SIMULATION OF FLUID-STRUCTURAL INTERACTION USING OPENFOAM

Hua-Dong Yao

Department of Applied Mechanics,
Chalmers University of Technology

Sep 15, 2014

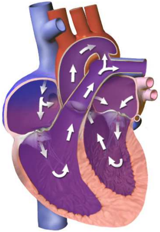
OUTLINE

- Fundamental knowledge on FSI
- FSI of large deformations simulated with OpenFOAM
  - Fluid solver
  - Structure solver
  - Coupling method
- Post-processing
- A tip
- Questions

FLUID-STRUCTURAL INTERACTIONS IN NATURE

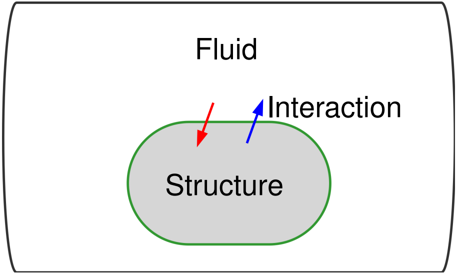
- A falling leave.
- A flapping flag.
- Blood circulation in the human heart.
- ...





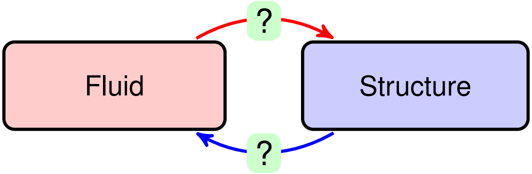
Approaches for FSI simulations

- The partitioned method is to separately solve the governing equations of the flow and structure with two independent solvers.
- The monolithic method is to simultaneously solve the governing equations of the flow and structure with a single solver.



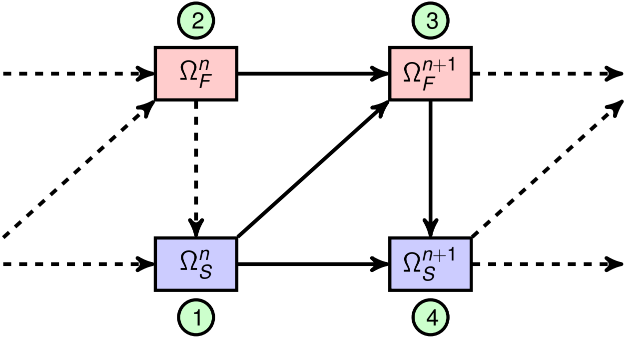
Coupling techniques for the partitioned method

- The FSI solvers of OpenFOAM in present are implemented using the partitioned method.
- There are two techniques coupling the fluid and structure solvers, either explicitly or implicitly.



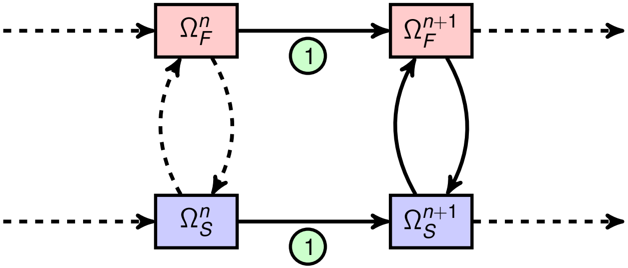
Explicit coupling

- The explicit coupling method applies to the weak interaction.
- OpenFOAM includes a weak-FSI solver.
- However, the weak-FSI solver is not our concern due to its simplicity.



Implicit coupling

- The implicit coupling method is suitable for the strong interaction.
- The focus in this lecture is the FSI of large structural deformations.



How to get the FSI solver of OpenFOAM?

- We will use the version foam-extend-3.1.

- The link for downloading the latest FSI solver is
  `http://openfoamwiki.net/index.php/Extend-bazaar/Toolkits/Fluid-structure_interaction`

- The new version of OpenFOAM that is coming will include this FSI package.

