

Development of *solidParticle* Library for the Modeling of Particle Transfer in EM Induced Turbulent Flows

Mihails Ščepanskis⁽¹⁾, Andris Jakovičs⁽¹⁾ , Egbert Baake⁽²⁾ , Bernard Nacke⁽²⁾

(1) – VTPMML, University of Latvia

(2) – ETP, Leibniz University of Hanover

e-mail: mihails.scepanskis@lu.lv

Supported by



20.10.2011

Multiphysical Modelling in OpenFOAM,
Riga



LATVIJAS
UNIVERSITATE
UNIVERSITY OF LATVIA
ANNO 1919

Outline

Motivation

Particle tracking algorithm

The structure of *solidParticle* library

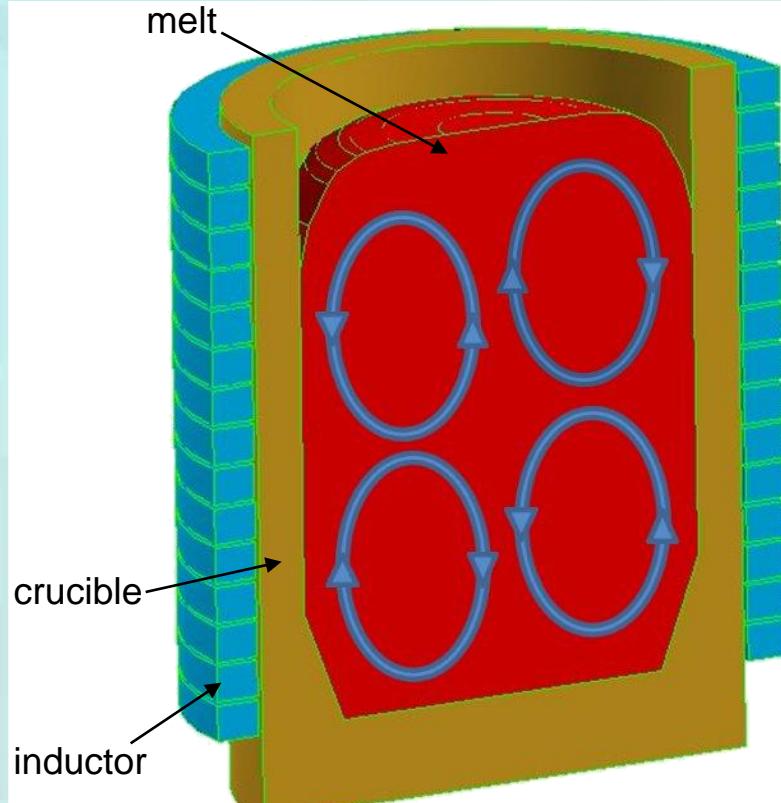
Development of *solidParticle* library

Results and analysis of the model

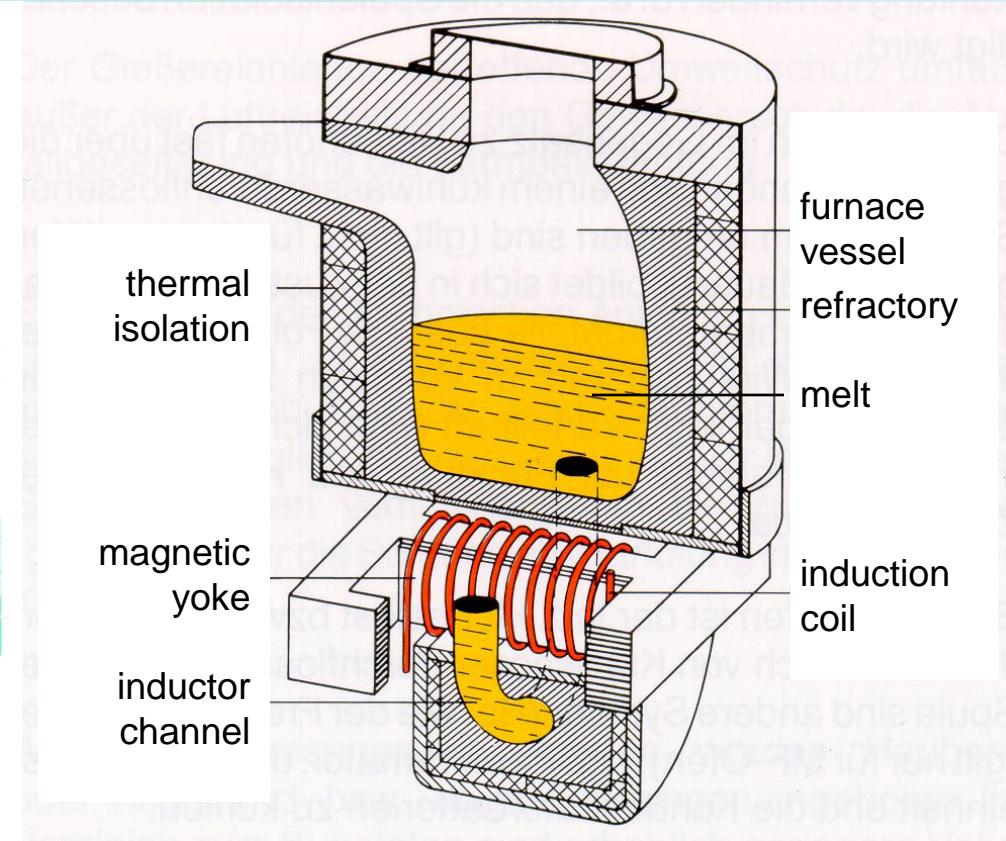
Conclusions

Motivation

Induction crucible furnace (ICF)

