Finite Element-Fictitious Boundary Methods for the Numerical Simulation of Complex Particulate Flows

Stefan Turek, Raphael Münster, Otto Mierka
Institut für Angewandte Mathematik, TU Dortmund
http://www.mathematik.tu-dortmund.de/LS3
http://www.featflow.de
Basic Flow Solver

Basic CFD tool – **FeatFlow**
(robust, parallel, efficient)

**HPC features:**
- Massively parallel
- GPU computing
- Open source

**Non-Newtonian flow module:**
- generalized Newtonian model
  (Power-law, Carreau, ... etc.)
- viscoelastic model
  (Giesekus, Oldroyd B, ... etc.)

**Multiphase flow module (resolved interfaces):**
- \( l/l \) – interface tracking (Level Set)
- \( g/l \) – interface capturing (FBM)
- \( g/l/l \) – combination of \( l/l \) and \( g/l \)

**Numerical features:**
- Higher order Q2P1 FEM schemes
- FCT & EO FEM stabilization techniques
- Use of unstructured meshes
- **Fictitious Boundary (FBM) methods**
- Dynamic adaptive grid deformation
- Newton-Multigrid solvers

**Engineering aspects:**
- Geometrical design
- Modulation strategy
- Optimization

FEM-based simulation tools for the accurate prediction of multiphase flow problems, particularly with **liquid-solid interfaces**
Consider the flow of $N$ solid particles in a fluid with density $\rho$ and viscosity $\mu$. Denote by $\Omega_f(t)$ the domain occupied by the fluid at time $t$, by $\Omega_i(t)$ the domain occupied by the $i$th-particle at time $t$ and let $\Omega = \Omega_f \cup \Omega_i$.

The fluid flow is modelled by the Navier-Stokes equations:

\[ \rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) - \nabla \cdot \sigma = f, \quad \nabla \cdot u = 0 \]

where $\sigma$ is the total stress tensor of the fluid phase:

\[ \sigma(X,t) = -pI + \mu [\nabla u + (\nabla u)^T] \]
The motion of particles can be described by the Newton-Euler equations. A particle moves with a translational velocity $U_i$ and angular velocity $\omega_i$ which satisfy:

\[
M_i \frac{dU_i}{dt} = F_i + F'_i + (\Delta M_i) g, \quad I_i \frac{d\omega_i}{dt} + \omega_i \times (I_i\omega_i) = T_i,
\]

- $M_i$ : mass of the $i$-th particle ($i=1,...,N$)
- $I_i$ : moment of inertia tensor of the $i$-th particle
- $\Delta M_i$ : mass difference between $M_i$ and the mass of the fluid
- $F_i$ : hydrodynamic force acting on the $i$-th particle
- $T_i$ : hydrodynamic torque acting on the $i$-th particle
Equations of Motion (II)

The position and orientation of the i-th particle are obtained by integrating the kinematic equations:

\[
\frac{dX_i}{dt} = U_i, \quad \frac{d\theta_i}{dt} = \omega_i, \quad \frac{d\omega_i}{dt} = l_i^{-1}T_i
\]

which can be done numerically by an explicit Euler scheme:

\[
X_i^{n+1} = X_i^n + \Delta t U_i^n \quad \omega_i^{n+1} = \omega_i^n + \Delta t (l_i^{-1}T_i^n) \quad \theta_i^{n+1} = \theta_i^n + \Delta t \omega_i^n
\]

Boundary Conditions

We apply the velocity \( u(X) \) as no-slip boundary condition at the interface \( \partial \Omega_i \) between the i-th particle and the fluid, which for \( X \in \Omega_i \) is defined by:

\[
u(X) = U_i + \omega_i \times (X - X_i)
\]
Fictitious Boundary Method

Eulerian Approach:
• Internal objects are represented as a boolean (in/out) function on the mesh
• Use of a fixed mesh possible
• Complex shapes are possible (surface triangulation, implicit functions)
• Higher accuracy possible by using mesh adaptation techniques
Hydrodynamic Forces

Hydrodynamic force and torque acting on the i-th particle

\[
F_i = -\int_{\partial \Omega} \sigma \cdot n_i \, d\Gamma_i, \quad T_i = -\int_{\partial \Omega} (X - X_i) \times (\sigma \cdot n_i) \, d\Gamma
\]

