FEM Techniques for Multiphase Flow Simulation

Stefan Turek
Institut für Angewandte Mathematik, Univ. Dortmund

http://www.mathematik.uni-dortmund.de/LS3
http://www.featflow.de

- FEM discretization and solution techniques
- Gas-liquid configurations
- Liquid-solid configurations
Incompressible Navier-Stokes Equations

\[ \rho(u_t + u \cdot \nabla u) = f + \mu \Delta u - \nabla p \quad , \quad \nabla \cdot u = 0 \]

\[ + \]

‘Complex’ Extensions for Multiphase Flows

⇒ **Discretization:** ‘Find \( u_h \in V_h \): \( a_h(u_h, v_h) = (f, v_h) \quad \forall v_h \in V_h \)’

⇒ **Solver:** ‘Solve (nonlinear) System \( A \cdot X = F \)’

⇒ **Software:** ‘**Fea**T**Flow**/**Fea**s**T**Flow’
1) **BB Condition:**
\[
\min_{q_h \in L_h} \max_{v_h \in V_h} \frac{(q_h, \nabla \cdot v_h)}{\|v_h\|_{1,h} \|q_h\|_0} \geq \alpha > 0
\]

→ Deformed and anisotropic meshes ?
→ Quantitative (nonlinear) stabilization strategies ?

2) **Korn’s Inequality:**
\[
\|D(v_h)\|_0 \geq \beta \|v_h\|_{1,h}
\]

→ For ‘discontinuous’ FEM ?

3) **Stabilization of the convective term** ‘\(u_h \cdot \nabla u_h\)’

→ Upwind/Streamline-Diffusion for ‘diffusion-dominated’ case !
→ FEM-Stabilization for pure ‘transport-dominated’ case ?

4) **Higher Order FEM Spaces w.r.t. 1), 2), 3)**
Discretization in Time

1) Second Order Accurate

2) (Semi) Implicit + (Strongly) A-Stable
   → Allowing huge time steps for (quasi-) stationary flow!
   → No CFL-type restrictions! ← local mesh adaptivity

3) Accuracy-Based (Implicit) Error Indicator
   → Based on local truncation error for all physical quantities!

4) Rigorous Error Control
   → Residual-based a posteriori error control via dual problem?
   → For user-defined time/space-averaged quantities?
Aspects of ‘Fast Solvers’

1) Newton Schemes:  \[ U^{n+1} = U^n - \tilde{F}_\delta(U^n)^{-1} F(U^n) \]
   → Frechet derivative \( F_\delta(U^n) \)?  Simplifications \( \tilde{F}_\delta(U^n) \)?
   → Solvers for linear problems?  Dependence on parameters (Re,...) ?

2) Linear Multigrid/DD Schemes:  \[ A_{lev} U_{lev} = F_{lev} \]
   → Robustness!  Efficiency!  Parallelism!
   → Re ?  Mesh size/deformations?  FEM types?

3) (De)Coupling:  \[ U^{n+1} = f(P^n), P^{n+1} = g(U^{n+1}) \]
   → Coupling of NSE with other components?
   → Decoupling of pressure from velocity/temperature/etc.?  
   ⇒ Pressure Schur Complement (PSC) solvers!
Key Ideas of MPSC Approaches

‘Re-interpretation of Navier-Stokes solvers (Chorin, Van Kan, Uzawa, etc.) as "incomplete solvers" for discrete saddle-point problems’

LOCAL MPSC (‘Multilevel Pressure Schur Complement’):

‘Fully coupled Newton-like solver as outer nonlinear procedure’
‘Solve "exactly" on "subsets/patches" and perform an outer Block–Gauß-Seidel/Jacobi iteration as smoother’
⇒ For (quasi-) stationary flow with ”large” time steps

GLOBAL MPSC (‘Multilevel Pressure Schur Complement’):