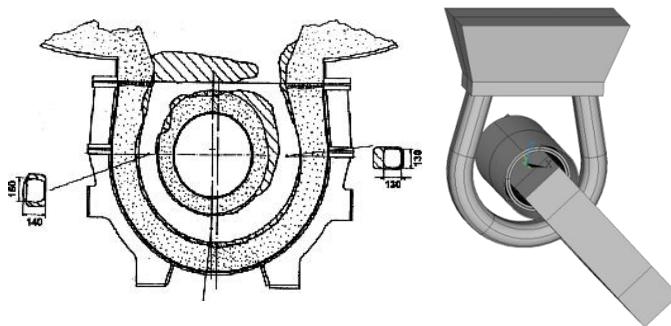


Chanelle induction furnace (CIF)

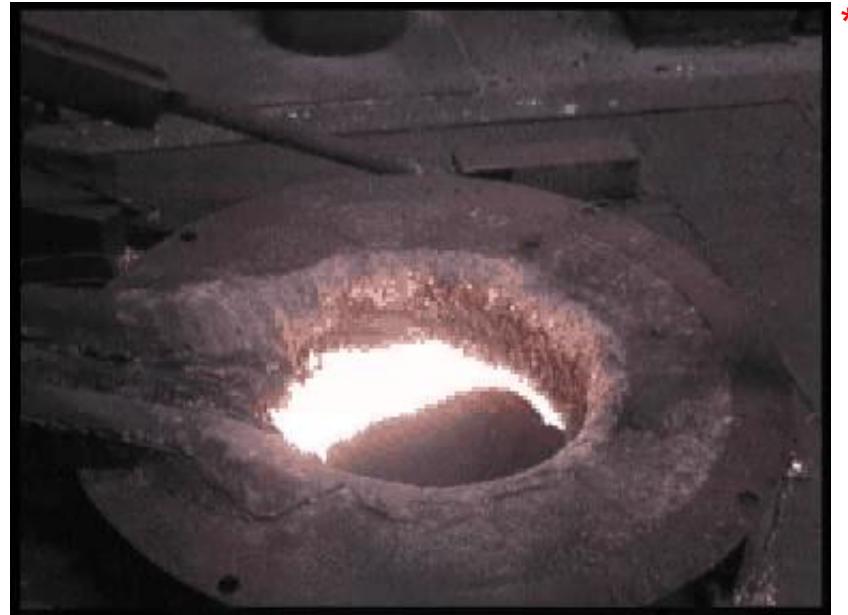


Motivation

Build up formation in the CIF



Mixing in carbon particles in ICF



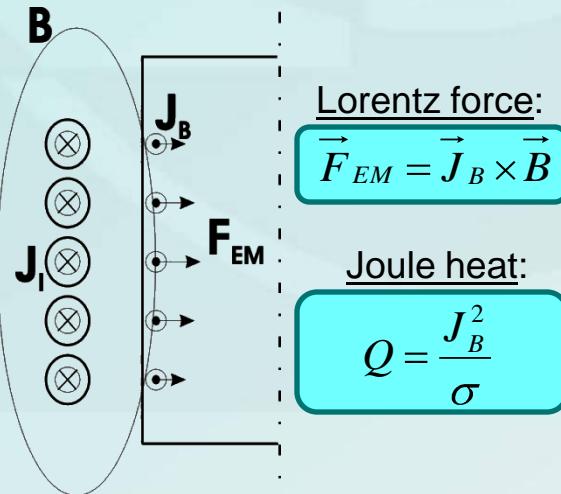
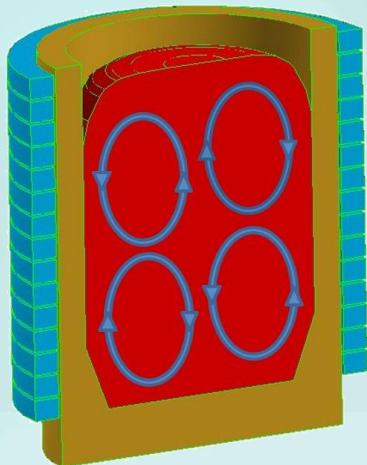
* Courtesy OTTO JUNKER GmbH



