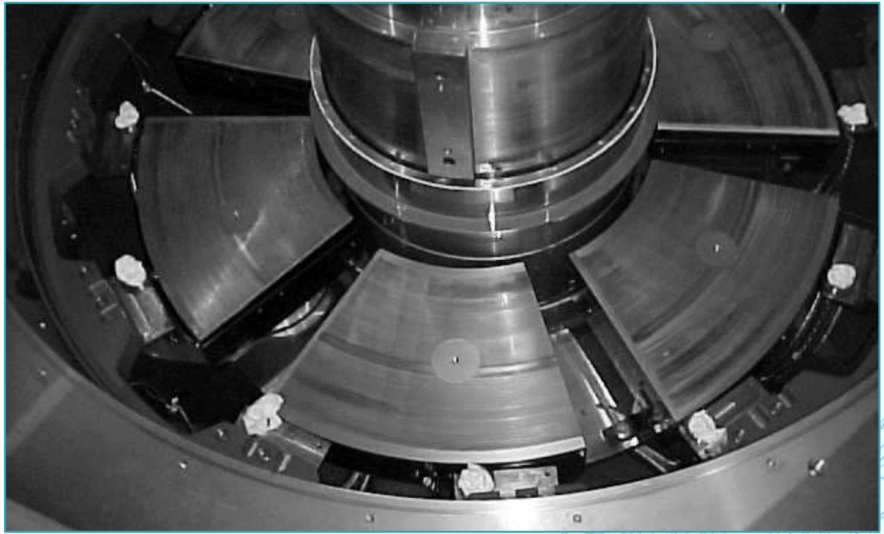
PIMPLE Algorithm and Partitioned FSI Solvers

23.06.2016 ▪ Petr Víta, Wilhelm Brandstätter

DEPARTMENT PETROLEUM ENGINEERING
MONTAN UNIVERSITÄT

Motivation
Hydrodynamic Bearing Simulation

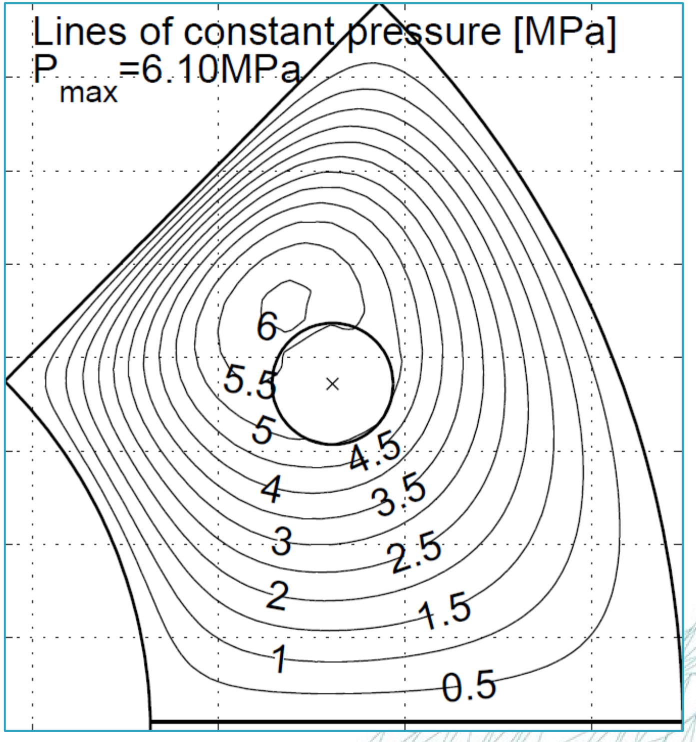
- **Kingsbury tilting pad hydrodynamic thrust bearing**
- *Bearing submerged in the lubricant pool*
- *Support is provided by a thin film of lubricant above pads*
- *Pads tilt and thus form a wedge of lubricant that carries the thrust*



Heinrichson, N., & Santos, I. (2007)