‘Outer (multigrid) coupling of velocity and pressure’
‘Newton/Multigrid solver for all scalar subproblems’
⇒ For highly nonstationary flow
Parallel Realization of GLOBAL MPSC

Dynamic simulations for ‘Channel Flow around Cylinder’ with 32 Mill. d.o.f.s for 6.500 implicit time steps (20 sec real time)

inflow boundary

outflow boundary

250 Mill. d.o.f.s are possible on LINUX clusters!
## LOCAL MPSC: Flow around Cylinder

### Nonlinear ‘Power-Law’ model

<table>
<thead>
<tr>
<th>Lev</th>
<th>Drag</th>
<th>Lift</th>
<th>$\Delta p$</th>
<th>Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.9475 + 03</td>
<td>0.3453 + 01</td>
<td>16.02</td>
<td>NNL/AVGMRES</td>
</tr>
<tr>
<td>4</td>
<td>0.9534 + 03</td>
<td>0.3957 + 01</td>
<td>15.82</td>
<td>20/94</td>
</tr>
<tr>
<td>5</td>
<td>0.9569 + 03</td>
<td>0.4069 + 01</td>
<td>15.86</td>
<td>39/186</td>
</tr>
</tbody>
</table>

### Nonconforming $\tilde{Q}_1 - Q_0$ FEM

<table>
<thead>
<tr>
<th>Lev</th>
<th>Drag</th>
<th>Lift</th>
<th>$\Delta p$</th>
<th>Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.9160 + 03</td>
<td>0.3738 + 01</td>
<td>15.74</td>
<td>NNL/AVMGM</td>
</tr>
<tr>
<td>5</td>
<td>0.9351 + 03</td>
<td>0.3995 + 01</td>
<td>15.82</td>
<td>13/2</td>
</tr>
<tr>
<td>6</td>
<td>0.9462 + 03</td>
<td>0.4059 + 01</td>
<td>15.85</td>
<td>13/2</td>
</tr>
</tbody>
</table>

### Continuous Newton + Multigrid Solvers !!!
‘Exploit hierarchical, but locally regular structures !!!’

I) Patch-oriented adaptivity

‘Many’ tensor product grids
‘Few’ unstructured grids

II) Generalized MG-DD solver: SCAR C

Exploit locally ‘regular’ structures (efficiency)
Recursive ‘clustering’ of anisotropies (robustness)
‘Strong local solvers improve global convergence !’
Concepts for Adaptive Meshing

1) macro-oriented adaptivity

2) patchwise ‘deformation’ adaptivity

3) (patchwise) ‘local’ adaptivity
**Example for SCApRC**

Parallel convergence rates for SCApRC-CG (2 global smoothing, V-cycle; 2 local ‘MGTRI’, F-cycle) with direct coarse mesh solver

### Global (parallel) convergence rates

<table>
<thead>
<tr>
<th>#NEQ-global</th>
<th>$AR$</th>
<th>$\rho$ (#IT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,704</td>
<td>3</td>
<td>0.03 (5)</td>
</tr>
<tr>
<td>44,544</td>
<td>3</td>
<td>0.03 (5)</td>
</tr>
<tr>
<td>431,317</td>
<td>10</td>
<td>0.09 (6)</td>
</tr>
<tr>
<td>1,722,773</td>
<td>10</td>
<td>0.08 (6)</td>
</tr>
<tr>
<td>9,344,533</td>
<td>10</td>
<td>0.09 (6)</td>
</tr>
<tr>
<td>37,366,805</td>
<td>$10^7$</td>
<td>0.09 (6)</td>
</tr>
</tbody>
</table>

### Local MFLOP/s rates (AMD ‘XP 1500+’)

<table>
<thead>
<tr>
<th>#NEQ-local</th>
<th>MV-V/C</th>
<th>MGTRI-V/C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$65^2$</td>
<td>245/1275</td>
<td>299/706</td>
</tr>
<tr>
<td>$257^2$</td>
<td>193/638</td>
<td>173/382</td>
</tr>
<tr>
<td>$1025^2$</td>
<td>168/614</td>
<td>146/288</td>
</tr>
</tbody>
</table>