- The solver is installed in the following way:

```
$ OFextend31   // Launch foam-extend-3.1 by typing 'OFextend31'.
$ run                          // Go to the local user directory 'run'.
$ mv downloadDir/Fsi_31.tar.gz . // Move the downloaded file here.
$ tar -xzvf Fsi_31.tar.gz                  // Extract this file here.
        // A folder called FluidStructureInteraction is generated.
$ cd FluidStructureInteraction/
$ cd src/
$ ./Allwmake
```

- The application of our interest is fsiFoam.

Files for setting up a computation

- There are in default three tutorial cases for fsiFoam.

  $ cd FluidStructureInteraction/
  $ cd run/fsiFoam/
  $ ls
  3dTube beamInCrossFlow HronTurekFsi3

- Take 3dTube for example,

  $ cd 3dTube/
  $ ls
  fluid makeLinks makeSerialLinks removeSerialLinks solid

- The folder, fluid, contains the files for the fluid computation.
- The folder, solid, contains the files for the structure computation.
- Running the files *Links will link the two computations on the interfaces between the fluid and solid domains.

Flow simulation

- Fluid is incompressible.
- The fluid solver utilizes the PISO algorithm.
- Dynamic mesh is used due to the displacement of FSI interface.
- The internal grids of the fluid mesh adjust their positions when the FSI interface moves.
- Settings for the dynamic mesh:

  $ vi fluid/constant/dynamicMeshDict

  ```
  dynamicFvMesh dynamicMotionSolverFvMesh;
  solver velocityLaplacian;            // Solve the Laplacian equation.
  diffusivity quadratic inverseDistance (fsiPatchNameFluidSide);
  ```

SETTINGS FOR FLOW SIMULATION

- The FSI interface also needs a boundary condition (BC).
- Its settings are the same as the wall BC, except for the velocity.

  $ vi fluid/0/U

  ```
  {
      type     movingWallVelocity;
      value     uniform (0 0 0);
  }
  ```

- The whole FSI simulation takes the settings of the IO and time control that are specified in the fluid solver.

  $ vi fluid/system/controlDict

Governing equations of Structure solver

- The solid is linear elastic.
- The governing equations are

$$\int_{V_S} \rho_S \frac{\partial \delta v}{\partial t} dV_S - \oint_{S_S} n_u \cdot (2\mu + \lambda) \nabla \delta \vec{u} \, dS_S = \oint_{S_S} n_u \cdot \vec{q} \, dS_S + \int_{V_S} \rho_S \frac{\partial \delta f_b}{\partial t} dV_S$$

$$q = \mu \left(\nabla \delta \vec{u} \right)^T + \lambda tr \left(\delta \vec{u} \right) I - (\mu + \lambda) \nabla \delta \vec{u} + \mu \nabla \delta \vec{u} \cdot \left(\nabla \delta \vec{u} \right)^T + \frac{1}{2} \lambda \left(\nabla \delta \vec{u} : \nabla \delta \vec{u} \right) I + \sum_S \cdot \delta F_S^T + \delta \sum_S \cdot \delta F_S^T$$

Discretization of structural equations

- The space discretization:

$$\rho_{PS} V_{PS} \frac{\partial \delta v_P}{\partial t} - \sum_f (2\mu_f + \lambda_f)\, n_{fS} \cdot \left(\nabla \delta \vec{u}\right)_f S_{fS} = \sum_f n_{fS} \cdot q_f S_{fS} + \sum_f \rho_S (\delta f_b)_P V_{PS}$$

$$n_{fS} \cdot \left(\nabla \delta \vec{u}\right)_f = |\Delta_{fS}| \frac{\delta \vec{u}_N - \delta \vec{u}_P}{|d_{fS}|} + (n_{fS} - \Delta_{fS}) \cdot \left(\nabla \delta \vec{u}\right)_f$$

- The backward time discretization:

$$\delta v_P^n = \frac{3\delta \vec{u}_P^n - 4\delta \vec{u}_P^o + \delta \vec{u}_P^{oo}}{2\delta t} \qquad \left(\frac{\partial \delta v_P}{\partial t}\right)^n = \frac{3\delta v_P^n - 4\delta v_P^o + \delta v_P^{oo}}{2\delta t}$$