Motivation (2)
Hydrodynamic Bearing Simulation

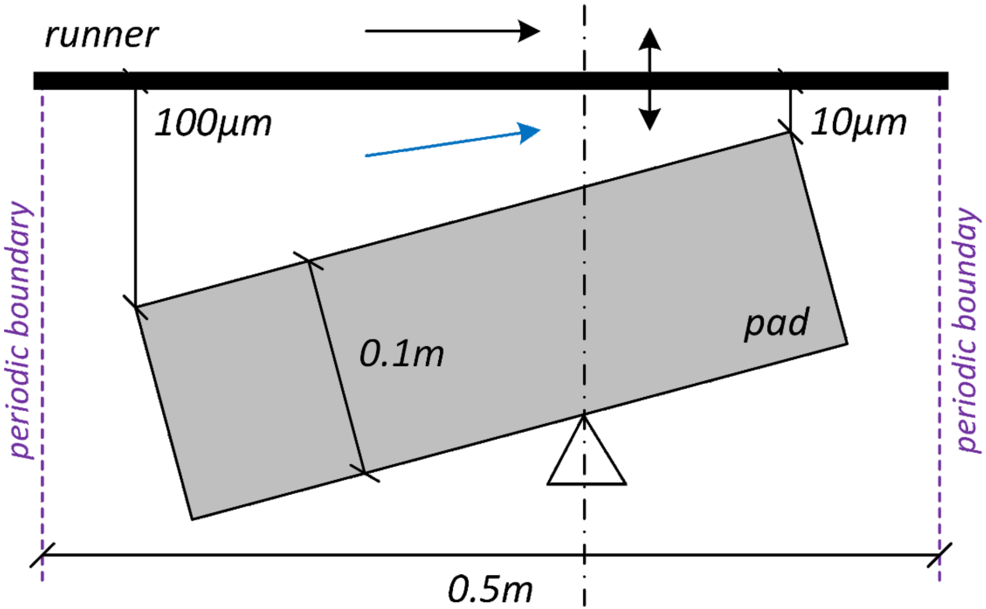
- **Interest in film thickness, pressure, temperature, friction losses etc.**
- **Software typically used for numerical investigations is limited**
- *Steady-state solutions, transient start-up scenarios usually not possible*
- *Investigations are focused on a single pad without its surrounding*



Heinrichson, N., & Santos, I. (2007)

Motivation (3)
Hydrodynamic Bearing Simulation

- **Problem is hiding many extreme parameters**
- *Pad length about half metre*
- *Lubricant wedge height in order of 10-100µm*
- *Laminar and turbulent flow*
- *Pressures up to 10-20MPa*
- *Temperature differences up to 50°C due to viscous dissipation*



FSI Solver
Basic Info

- **Developed solver was a partitioned FSI solver based on an historical icoFsiElasticNonLinSolidFoam**
- *Flow solver used was an adapted rhoPisoFoam with a modified thermophysical model for customer's lubricants*
- *Solid solver in total and updated Lagrangian formulation*
- *Explicit coupling with Aitken adaptive under-relaxation*
- *Conjugate heat transfer between fluid and solid*
- *Additional bearing model (tilting, mirrored runner floating, material model, boundary condition for the pad support, etc.)*

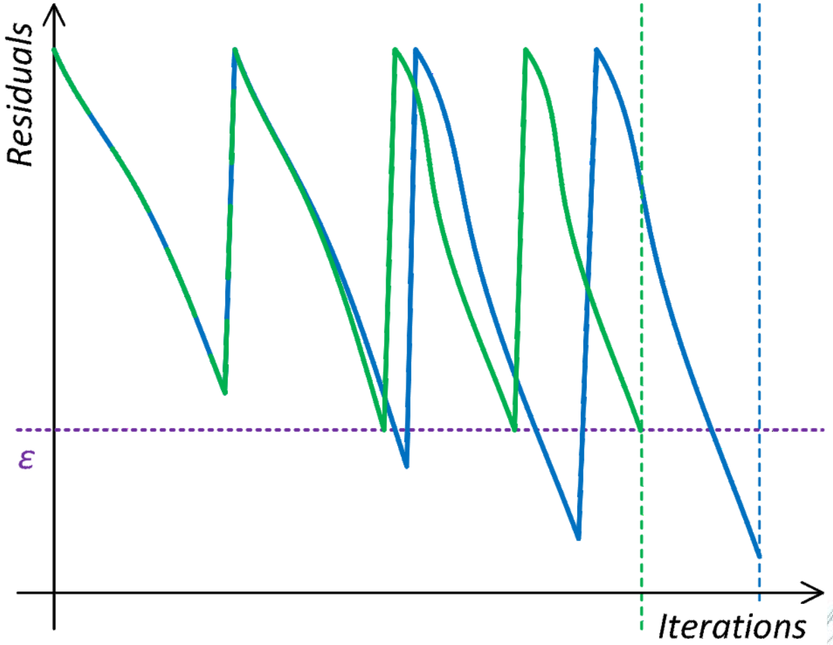
FSI Solver (2)
Stability Problems

- **The length scales differences complicated grid modelling, mesh motion, time stepping and generally solution stability**
- *Extreme value changes where gap displacement of 10µm would induce pressure differences of up to 1 MPa*
- *Mesh distortions especially around the gap area*
- **Aitken under-relaxation stabilized solution during interface search inside a single time step**
- **Transient time advance eventually always crashed**