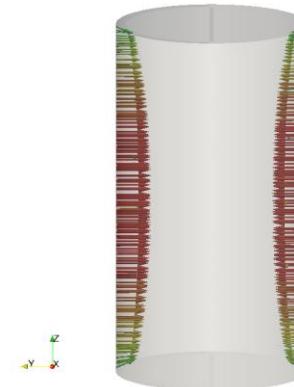
Particle Tracking Algorithm

Electromagnetic calculation

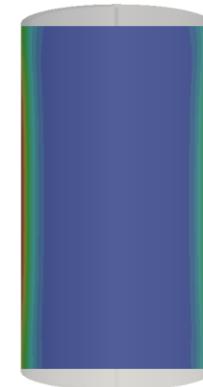
- ANSYS (commercial software)
- GetDP (open-source software)



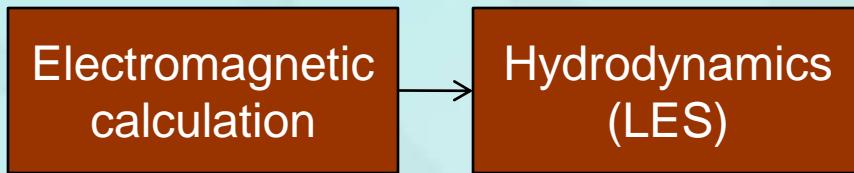
Lorentz force density:



Joule heat:

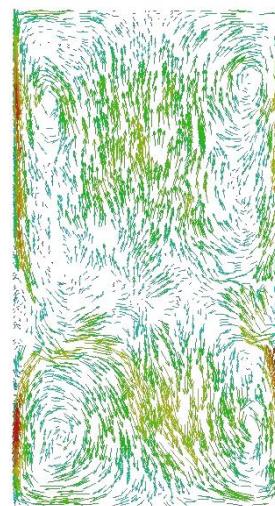


Particle Tracking Algorithm

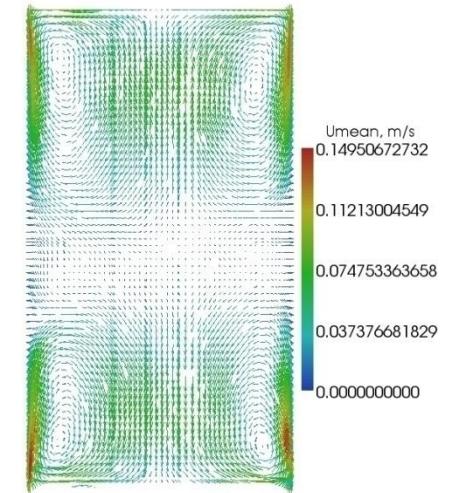


OpenFOAM-1.6
modified *pisoFOAM* solver

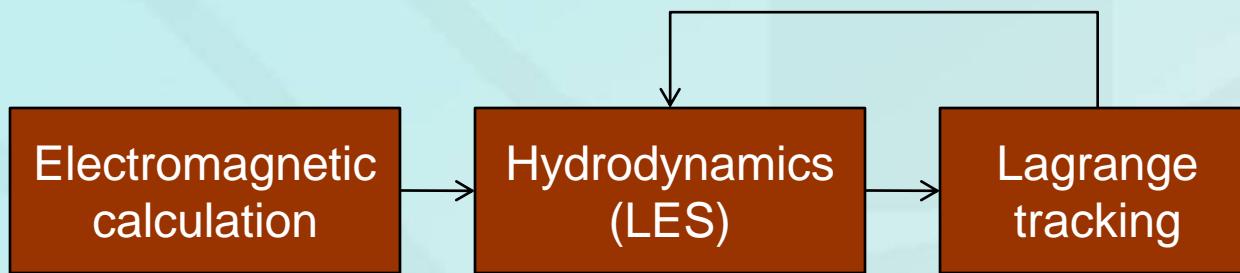
```
volVectorField FF = - F/rho0 + beta*(T0 - T)*q;  
// Momentum predictor  
fvVectorMatrix Ueqn  
{  
    fvm::ddt(U)  
    + fvm::div(phi, U)  
    + turbulence->divDevReff(U)  
};  
UEqn.relax();  
if (momentumPredictor)  
{  
    solve(UEqn == -fvc::grad(p)-FF);  
}
```



Averaged flow velocity:



Particle Tracking Algorithm



OpenFOAM-1.6
developed *solidParticle* library

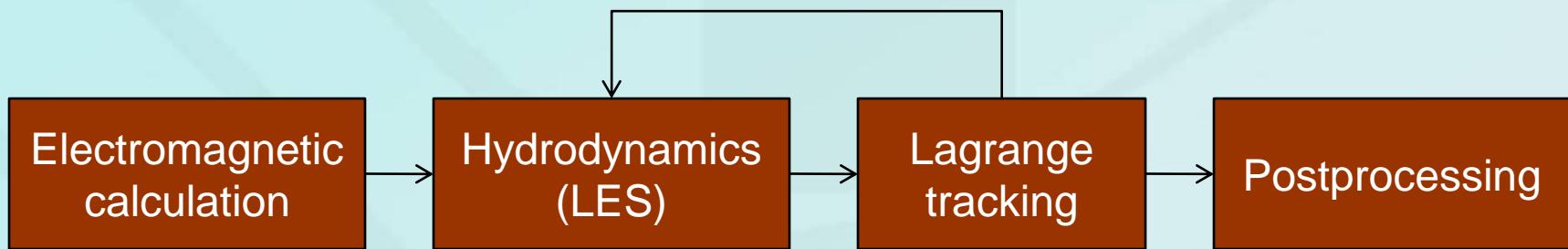
LES-based Euler-Lagrange one-way coupled model (in the limit of dilute conditions)