Specification of Structural properties

- The files in the directory 'constant'
  $ cd myCase/solid/constant/
  $ ls
  polyMesh rheologyProperties stressProperties

- $ vi rheologyProperties

```
planeStress    no;          // 'yes' for 2D cases, 'no' for 3D cases.
rheology
{
    type    linearElastic;    // The structure is of linear elasticity.
    rho    rho    [1 -3 0 0 0 0 0] 1000;          // Density.
    E    E    [1 -1 -2 0 0 0 0] 5.6e6;          // Young's modulus.
    nu    nu    [0 0 0 0 0 0 0] 0.4;          // Poisson's ratio.
}
```

Specification of Structural solver

- $ cd myCase/solid/constant/
  $ vi stressProperties

```
stressModel    unsTotalLagrangianStress;
unsTotalLagrangianStressCoeffs
{
    nCorrectors    1000;
    convergenceTolerance    1e-7;
    relConvergenceTolerance    1e-3;
    nonLinear    yes;
    debug    no;
    moveMesh    yes;
}
```

Numerical schemes of structure solver – I

- $ cd myCase/solid/system/
  $ vi fvSchemes

```
d2dt2Schemes
{
    default     none;
    d2dt2(D)    backward;
}
ddtSchemes
{
    default     none;
    ddt(D)      backward;
}
```

- D relates to the displacement $\vec{u}$ in the structural equations.

NUMERICAL SCHEMES OF STRUCTURE SOLVER – II

- $ cd myCase/solid/system/
  $ vi fvSchemes

```
laplacianSchemes
{
    default    none;
    laplacian(DD,D)    Gauss linear skewCorrected 1;
                                  // Gauss linear corrected
}
snGradSchemes
{
    default    none;
    snGrad(D)    skewCorrected 1;             // corrected
}
```

Advantage: effectiveness for bad mesh quality.
Disadvantage: high computation cost.

- DD means the displacement increment $\delta\vec{u}$ in the structural equations.

NUMERICAL SCHEMES OF STRUCTURE SOLVER – III

- $ cd myCase/solid/system/
  $ vi fvSchemes

```
interpolationSchemes
{
    default    none;
    interpolate(mu)    linear;              // leastSquares
    interpolate(lambda)    linear;          // leastSquares
}
```

Advantage: effectiveness for bad mesh quality.
Disadvantage: high computation cost.

Matrix algorithms of structure solver – I

- $ cd myCase/solid/system/
  $ vi fvSolution

```
solvers
{
    D
    {
        solver    PCG;
        preconditioner    DIC;
        tolerance    1e-09;
        relTol    0.1;
    }
    DD
    {
        solver    PCG;
        preconditioner    DIC;
        tolerance    1e-09;
        relTol    0.1;
    }
}
```

MATRIX ALGORITHMS OF STRUCTURE SOLVER – II

- $ cd myCase/solid/system/
  $ vi fvSolution

```
relaxationFactors
{
        D     0.5;
        DD    0.5;
}
```

Boundary conditions of structure solver – I

- $ cd myCase/solid/0/
  $ vi D

```
fsiPatchName
{
    type        tractionDisplacement;                    // The BC type
    traction    uniform ( 0 0 0 );     // Externally imposed traction
    pressure    uniform 0;           // Externally imposed pressure
    value       uniform (0 0 0); // Externally imposed displacement
}
```

- Traction is the force per unit area on a surface, including the normal and shear components.

