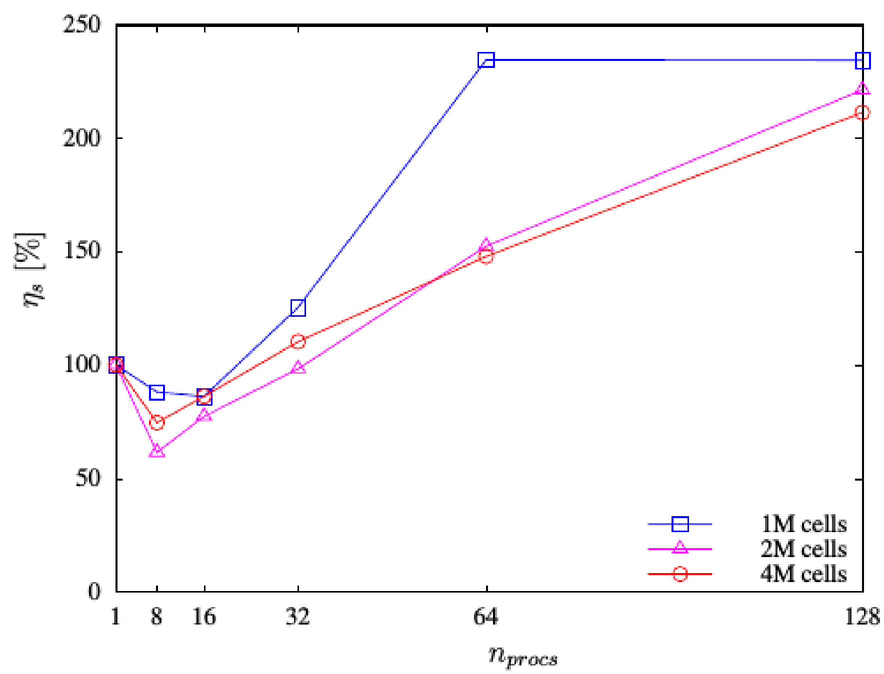




Outstanding parallel efficiency for the unstructured selector

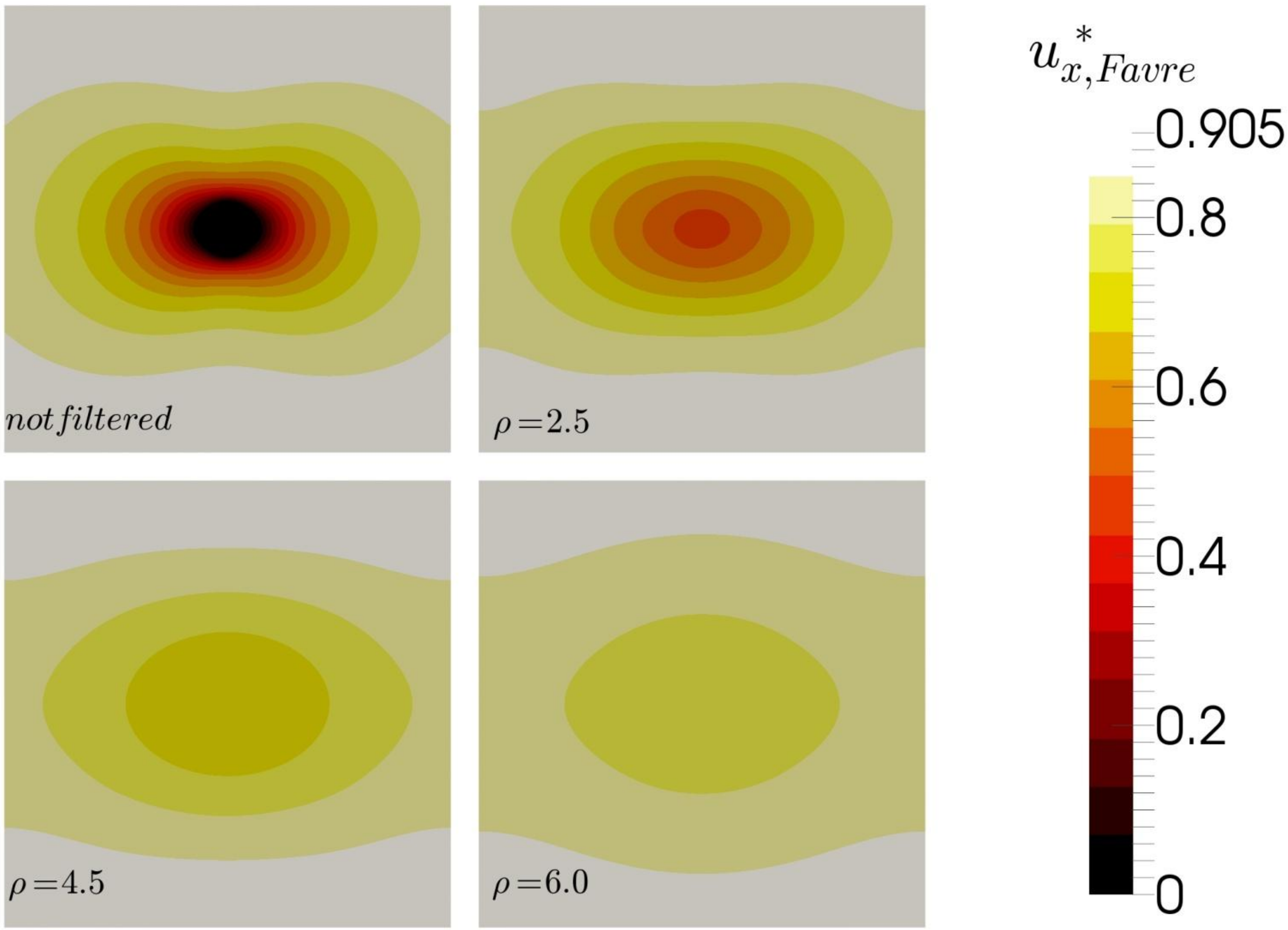
- CPPPO makes efficient use of parallel resources.
- The algorithms are well suited for massively parallel applications.