Assumptions:

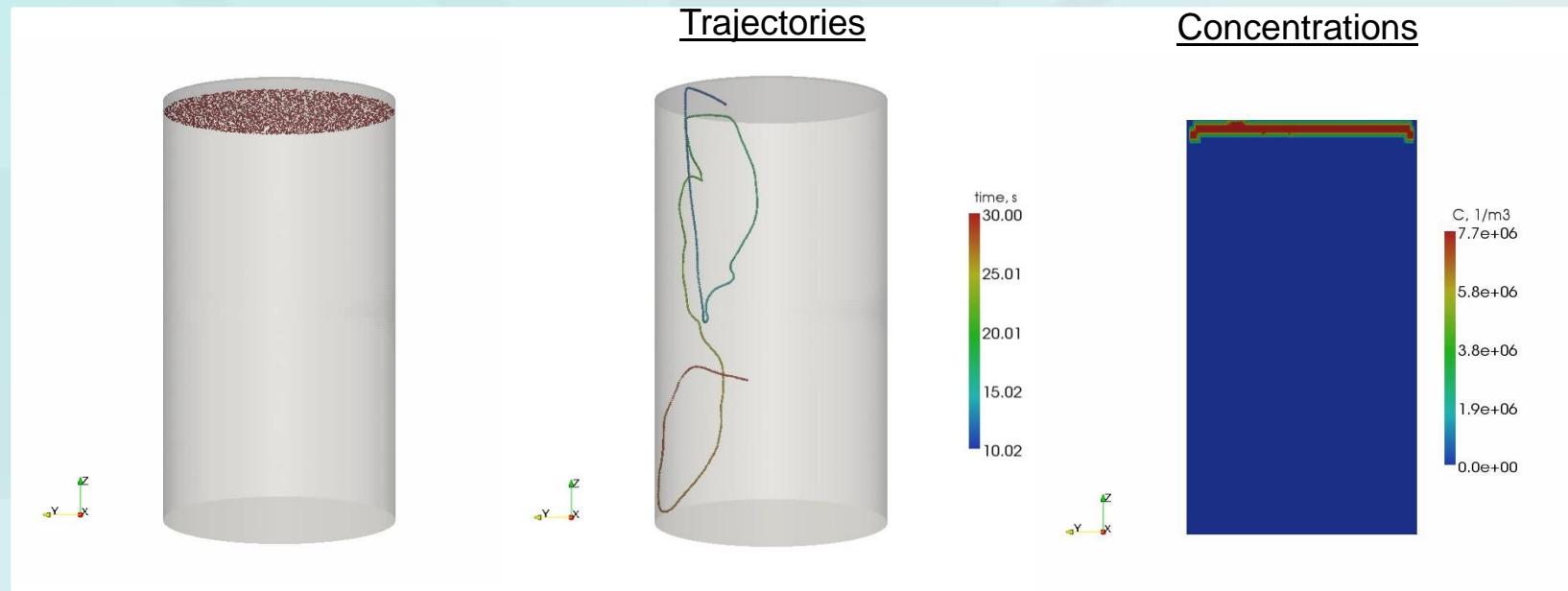
- Non-conductive spherical particles;
- The particle-particle interaction is negligible;
- The particles do not affect the pattern and the velocities of the flow.



Particle Tracking Algorithm



ParaView + self-made tools



The Structure of *solidParticle* Library

```
void Foam::solidParticleCloud::move(const dimensionedVector& g)
    for each particle in cloud
void Foam::solidParticle::move(solidParticle::trackData& td)

template<class particleType>
template<class trackData>
Foam::scalar Foam::Particle<ParticleType>::trackToFace
(const vector& endPosition, trackData& td)

template<class ParticleType>
void Foam::Particle<ParticleType>::findFaces
(const vector& position, DynamicList<label>& faceList)
const
```

Flowchart illustrating the movement logic:

```
graph TD
    A{cross face} -- no --> B["move to endPosition"]
    A -- yes --> C{internal face}
    C -- no --> D["void Foam::solidParticle::hitWallPatch  
const wallPolyPatch& wpp,  
solidParticle::trackData& td  
change velocity"]
    C -- yes --> E["move to neighbour cell"]
    E --> F["calculate velocity (solve Lagrangian equation)"]
```

The defects of the library:

- Problems in parallel calculations;
- Particles should not exceed the size of the cells.