Cells $\text{lnmin}$

Cells $\text{icappaav}$

- p.11/38
GAS-LIQUID REACTORS

LIQUID-SOLID PARTICULATED FLOW
Drift-Flux model with mass transfer and chemical reaction
(Gas holdup $\epsilon$, Number density $n$, Mass flux $N$, Interfacial area $a_S$)

$$\frac{\partial \mathbf{v}_L}{\partial t} + \mathbf{v}_L \cdot \nabla \mathbf{v}_L = -\nabla P + \nu \Delta \mathbf{v}_L - \epsilon \mathbf{g}, \quad \nabla \cdot \mathbf{v}_L = 0, \quad \mathbf{v}_G = \mathbf{v}_L + \mathbf{v}_{\text{slip}} + \mathbf{v}_{\text{drift}}$$
Gas-Liquid Reactors

Drift-Flux model with mass transfer and chemical reaction
(Gas holdup $\epsilon$, Number density $n$, Mass flux $N$, Interfacial area $a_S$)

\[
\frac{\partial v_L}{\partial t} + v_L \cdot \nabla v_L = -\nabla P + \nu \Delta v_L - \epsilon g , \quad \nabla \cdot v_L = 0 , \quad v_G = v_L + v_{\text{slip}} + v_{\text{drift}}
\]

\[
\frac{\partial \tilde{\rho}_G}{\partial t} + \nabla \cdot (\tilde{\rho}_G v_G) = -a_S m_\mu N , \quad N = Ek_L^0 (p/H - c_A)
\]

\[
\frac{\partial n}{\partial t} + \nabla \cdot (n v_G) = 0 , \quad \epsilon = \frac{\tilde{\rho}_G RT}{p m_\mu} , \quad a_S = (4\pi n)^{1/3}(3\epsilon)^{2/3}
\]

\[
\frac{\partial \tilde{c}_A}{\partial t} + \nabla \cdot (\tilde{c}_A v_L) = \nabla \cdot (\tilde{D}_A \nabla c_A) - \tilde{k}_2 c_A c_B + a_S N
\]

\[
\frac{\partial \tilde{c}_B}{\partial t} + \nabla \cdot (\tilde{c}_B v_L) = \nabla \cdot (\tilde{D}_B \nabla c_B) - \nu_B \tilde{k}_2 c_A c_B
\]

\[
\frac{\partial \tilde{c}_P}{\partial t} + \nabla \cdot (\tilde{c}_P v_L) = \nabla \cdot (\tilde{D}_P \nabla c_P) + \nu_P \tilde{k}_2 c_A c_B
\]
Properties of Drift-Flux Simulations

No explicit tracking of bubbles via Euler-Euler formulation

- Modelling of ‘small-scale’ effects (single bubbles)?
- Improved Two-Fluid model?
- Two-Phase turbulence?  Population Balance models?
Extension I: Two-Phase Turbulence

‘Effective’ viscosity \((k - \varepsilon \text{ model})\)

\[
\nu_{\text{eff}} = \nu + \nu_T, \quad \nu_T = C_\mu \frac{k^2}{\varepsilon} + C_g \varepsilon |\mathbf{u}_{\text{slip}}| r
\]

with

\[
k = \frac{1}{2} \langle |\mathbf{u}'|^2 \rangle
\]

\[
\varepsilon = \frac{\nu}{2} \langle |\nabla \mathbf{u}' + (\nabla \mathbf{u}')^T|^2 \rangle
\]

turbulent kinetic energy dissipation rate

\[
\frac{\partial k}{\partial t} + \nabla \cdot \left( k \mathbf{u} - \frac{\nu_T}{\sigma_k} \nabla k \right) = P_k + S_k - \varepsilon
\]

\[
\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \left( \varepsilon \mathbf{u} - \frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) = \frac{\varepsilon}{k} (C_1 P_k + C_\varepsilon S_k - C_2 \varepsilon)
\]