FSI Solver (3)
PISO Algorithm

- **Used flow solver, rhoPisoFoam, was unfortunately unstable for given meshes and time steps**
- *Minimal Courant number of 10 on the graded grid in the gap*
- *Lowering the cell size leads to the cells count explosion*
- *Lowering the time step leads to instability of the solid solver and/or excessively long simulation times*
- **We tried separate fluid and solid time steps with a limited success, but the idea pointed us into right direction…**

PIMPLE Algorithm
Basics



- Merged PISO and SIMPLE algs.
- PIMPLE allows transient solutions of higher Courant numbers
- PIMPLE iterates and under-relaxes the solution of pressure-velocity coupling within a time step
- PIMPLE-loop ends by reaching either a static number of iterations or a residual limit ε

Residual Control
solutionControl classes

- **OpenFOAM Extend project code base lacks the option to prematurely end the PIMPLE-loop using solution residuals**
- **OpenFOAM Foundation version does have residual control**
- *src/finiteVolume/cfdTools/general/solutionControl*
- *solutionControl classes do a bit more as just residual control and are connected to some additional API changes*
- **We distilled the mechanism itself and created a stand-alone light-weight class compatible with OpenFOAM Extend code**

Residual Control (2)
residualControl Class

- **Reads residualControl dictionary from the fvSolution-file**
- *Dictionary format is fully compatible with OpenFOAM Foundation version of solution control*
- **Registers solution residuals**

```
void registerPerf(const word& fieldName,
  const lduMatrix::solverPerformance& perf);
void reset();
```

- **Checks the solution convergence**

```
bool converged() const;
```

Residual Control (3)
pimpleFoam

createFields.H:

```
residualControl residuals(
  mesh, "PIMPLE"
);
```

UEqn.H:

```
residuals.registerPerf(
  U.name(),
  solve(UEqn() ==
     -fvc::grad(p))
);
```

pimpleFoam.C:

```
int oCorr=0;
do {
  …
  #include "UEqn.H"
  for(…) {
    #include "pEqn.H"
  }
  turbulence->correct();
  oCorr++;
} while(oCorr<nOuterCorr
  && !residuals.converged());
```

PIMPLE-FSI Solver
PIMPLE-FSI-loop

- We reworked the FSI solver replacing the flow solver, originally rhoPisoFoam, with rhoPimpleFoam

Iterate until FSI error is low

Solve fluid, PIMPLE-loop, until error is low

Apply fluid forces

Solve solid displacement until error is low

Move fluid mesh using Aitken under-relaxation

Move solid mesh

Time and iteration management

PIMPLE-FSI Solver (2) pimpleFsiElasticNonLinULSolidFoam

```
label oCorr=0, oFSICorr=0;
bool moveFluid=false;
do // PIMPLE-FSI-loop
{
  if (moveFluid) {
    … // fluid mesh movement
    moveFluid=false;
  }

  #include "solveFluid.H"

  if(fluidResiduals.converged()
    || (oCorr==(nOCorr-1))) {
```