The Structure of *solidParticle* Library

```
void Foam::solidParticleCloud::move(const dimensionedVector& g)
    for each particle in cloud
void Foam::solidParticle::move(solidParticle::trackData& td)

template<class particleType>
template<class trackData>
Foam::scalar Foam::Particle<ParticleType>::trackToFace
(const vector& endPosition, trackData& td)

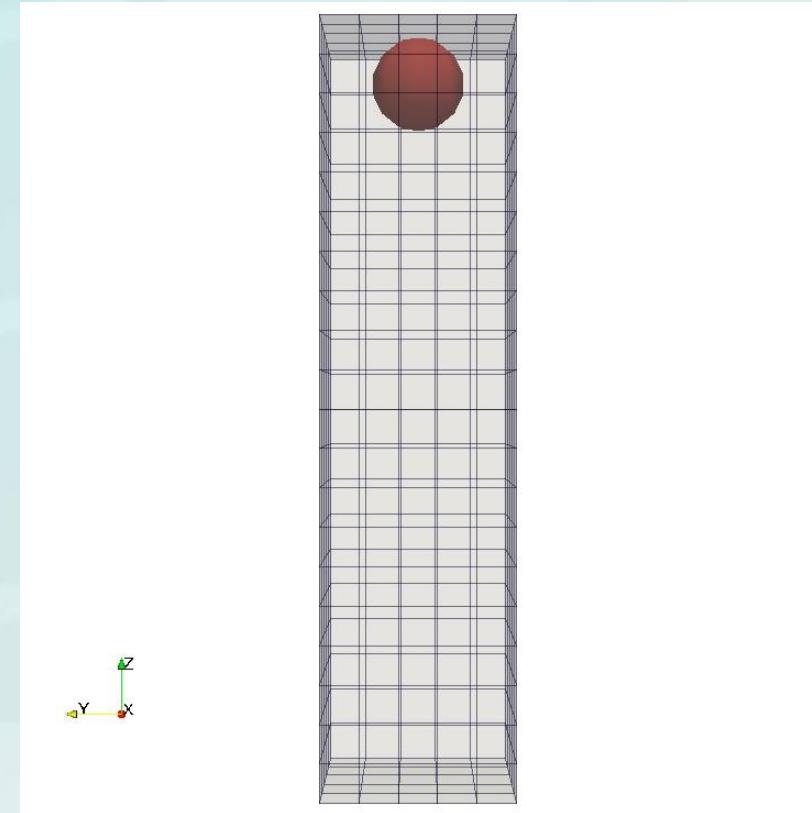
template<class ParticleType>
void Foam::Particle<ParticleType>::findFaces
(const vector& position, DynamicList<label>& faceList)
const
```

Flowchart illustrating the movement logic:

```
graph TD
    A{cross face} -- no --> B["move to endPosition"]
    A -- yes --> C{internal face}
    C -- no --> D["void Foam::solidParticle::hitWallPatch  
const wallPolyPatch& wpp,  
solidParticle::trackData& td)  
  
change velocity"]
    C -- yes --> E["move to neighbour cell"]
    E --> F["calculate velocity (solve Lagrangian equation)"]
```

The defects of the library:

- Problems in parallel calculations;
- Particles should not exceed the size of the cells. **Treated**



The Development of *solidParticle* Library

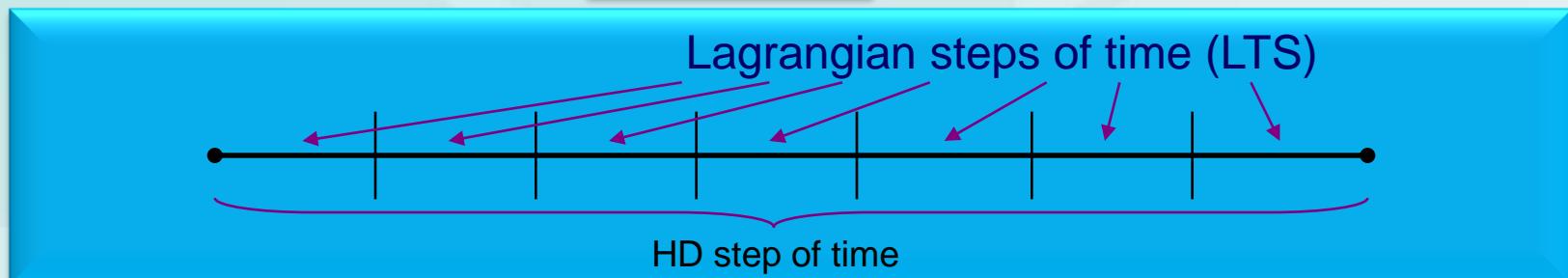
solidParticle library (*solidParticle.C*):

$$\frac{d\mathbf{u}_p}{dt} = \underbrace{C_D \cdot (\mathbf{u}_f - \mathbf{u}_p)}_{\text{drag force}} + \underbrace{\left(1 - \frac{\rho_f}{\rho_p}\right) \cdot \mathbf{g}}_{\text{buoyancy force}}, \quad C_D = \frac{18\nu}{d^2} \frac{\rho_f}{\rho_p} (1 + 0.15 \cdot Re_p^{0.687}) \quad (\text{Schiller \& Naumann, 1933})$$