Force Calculation with Fictitious Boundary Method

The FBM can only decide:

- `INSIDE` (1) and `OUTSIDE` (0)
- Only first order accuracy

Alternative:

Replace the surface integral by a volume integral
Define an *indicator function* $\alpha_i$:

$$
\alpha_i(X) = \begin{cases} 
1 & \text{for } X \in \Omega_i \\
0 & \text{for } X \in \Omega_f
\end{cases}
$$

**Remark:** The gradient of $\alpha_i$ is zero everywhere except at the surface of the $i$-th Particle and approximates the normal vector (in a weak sense), allowing us to write:

$$
F_i = -\int_{\Omega_T} \sigma \cdot \nabla \alpha_i \, d\Omega, \\
T_i = -\int_{\Omega_T} (X - X_i) \times (\sigma \cdot \nabla \alpha_i) \, d\Omega
$$

On the finite element level we can compute this by:

$$
F_i = -\sum_{T \in T_{h,i}} \int_{\Omega_T} \sigma_h \cdot \nabla \alpha_{h,i} \, d\Omega, \\
T_i = -\sum_{T \in T_{h,i}} \int_{\Omega_T} (X - X_i) \times (\sigma_h \cdot \nabla \alpha_{h,i}) \, d\Omega
$$

$\alpha_{h,i}(x)$: finite element interpolant of $\alpha(x)$

$T_{h,i}$: elements intersected by $i$-th particle
Integration over $\Omega_T$ too expensive:
- Gradient is non-zero on $\partial\Omega_i$
- Information available from FBM
- Evaluate boundary cells only
- Visit each cell only once

\[ F_i = - \sum_{T \in h_i} \int_{\Omega_T} \sigma \cdot \nabla \alpha_{h_i} d\Omega, \]

\[ T_i = - \sum_{T \in h_i} \int_{\Omega_T} (X - X_i) \times (\sigma \cdot \nabla \alpha_{h,i}) d\Omega \]
Numerical Solution Scheme

Solve for velocity and pressure applying FBM-conditions

\[ \text{NSE} \left( u_f^{n+1}, p^{n+1} \right) = \text{BC} \left( \Omega_i^n, u_i^n \right) \]

Calculate hydrodynamic force, torque and apply

\[ F_{i, n+1}, T_{i, n+1} \]

Contact force calculation

\[ F_{C_i}^{n+1} \]

Compute new velocity and angular velocity

\[ u_i^{n+1} = u_i^n + \Delta t \left( F_{C_i}^{n+1} / M_i \right) \omega_i^{n+1} = \omega_i^n + \Delta t \omega_i^{n-1} (r \times F_{c,i}) \]

Position update

\[ X_i^{n+1} = X_i^n + \Delta t u_i^n, \theta_i^{n+1} = \theta_i^n + \Delta t \omega_i^n \]
Grid Deformation Method

Further improvement via adaptive Grid Deformation which preserves the (local) logical structure (→ GPU)
Contact Force Calculation

- Contact force calculation realized as a three step process
  → Broadphase
  → Narrowphase
  → Contact/Collision force calculation
- Worst case complexity for collision detection is $O(n^2)$
  → Computing contact information is expensive
  → Reduce number of expensive tests → Broad Phase
- **Broad phase**
  → Simple rejection tests exclude pairs that cannot intersect
  → Use hierarchical spatial partitioning
- **Narrow phase**
  → Uses Broadphase output
  → Calculates data neccessary for collision force calculation
For a single pair of colliding bodies we compute the impulse $f$ that causes the velocities of the bodies to change:

$$f = \frac{-(1 + \varepsilon)(n_1(v_1 - v_2) + \omega_1(r_{11} \times n_1) - \omega_2(r_{12} \times n_1))}{m_1^{-1} + m_2^{-1} + (r_{11} \times n_1)^T I_1^{-1}(r_{11} \times n_1) + (r_{12} \times n_1)^T I_2^{-1}(r_{12} \times n_1)}$$