Modelling of production terms

\[
P_k = \frac{\nu_T}{2} |\nabla \mathbf{u} + \nabla \mathbf{u}^T|^2 \quad \text{shear induced turbulence}
\]

\[
S_k = -C_k \varepsilon \nabla p \cdot \mathbf{u}_{\text{slip}} \quad \text{bubble induced turbulence}
\]


**Extension II: Population Balance Models**

\[ \frac{\partial f}{\partial t} + \nabla \cdot (f \, u_G) + \frac{\partial}{\partial m} (f \, \dot{m}) = Q - S \]

- \( f = f(x, m, t) \) \quad Bubble size distribution
- \( u_G = u + u_{\text{slip}}(m) \) \quad Gas velocity
- \( \dot{m} = E k_L^0 (c_A - \frac{p}{H}) \eta a_B \) \quad Mass transfer rate
- \( Q - S \) \quad Coalescence and breakup rates

Calculation of the local gas holdup

\[ \epsilon(x, t) = \frac{RT}{p \eta} \int_{0}^{\infty} m \, f(x, m, t) \, dm \]

**Treatment of Integro-Differential equation ???

**Coupling with CFD ???**
Extension III: Forces in Two-Fluid Model

\[
\tilde{\rho}_G \left[ \frac{\partial \mathbf{u}_G}{\partial t} + \mathbf{u}_G \cdot \nabla \mathbf{u}_G \right] = \epsilon \nabla \cdot \mathbf{S}_G + \tilde{\rho}_G \mathbf{g} + \mathbf{f}_{\text{int}}
\]

\[
\tilde{\rho}_L \left[ \frac{\partial \mathbf{u}_L}{\partial t} + \mathbf{u}_L \cdot \nabla \mathbf{u}_L \right] = (1 - \epsilon) \nabla \cdot \mathbf{S}_L + \tilde{\rho}_L \mathbf{g} - \mathbf{f}_{\text{int}}
\]

with: \( \mathbf{f}_{\text{int}} = \mathbf{f}_D + \mathbf{f}_{VM} + \mathbf{f}_L \)

\[
\mathbf{f}_D = -\epsilon C_D \frac{3 \rho_L}{8} \frac{\mathbf{r}}{|\mathbf{u}_G - \mathbf{u}_L|} (\mathbf{u}_G - \mathbf{u}_L) \quad (\text{drag force})
\]

\[
\mathbf{f}_{VM} = -\epsilon C_{VM} \rho_L \left( \frac{\mathbf{d}\mathbf{u}_G}{\mathbf{d}t} - \frac{\mathbf{d}\mathbf{u}_L}{\mathbf{d}t} \right) \quad (\text{virtual mass force})
\]

\[
\mathbf{f}_L = -\epsilon C_L \rho_L (\mathbf{u}_G - \mathbf{u}_L) \times (\nabla \times \mathbf{u}_L) \quad (\text{lift force})
\]
‘ADMIRE Benchmark’ (cf. Pfleger/BASF)

⇒ Full Model !?
Extension IV: ‘Single Bubble’

\[
\rho_i \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) - \nabla \cdot (2\mu_i \mathbf{S}) + \nabla p = \rho_i \mathbf{g}
\]

\[
\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega_i , \ i = 1, 2, \ldots
\]

\[
\left[ \mathbf{v} \right]_{\Gamma} \cdot \mathbf{n} = 0 , \quad - \left[ -p \mathbf{I} + 2\mu(x) \mathbf{S} \right]_{\Gamma} \cdot \mathbf{n} = \kappa \sigma \mathbf{n}
\]

→ (Implicit) Reconstruction via ‘Indicator function’ \( \phi \)!
**Level-Set vs. VOF Method**