M. Ščepanskis, A. Jakovičs, E. Baake *J. Phys.: Conf. Ser.* Forthcoming 4th quarter 2011

$$\underbrace{\left(1 + \frac{C_A}{2} \frac{\rho_f}{\rho_p}\right) \cdot \frac{d\mathbf{u}_p}{dt}}_{d\mathbf{u}_p/dt + \text{added mass force}} = \underbrace{C_D \cdot \mathbf{U}}_{\text{drag force}} + \underbrace{\left(1 - \frac{\rho_f}{\rho_p}\right) \cdot \mathbf{g}}_{\text{buoyancy force}} - \underbrace{\frac{3}{4} \frac{1}{\rho_p} \mathbf{f}_{\text{em}}}_{\text{EM force}} + \underbrace{\frac{\rho_f}{\rho_p} C_L \cdot (\mathbf{U} \times (\nabla \times \mathbf{U}))}_{\text{lift force}} + \underbrace{\left(1 + \frac{C_A}{2} \frac{\rho_f}{\rho_p}\right) \cdot \frac{D\mathbf{u}_f}{Dt}}_{\text{acceleration} + \text{added mass}} *$$

non-linearities



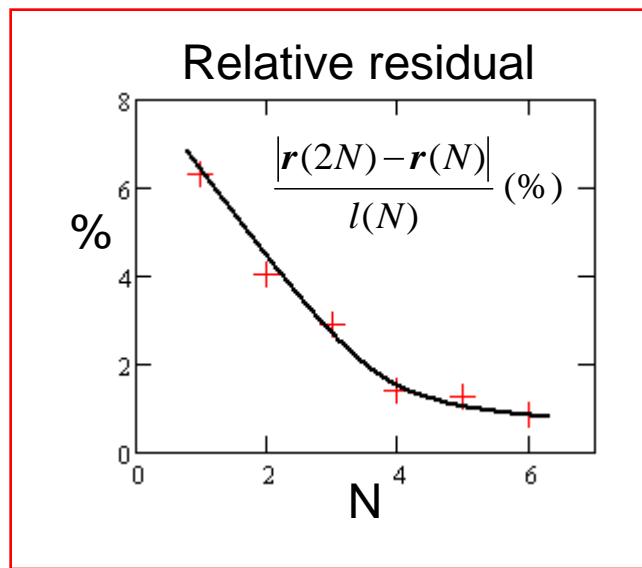
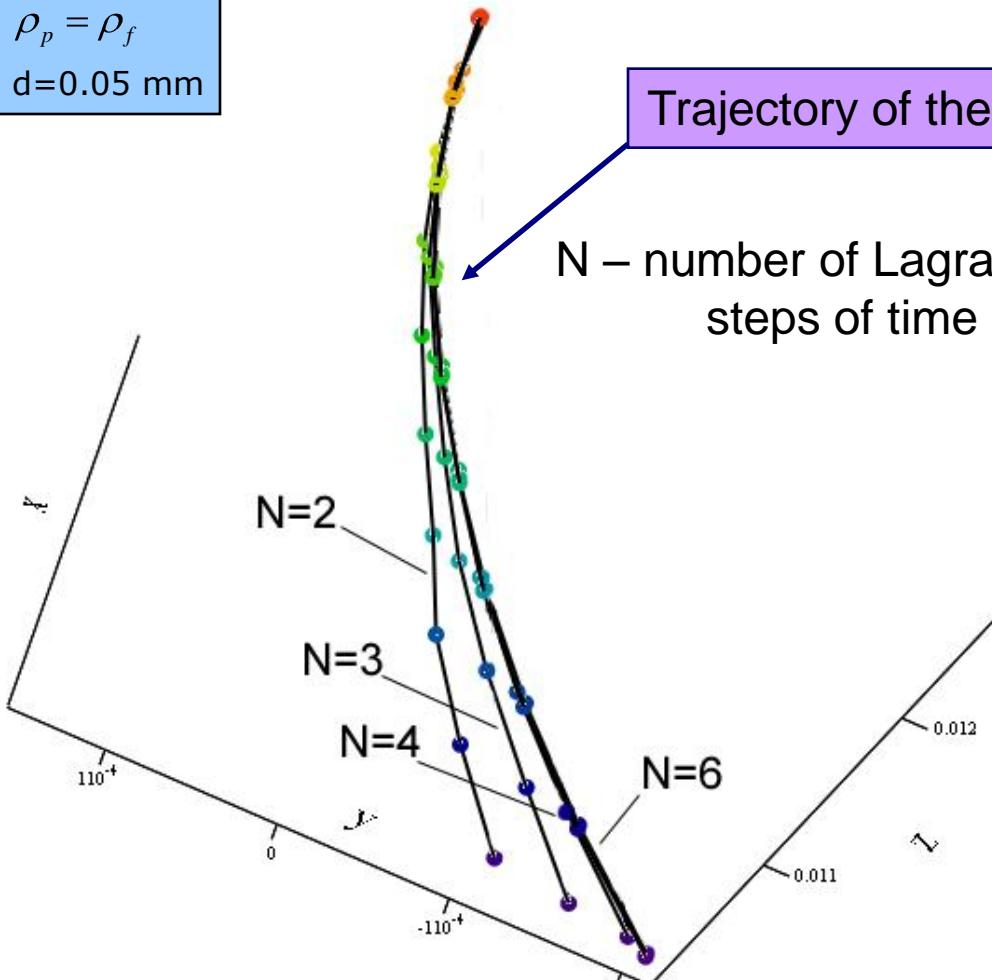
* $C_D(\mathbf{U})$ – Schiller & Naumann, 1933; C_A – Odar & Hamilton, 1964; $C_L(\mathbf{U})$ – Legendre & Magnaudet, 1997