Using the impulse $f$, the change in linear and angular velocity can be calculated:

$$v_1(t + \Delta t) = v_1(t) + \frac{fn_1}{m_1}, \quad \omega_1(t + \Delta t) = \omega_1(t) + I_1^{-1}(r_{11} \times fn_1)$$

$$v_2(t + \Delta t) = v_2(t) - \frac{fn_1}{m_2}, \quad \omega_2(t + \Delta t) = \omega_2(t) - I_2^{-1}(r_{12} \times fn_1)$$
In the case of multiple colliding bodies with $K$ contact points the impulses influence each other. Hence, they are combined into a system of equations that involves the following matrices and vectors:

- $N$: matrix of contact normals
- $C$: matrix of contact conditions
- $M$: rigid body mass matrix
- $f$: vector of contact forces ($f_i \geq 0$)
- $f^{ext}$: vector of external forces (gravity, etc.)

\[
\begin{align*}
N^T C^T M^{-1} C N \cdot \Delta f^{t+\Delta t} + N^T C^T (u^t + \Delta t M^{-1} + f^{ext}) & \geq 0, \quad f \geq 0 \\
A & \quad x & b
\end{align*}
\]

A problem of this form is called a linear complementarity problem (LCP) which can be solved with efficient iterative methods like the Projected Gauss-Seidel solver (PGS).

Kenny Erleben, *Stable, Robust, and Versatile Multibody Dynamics Animation*
Examples

- Flows with complex geometries
- Fluidized bed
- Particulate flow demonstrating incompressibility
- GPU sedimentation example
- Numerical results and benchmark test cases
- Comparison of results with other groups
Benchmarking and Validation (I)

Free fall of particles:
- Terminal velocity
- Different physical parameters
- Different geometrical parameters

\[ d_s = 0.3, \quad \rho_s = 1.14 \]

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( U_{\text{flow}} )</th>
<th>( U_{\text{exp}} )</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>5.885</td>
<td>6.283</td>
<td>6.33</td>
</tr>
<tr>
<td>0.05</td>
<td>4.133</td>
<td>3.972</td>
<td>4.05</td>
</tr>
<tr>
<td>0.1</td>
<td>2.588</td>
<td>2.426</td>
<td>6.66</td>
</tr>
<tr>
<td>0.2</td>
<td>1.492</td>
<td>1.401</td>
<td>6.50</td>
</tr>
</tbody>
</table>

\[ d_s = 0.2, \quad \rho_s = 1.14 \]

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( U_{\text{flow}} )</th>
<th>( U_{\text{exp}} )</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02</td>
<td>4.370</td>
<td>4.334</td>
<td>0.83</td>
</tr>
<tr>
<td>0.05</td>
<td>2.699</td>
<td>2.489</td>
<td>8.44</td>
</tr>
<tr>
<td>0.1</td>
<td>1.649</td>
<td>1.552</td>
<td>6.25</td>
</tr>
<tr>
<td>0.2</td>
<td>0.946</td>
<td>0.870</td>
<td>8.74</td>
</tr>
</tbody>
</table>

\[ d_s = 0.3, \quad \rho_s = 1.02 \]

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( U_{\text{flow}} )</th>
<th>( U_{\text{exp}} )</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>2.167</td>
<td>2.107</td>
<td>2.84</td>
</tr>
<tr>
<td>0.02</td>
<td>1.495</td>
<td>1.436</td>
<td>4.11</td>
</tr>
<tr>
<td>0.05</td>
<td>0.809</td>
<td>0.749</td>
<td>7.81</td>
</tr>
<tr>
<td>0.1</td>
<td>0.402</td>
<td>0.404</td>
<td>0.44</td>
</tr>
<tr>
<td>0.2</td>
<td>0.218</td>
<td>0.216</td>
<td>1.02</td>
</tr>
</tbody>
</table>

\[ d_s = 0.2, \quad \rho_s = 1.02 \]

<table>
<thead>
<tr>
<th>( \nu )</th>
<th>( U_{\text{flow}} )</th>
<th>( U_{\text{exp}} )</th>
<th>Relative error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>1.4660</td>
<td>1.4110</td>
<td>3.90</td>
</tr>
<tr>
<td>0.02</td>
<td>0.9998</td>
<td>0.9129</td>
<td>9.52</td>
</tr>
<tr>
<td>0.05</td>
<td>0.4917</td>
<td>0.4603</td>
<td>6.82</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2637</td>
<td>0.2571</td>
<td>2.57</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1335</td>
<td>0.1317</td>
<td>1.37</td>
</tr>
</tbody>
</table>