Boundary conditions of structure solver – II

- $ cd myCase/solid/0/
  $ vi D

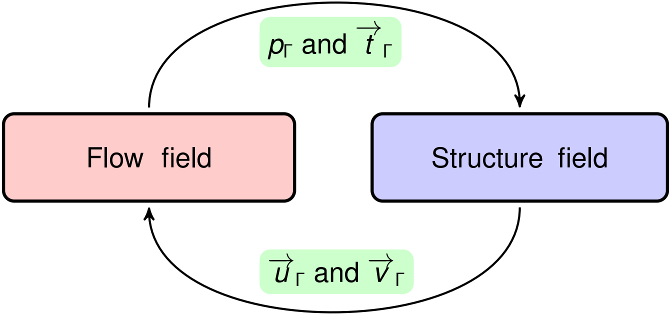
```
fixedPatchName
{
    type     fixedDisplacement;                    // The BC type
    value     uniform (0 0 0); // Externally imposed displacement
}
```

- Structural BC types are given by the following source codes.

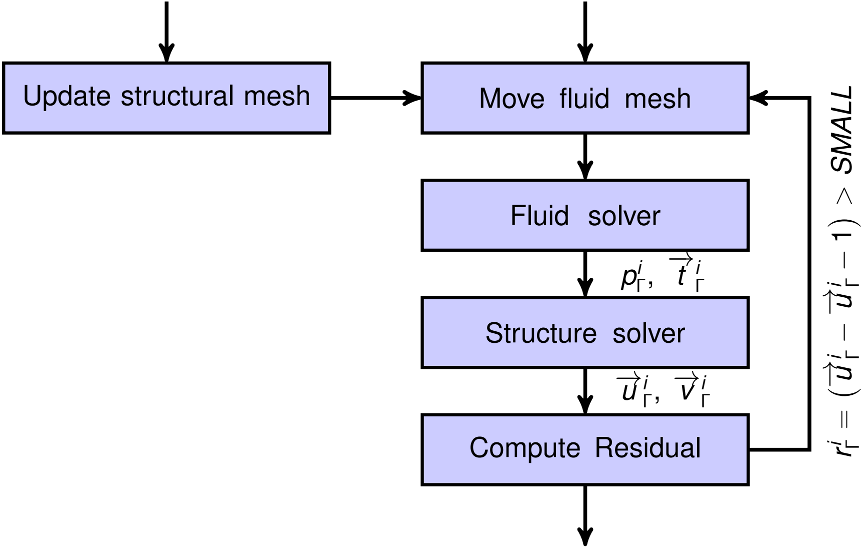
```
$ OFextend31
$ src           // Go to the installation directory of the source codes
$ cd ./solidModels/fvPatchFields/
$ ls
```

Coupling method – interaction

- The exchanged variables on the interfaces are:
  - pressure (p_Γ) and viscous force ($\vec{t}_\Gamma$) in the fluid side,
  - displacement increment ($\vec{u}_\Gamma$) and velocity ($\vec{v}_\Gamma$) in the structure side.



Coupling method – interaction



Aitken relaxation

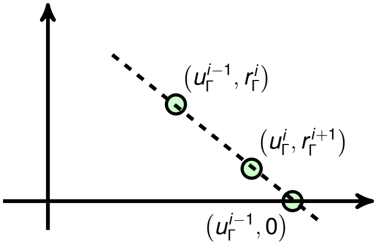
- The Aitken relaxation applies to accelerate the coupling process.

$$u_\Gamma^i = \widetilde{S} \circ \widetilde{F}\left(u_\Gamma^{i-1}\right)$$

$$r_\Gamma^i = \widetilde{u}_\Gamma^i - \widetilde{u}_\Gamma^{i-1}$$

$$r_\Gamma^{i+1} = \widetilde{u}_\Gamma^{i+1} - \widetilde{u}_\Gamma^i$$

$$0 = r_\Gamma^{i+1} + \frac{r_\Gamma^{i+1} - r_\Gamma^i}{u_\Gamma^i - u_\Gamma^{i-1}}\left(u_\Gamma^{i+1} - u_\Gamma^i\right)$$