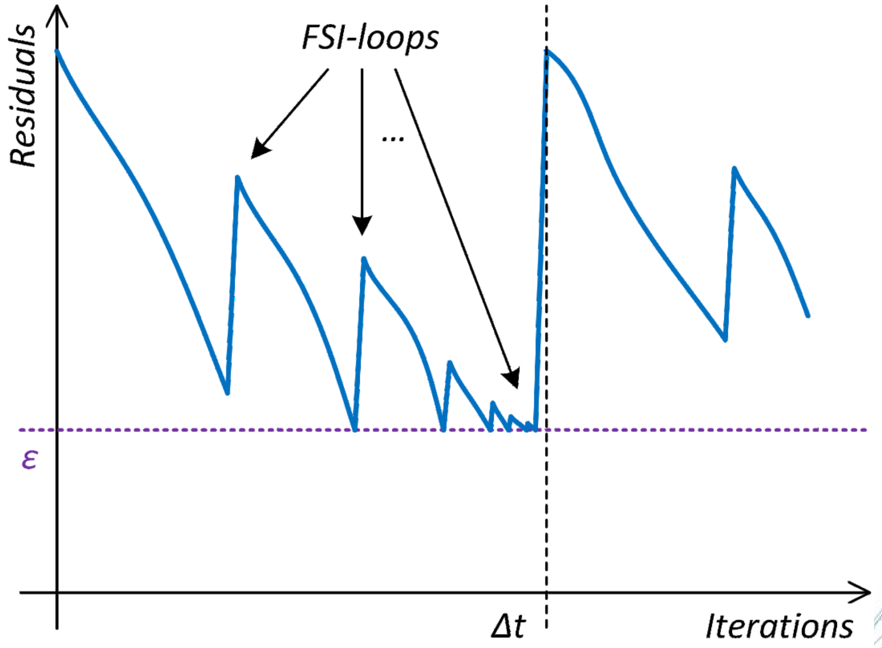
(cont.)

```
    … // apply forces, solve solid
    moveFluid=true;
    oFSICorr++; // FSI iterations

    oCorr=-1; // recalculate fluid
    fluidResiduals.reset();
  }
  oCorr++; // PIMPLE iterations
} while (
  (fsiResidualNorm>oCorrTolerance)
  && (oFSICorr<nOFSICorr));
…
```

PIMPLE-FSI Solver (3)
Behaviour

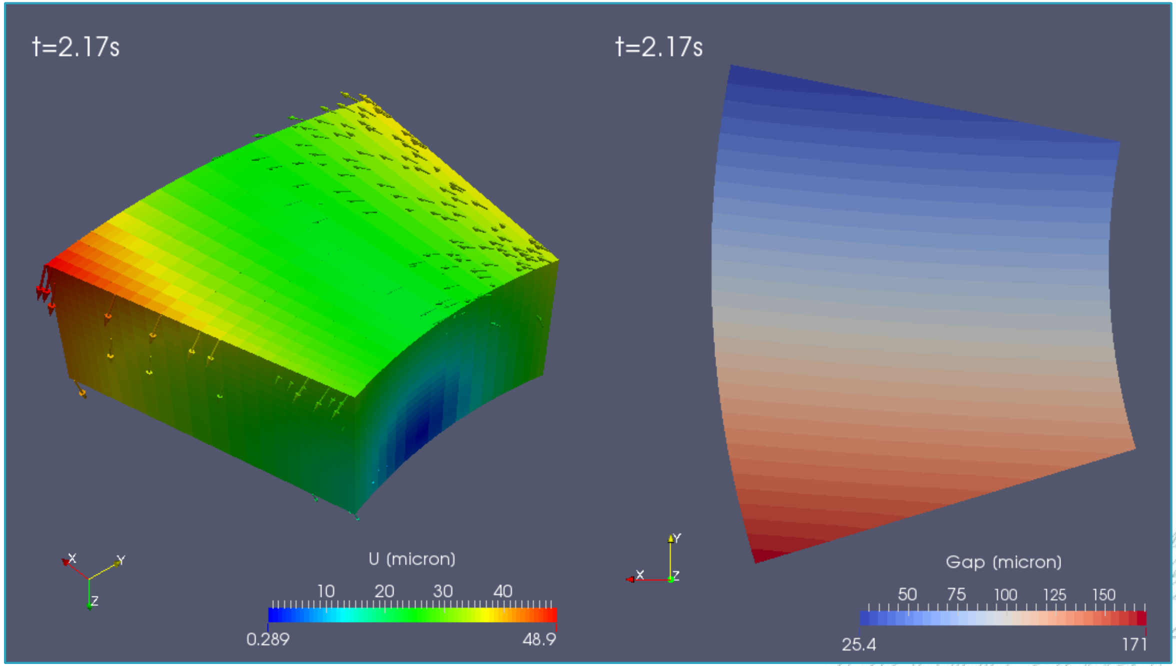
- **PIMPLE-FSI solver shows much higher stability**
- *PIMPLE flow solver better deal with meshes of low and/or deteriorated quality*
- *PIMPLE flow solver naturally stabilizes fluid-solid interaction*



- **PIMPLE-FSI solver with residual control converges quicker**

PIMPLE-FSI Solver (4)
Hydrodynamic Bearing Simulation

- **Stability by higher Courant numbers**
- *Courant of 100+*
- **Transient bearing start-up possible**
- *Unthinkable before due to fluid-structure interaction*



References and Discussion

Heinrichson, N., & Santos, I. (2007). On the Design of Tilting-Pad Thrust Bearings. Ph.D. thesis. Retrieved June 22, 2016, from http://orbit.dtu.dk/.