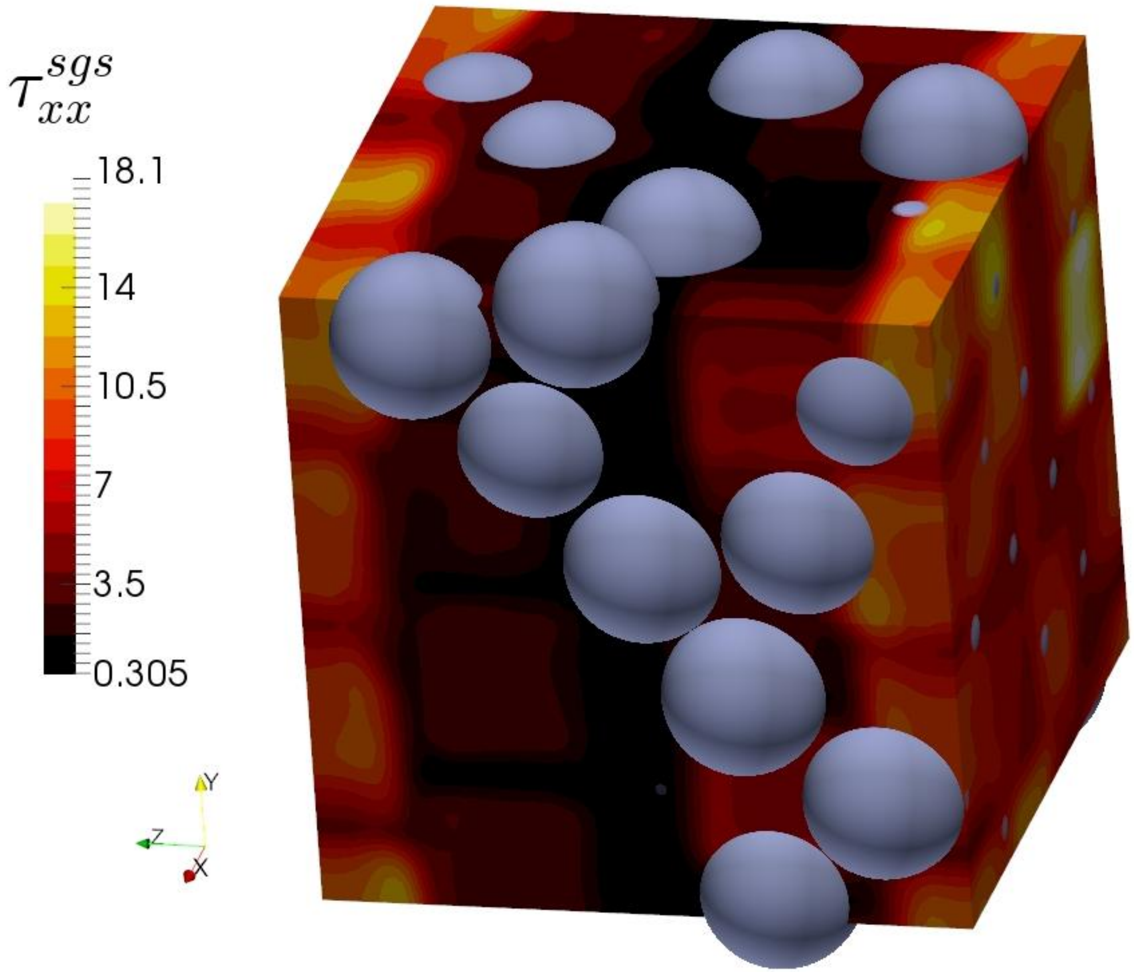
Application - Stokes flow

$$\rho = \frac{R_f}{R}$$



Application – fixed bed

DNS using CFDEM® IB Solver and CPPPO



Conclusions

- CPPPO can evaluate (spatially) filtered fields and their **variance/covariance.**
- CPPPO is well suited for **parallel applications**.
- CPPPO is provided with a **full documentation**, examples and additional scripts.
- Input scripts are easy to **generate automatically**.

CPPPO: Compilation of Fluid/Particle Post-Processing Routines

Thank you for your attention!

Graz University of Technology

MSc. Federico Municchi
Ass. Prof. Dipl. Ing. Dr. Tech. Stefan Radl
Dipl. Ing. Dr. Tech. Christoph Goniva

July 8 2015, OpenFOAM® User Meeting PFAU X
Graz, IPPT