$$u_\Gamma^{i+1} = u_\Gamma^i - r_\Gamma^{i+1}\frac{\widetilde{u}_\Gamma^i - \widetilde{u}_\Gamma^{i-1}}{r_\Gamma^{i+1} - r_\Gamma^i} = u_\Gamma^i + \underbrace{\frac{u_\Gamma^i - u_\Gamma^{i-1}}{u_\Gamma^{i-1} - u_\Gamma^i + \widetilde{u}_\Gamma^{i+1} - \widetilde{u}_\Gamma^i}}_{\omega_{i+1}}\underbrace{\left(\widetilde{u}_\Gamma^{i+1} - u_\Gamma^i\right)}_{r_\Gamma^{i+1}}$$

$$u_\Gamma^i := u_\Gamma^{i-1} - \omega_i r_\Gamma^i \qquad \omega_{i+1} = -\omega_i \frac{\left(r_\Gamma^i, r_\Gamma^{i+1} - r_\Gamma^i\right)}{\left\|r_\Gamma^{i+1} - r_\Gamma^i\right\|^2}$$

Settings for coupling the solvers

- The setting file is located in the fluid solver.

  $ vi fluid/constant/fsiProperties

```
solidPatch plate;
solidZone plateZone;
fluidPatch plate;
fluidZone plateZone;
relaxationFactor 0.05;
interfaceDeformationLimit 0;
outerCorrTolerance 1e-6;
nOuterCorr 30;
interpolatorUpdateFrequency 0;
couplingScheme Aitken;
couplingReuse 0;
coupled no;
```

CREATION OF COUPLED ZONES

- The coupling process requires the information of FSI zones.
- The fluid and structure solvers use the same way to create these zones.
- Take the fluid solver for example,

  ```
  $ cd myFsiCase/fluid/
  $ setSet -batch setBatch
  $ setsToZones -noFlipMap
  ```
- A convenient way is to write the above commands in a batch file.
- The file setBatch saves the following contents:

  ```
  faceSet fsiPatchZoneName new patchToFace fsiPatchName
  quit
  ```

Link the solvers for coupling

- A single-processor computation links the local main directories.

```
$ cd myCase/fluid/constant/
$ ln -s ../../solid/constant solid

$ cd ../system
$ ln -s ../../solid/system solid

$ cd ../0
$ ln -s ../../solid/0 solid
```

- A parallel computation needs to link the above directories for every processor, in addition to the local main directories.

```
foreach proc(processor*)
cd $proc
cd 0
ln -s ../../../solid/$proc/0 solid
cd ../constant
ln -s ../../../solid/$proc/constant solid
cd ../..
```

Link the solvers

- We can write the above commands into a batch file for convenience.

- The FSI package, which we downloaded before, contains three tutorial cases in default.

```
$ cp FluidStructureInteraction/run/fsiFoam myFsiTut -r
$ cd myFsiTut
$ ls
    3dTube    beamInCrossFlow    HronTurekFsi3
```

How to run a computation?

- Take for instant the tutorial case of a three-dimensional tube,

```
$ cd myFsiTut/3dTube/
$ ./makeSerialLinks fluid solid
$ ./Allrun
```

POST-PROCESSING

- The post-processing tool can be paraFoam or other commercial softwares.
- The application paraFoam is able to automatically load the results of both the domains.

  $ paraFoam
- In regard of Tecplot, the command is

  $ foamToTecplot360

A TIP

- A patch imposed with the symmetric BC must be precisely flat.
- Otherwise, the nodes on this patch could deviate from their original positions during the computation.
- The reason is that computation of the node motions adopts the normal directions of the local cell faces.

AN EXERCISE

- Let us do an exercise with the tutorial case of a flapping plate.
- Please change the BC type of the fixed walls from fixedDisplacement to timeVaryingFixedDisplacement.

QUESTIONS

- What is the implicit coupling method?
- How to set the IO and time controls for a FSI computation?
- How to specify the BC types for the variables on the interface between the fluid and structure domains?
- Which method is utilized to accelerate the convergence speed of the coupling progress?
- Which variables are transferred on the interface?
- Which two preliminary works should we do to establish the coupling?

Thanks

CHALMERS