Source: Glowinski et al. 2001
**Benchmarking and Validation (II)**

**Settling of a sphere towards a plane wall:**
- Sedimentation Velocity
- Particle trajectory
- Kinetic Energy
- Different Reynolds numbers

**Setup**

**Computational mesh:**
- 1,075,200 vertices
- 622,592 hexahedral cells
- Q2/P1:
  \[ \rightarrow 50,429,952 \text{ DoFs} \]

**Hardware Resources:**
- 32 Processors
**Observations**

- Velocity profiles compare well to ten Cate's data
- Maximum velocity close to experiment
- Flow features are accurately resolved

<table>
<thead>
<tr>
<th>$Re$</th>
<th>$u_{\text{max}}/u_\infty$</th>
<th>$u_{\text{max}}/u_\infty$</th>
<th>$u_{\text{max}}/u_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ten Cate</td>
<td>exp</td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>0.945</td>
<td>0.894</td>
<td>0.947</td>
</tr>
<tr>
<td>4.1</td>
<td>0.955</td>
<td>0.950</td>
<td>0.953</td>
</tr>
<tr>
<td>11.6</td>
<td>0.953</td>
<td>0.955</td>
<td>0.959</td>
</tr>
<tr>
<td>31.9</td>
<td>0.951</td>
<td>0.947</td>
<td>0.955</td>
</tr>
</tbody>
</table>

Tab. 1 Comparison of the $u_{\text{max}}/u_\infty$ ratios between the FEM-FBM, ten Cate's simulation and ten Cate's experiment
Comparison of FEM-FBM and the experimental values and the LBM results of the group of Sommerfeld

Source: 13th Workshop on Two-Phase Flow Predictions 2012
Acknowledgements: Ernst, M., Dietzel, M., Sommerfeld, M.
Multi-level Analysis

FEM-Multigrid Framework

- Increasing the mesh resolution produces more accurate results
  Test performed at different mesh levels
  - Maximum velocity is approximated better ✓
  - Shape of the velocity curve matches better ✓
Fluidized Bed Example
DGS Configuration
Large-Scale Examples
Extensions & Future Activities

Fluidics

- Viscoelastic fluids
- Multiphase problems
  → Liquid-Liquid-Solid
  → Liquid-Gas-Solid

Hardware-Oriented Numerics

- Improve parallel efficiency of collision force computation
- Further develop collision detection and collision force computation on GPUs
Fluid Prilling and Encapsulation (I)

- Numerical simulation of micro-fluidic drug encapsulation ("monodisperse compound droplets")
- Polymeric “bio-degradable” outer fluid with generalized Newtonian behaviour
- Optimization w.r.t. boundary conditions, flow rates, droplet size, geometry

Jet Configuration

- Core material is defined as the specific material that requires to be coated (liquid, emulsion, colloid or solid)
- Shell material is present to protect and stabilize the core (Alginate, Chitosan, Gelatin, Pectin, Waxes, Starch)

In Pharmaceutics

- Controlled drug release
- Protection of chemically active ingredients (from both sides)
- Protection against shear stress in stirred reactors
- Protection against evaporation
- Taste or odor masking
**Fluid Prilling and Encapsulation (II)**

<table>
<thead>
<tr>
<th>mgLS(^{(2)})-FBM-FEM flow module</th>
<th>Tasks related to code development</th>
<th>Tasks related to application</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Multiple Level Set fields for simulation of liquid core encapsulation - l/l/g</td>
<td>• Validation via experimental results</td>
</tr>
<tr>
<td></td>
<td>• Fictitious boundary method for particle encapsulation - s/l/g</td>
<td>• Modulation for monodisperse compound drops</td>
</tr>
</tbody>
</table>

Aqueous solutions of alginates have shear-thinning characteristics.

Preliminary simulation results for encapsulation of solid particles.

---

**Ketoprofen/Ketoprofen Lysinate core**

**Alginate shell**