1. "Easy" integration of surface tension (in weak formulation)
2. High order schemes due to high smoothness

→ **Reinitialisation**: Signed distance function

\[
\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \quad , \quad |\nabla \phi| = 1
\]

(Nonstationary) Reformulation:

\[
\frac{\partial \phi}{\partial t} = \text{sign}(\phi_{old})(1 - |\nabla \phi|)
\]

\[
\frac{\partial \phi}{\partial t} + \text{sign}(\phi_{old})|\nabla \phi| = \text{sign}(\phi_{old})
\]

\[
\frac{\partial \phi}{\partial t} + \text{sign}(\phi_{old})\frac{\nabla \phi \cdot \nabla \phi}{|\nabla \phi|} = \frac{\partial \phi}{\partial t} + \mathbf{w} \cdot \nabla \phi = \text{sign}(\phi_{old})
\]
Problems for Discretization

TG for Level Set with/without reinitialisation vs. VOF

→ Oscillation free + highly accurate discretization !!!
Future Challenges

- A posteriori error control + adaptive mesh refinement
- Robust + accurate FEM discretization of higher order
- Efficient solution of (nonlinear) ‘reinitialisation part’
- (Implicit) coupling with CFD part
- Preserve the high efficiency of FEATFLOW!
Liquid-Solid Flow

Development of simulation tools for the understanding of:

1. Interaction of solid particles with flow (→ ‘many complex objects’ !)
2. Behaviour of non-newtonian fluids

Exploit the high efficiency and accuracy of FEATFLOW !
‘Fictitious Boundary Method’ on fixed meshes + Operator-Splitting
The ‘Fictitious Boundary Method’

1. Use (rough) boundary parametrization and locally refined coarse mesh for large-scale structures!
2. Describe fine-scale structures and time-dependent objects via (level-dependent) inner points!
3. Use projectors onto the "right" b.c.’s in iterative components!

Computational mesh independent of ‘internal objects’
Define a function $\alpha$ as

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

**Remark:** $\nabla \alpha_p = 0$ everywhere except at wall surface of the particles, and equal to the normal vector $n_p$ defined on the global grid

$$n_p = \nabla \alpha_p \quad (0)$$

Force acting on the wall surface of the particles can be computed by

$$F_p = \int_{\Gamma_p} \sigma \cdot n_p \, d\Gamma_p = \int_{\Omega_T} \sigma \cdot \nabla \alpha_p \, d\Omega_T$$

with $\Omega_T = \Omega_f \cup \Omega_p$ (analogously for the torque)
‘Flow around Cylinder’ Benchmark

LEVEL 6 ≈ 280.000 elements

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>ch. mesh I</th>
<th>ch. mesh II</th>
<th>ch. mesh I</th>
<th>ch. mesh II</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.5529+01</td>
<td>0.5569+01</td>
<td>0.1216-01</td>
<td>0.2443-03</td>
</tr>
<tr>
<td>4</td>
<td>0.5353+01</td>
<td>0.5575+01</td>
<td>0.1074-01</td>
<td>0.0014-01</td>
</tr>
<tr>
<td>5</td>
<td>0.5427+01</td>
<td>0.5572+01</td>
<td>0.6145-02</td>
<td>0.0812-01</td>
</tr>
<tr>
<td>6</td>
<td>0.5501+01</td>
<td>0.5578+01</td>
<td>0.9902-02</td>
<td>0.1020-01</td>
</tr>
</tbody>
</table>

\[ C_d = 0.55795+01 \]
\[ C_l = 0.10618-01 \]

LEVEL 6 ≈ 150.000 elements

LEVEL 2 ≈ 10.000 elements

<table>
<thead>
<tr>
<th>LEVEL</th>
<th>( C_d )</th>
<th>( C_l )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.55201+01</td>
<td>0.1057-01</td>
</tr>
<tr>
<td>3</td>
<td>0.55759+01</td>
<td>0.1036-01</td>
</tr>
<tr>
<td>4</td>
<td>0.55805+01</td>
<td>0.1041-01</td>
</tr>
</tbody>
</table>
Consider 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 $t$, and by $\Omega_p(t)$ the domain occupied by the particle $p$ at time $t$.