The Development of *solidParticle* Library

$\rho_p = \rho_f$
 $d=0.05 \text{ mm}$

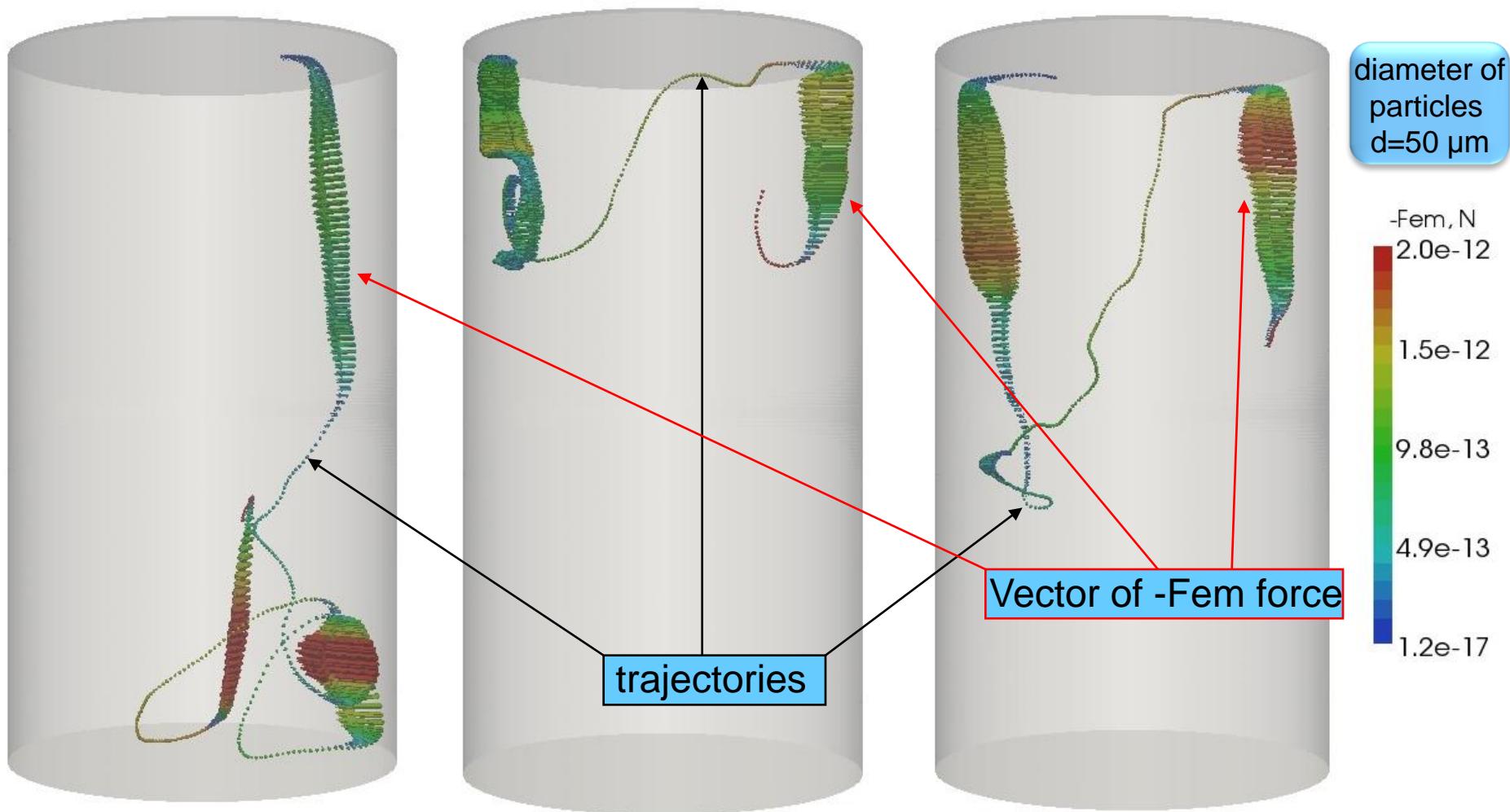
Trajectory of the particle for different LST

N – number of Lagrangian
steps of time (LST)