Fluid flow is modeled by the Navier-Stokes equations in $\Omega_f(t)$

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

where $\sigma$ is the total stress tensor in the fluid phase, which is defined as

$$\sigma(X, t) = -p \mathbf{I} + \mu \left[ \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right]$$
Model for Particle Motion

Motion of particles is modeled by the Newton-Euler equations, i.e., the translational velocities $U_p$ and angular velocities $\omega_p$ of the $p$-th particle satisfy

$$
M_p \frac{dU_p}{dt} = F_p + (\Delta M_p) \, g,
$$

$$
I_p \frac{d\omega_p}{dt} + \omega_p \times (I_p \omega_p) = T_p
$$

with $M_p$ the mass of the $p$-th particle ($p = 1, \ldots, N$); $I_p$ the moment of inertia tensor of the $p$-th particle; $\Delta M_p$ the mass difference between the mass $M_p$ and the mass of the fluid occupying the same volume.

$F_p$ and $T_p$ are the hydrodynamical forces and the torque at mass center acting on the $p$-th particle

$$
F_p = \int_{\Gamma_p} \sigma \cdot n_p \, d\Gamma_p, \quad T_p = \int_{\Gamma_p} (X - X_p) \times (\sigma \cdot n_p) \, d\Gamma_p
$$

with $X_p$ the position of the center of gravity of the $p$-th particle, $\Gamma_p = \partial \Omega_p$ the boundary of $p$-th particle, $n_p$ is the unit normal vector on the boundary $\Gamma_p$. 
Interaction between Particle and Fluid

No-slip boundary conditions at interface $\Gamma_p$ between particle and fluid, i.e., for any $X \in \Gamma_p$, the velocity $\mathbf{u}(X)$ is defined by

$$\mathbf{u}(X) = U_p + \omega_p \times (X - X_p)$$

Position $X_p$ of the $p$-th particle and its angle $\theta_p$ are obtained by integration of the kinematic equations

$$\frac{d X_p}{d t} = U_p, \quad \frac{d \theta_p}{d t} = \omega_p$$
The algorithm for $t^n \rightarrow t^{n+1}$ consists of the following 3 substeps:

1. Fluid velocity and pressure: $NSE(u_f^{n+1}, p^{n+1}) = BC(\Omega_p^n, u_p^n)$
2. Calculate hydrodynamic forces: $F_p^{n+1}$
3. Calculate velocity of particles: $u_p^{n+1} = g(F_p^{n+1})$
4. Update position of particles: $\Omega_p^{n+1} = f(u_p^{n+1})$
Rotating Airfoil (cf. Glowinski)
Challenges (for ‘many’ Particles)

- Adaptive time-stepping + locally adaptive grid alignment
- Accurate calculation of forces ($\rightarrow \alpha \in [0, 1]$) for ‘complex’ shapes
- Efficient data structures for treating the particles/particle forces
- (Better) Collision models
- Nonlinear fluids (‘Kissing, Drafting, Thumbling’)
- 3D
- "100.000" particles...
Future Multiphase CFD Tools

Mathematical Key Techniques

(Special) Implicit FEM schemes in space/time, adaptivity/error control, FCT/TVD stabilization, hierarchical Newton/multigrid solvers,...

Hardware-oriented Numerics for PDE, High Performance Computing techniques,...

‘Higher (guaranteed) accuracy with less unknowns via hierarchical solvers with ‘optimal’ numerical complexity while exploiting the available huge sequential/parallel GFLOP/s rates’
Conclusions:

There is a huge potential...

There is much to do...:-)