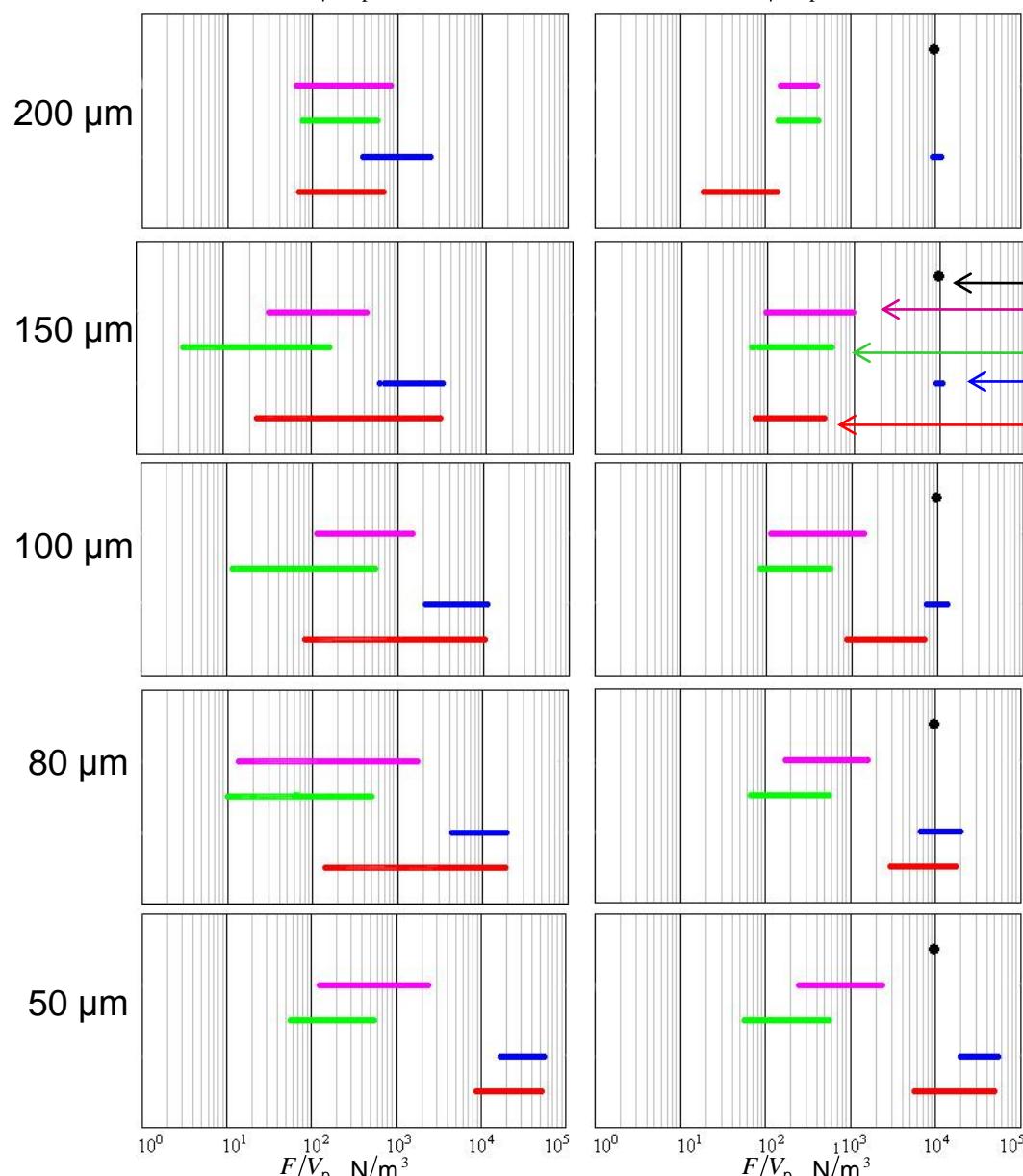
Results and Analysis of the Model

EM force



$$\rho_f/\rho_p = 1.0$$

$$\rho_f/\rho_p = 1.5$$



Results and Analysis of the Model

Buoyancy force
Added mass force
Acceleration force
Drag force
Lift force

$S = \rho_f/\rho_p = 1: d < 80 \mu\text{m}$
- drag, lift and M forces;

$S = 1.5: d \leq 100 \mu\text{m}$
- drag, lift, buoyancy and EM forces;

$S = 1.5: d > 100 \mu\text{m}$
- drag, buoyancy and EM forces;

M. Ščepanskis, A. Jakovičs, E. Baake J. Phys.: Conf. Ser. Forthcoming 4th quarter 2011

Conclusions

LES-based Euler-Lagrange model is created for non-conductive particles in EM induced flow inside the metallurgical furnaces by development of solidParticle library and pisoFoam solver.

The model takes into account drag, buoyancy, EM, lift, acceleration and added mass forces.

Key publications:

- A. Umbrashko et. al. Met. Trans. B 37B 831-8
- M. Ščepanskis et. al. Magnetohydrodynamics 46(4) 413-23
- M. Ščepanskis et. al. J. Phys.: Conf. Ser. Forthcoming 4th quarter 2011

Thank you for attention



**LATVIJAS
UNIVERSITĀTE**
ANNO 1919

LV-1002, Zelļu iela 8 Rīga, Latvia. tel. +371 67033780
<http://www.modlab.lv>

University of Latvia

Faculty of Physics and
Mathematics

Department of Physics

Laboratory for Mathematical Modeling of
Environmental and Technological
Processes



111
102
1004

Leibniz
Universität
Hannover

D-30167, Wilhelm-Busch-Str. 4, Germany
<http://www.etp.uni-hannover.de>

**Gottfried Wilhelm Leibnitz
University of Hanover**

Faculty of Electrical
Engineering and Computer
Science
Institute of Electrotechnology