Implement solid particles in a VOF solver

Implement solid particles in a VOF solver

Contents

- You will add an additional class to an existing solver, and that way extend the solver functionality.

Prerequisites

- You are familiar with the directory structure of OpenFOAM applications.
- You are familiar with user compilation procedures of applications.
- You are familiar with the fundamental high-level components of application codes, and how new classes can be introduced to an application.

Learning outcomes

- You will practice high-level coding and modification of solvers.
- You will adapt case set-ups according to the new solver.
- You will improve your understanding of classes and object orientation, from a high-level perspective.

Note that you will be asked to pack up your final cleaned-up directories and submit them for assessment of completion.

Copy the interFoam solver, clean up, re-name and compile

Make sure that you understand all the steps in the following procedure. Particular attention is paid to cleaning up. What is being cleaned up?

```
cd $WM_PROJECT_DIR
cp -r --parents applications/solvers/multiphase/interFoam $WM_PROJECT_USER_DIR
cd $WM_PROJECT_USER_DIR/applications/solvers/multiphase
mv interFoam solidParticleInterFoam
cd solidParticleInterFoam
rm -r interMixingFoam overInterDyMFoam
wclean
rm -rf Make/linux*
mv interFoam.C solidParticleInterFoam.C
sed -i.orig s/interFoam/solidParticleInterFoam/g Make/files
sed -i s/FOAM_APPBIN/FOAM_USER_APPBIN/g Make/files
sed -i s/"..\/VoF"/"\$(FOAM_SOLVERS)\/multiphase\/VoF"/g Make/options
wmake
```

Spend some extra time understanding the last `sed`-line!!! Why is it needed?

At this point you should check that the code still works for the damBreak tutorial.

Add particles to the solver

Now we will add functionality from the `solidParticleCloud` class.
Modify `solidParticleInterFoam.C`:

Include the class declarations of `solidParticleCloud.H`.
At the header, add:

```
#include "solidParticleCloud.H"
```

Create a solidParticleCloud object.
After `#include "setInitialDeltaT.H"`, add after the if-statement:

```
solidParticleCloud particles(mesh);
```

Move the particles.
Before `runTime.write();`, add:

```
particles.move(g);
```

Make sure that you understand why these lines are added, and why at these locations!

Add particles to the solver

We need to add some libraries when we compile.
Make sure that `Make/options` has the following lines (where '. . .' is the original lines):

```
EXE_INC = \
    ... \
    -I$(LIB_SRC)/lagrangian/basic/lnInclude \
    -I$(LIB_SRC)/lagrangian/solidParticle/lnInclude

EXE_LIBS = \
    ... \
    -llagrangian \
    -lsolidParticle
```

Compile:

```
wmake
```

Why do we have to add these? Make sure that you understand all lines!

Add particles to the interFoam/damBreak case

Base the case on the original `damBreak` case:

```
run
cp -r $FOAM_TUTORIALS/multiphase/interFoam/RAS/damBreak/damBreak solidParticleDamBreak
cd solidParticleDamBreak
```

The modifations that need to be made are not obvious. There are no examples using the solidParticle class in the original code. I looked at examples for similar classes, as well as the source code. Some time between v1706 and v1806 there was a significant change in the required files, and here we set it up for v1906.
Create a directory for info of the particles:

```
mkdir -p 0/lagrangian/defaultCloud
```

We will add files for diameter (`d`), positions (`positions`), velocity (`U`), original ID (`origId`) and original processor ID (`origProcId`).
Create a directory for the particle cloud properties, which we will only use to specify the format of positions of the particles (the default in v1906 seems to be `coordinates`, be but we will use the old `positions`):

```
mkdir -p 0/uniform/lagrangian/defaultCloud
```

We will set the particle properties in `constant/particleProperties`.

Add particles to the interFoam/damBreak case

We will set up a case with 2 particles. That number is determined by the solver by the number of entries in the lists in the following files.
Diameter file (`0/lagrangian/defaultCloud/d`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       scalarField;
    location    "0/lagrangian/defaultCloud";
    object      d;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

2{2.0e-3}

// ************************************************************************* //
```

The number of particles is 2, and since they have the same diameter their diameters are in a list given by the curly brackets.

Add particles to the interFoam/damBreak case

Positions file (`0/lagrangian/defaultCloud/positions`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       Cloud<solidParticle>;
    location    "0/lagrangian/defaultCloud";
    object      positions;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

2
(
(2e-2 0.56 0.005) -1
(3e-2 0.56 0.005) -1
)

// ************************************************************************* //
```

The particles start at different positions, so the list is given with regular brackets. The `-1` at the end of the line means that the class will have to find the cells that they are located in. In the time directories it will be stated which cell they are in.

Add particles to the interFoam/damBreak case

Velocity file (`0/lagrangian/defaultCloud/U`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       vectorField;
    location    "0/lagrangian/defaultCloud";
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

2
(
(1.7e-1 0 0)
(1.7 0 0)
)

// ************************************************************************* //
```

The particles start with different velocities.

Add particles to the interFoam/damBreak case

Original ID file (`0/lagrangian/defaultCloud/origId`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       labelField;
    location    "0/lagrangian/defaultCloud";
    object      origId;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

2
(
0
1
)

// ************************************************************************* //
```

The particles are numbered with labels.

Add particles to the interFoam/damBreak case

Original processor ID file (`0/lagrangian/defaultCloud/origProcId`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       labelField;
    location    "0/lagrangian/defaultCloud";
    object      origProcId;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

2{0}

// ************************************************************************* //
```

Both particles originate in the processor0 region (I only tested this running sequentially, so I have not checked how this is affected when running in parallel).

Add particles to the interFoam/damBreak case

cloudProperties file (`0/uniform/lagrangian/defaultCloud/cloudProperties`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "0/uniform/lagrangian/defaultCloud";
    object      cloudProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

geometry        positions;

// ************************************************************************* //
```

I interpret the purpose of this file (for this class) to define if we use a `positions` file or a `coordinates` file for the particle positions. It may be the case that the case set-up would be simpler with the `coordinates` option, but I have not checked that up. The class writes out `coordinates` files in addition to the `positions` files in the time directories.

Add particles to the interFoam/damBreak case

Particle properties file (`constant/particleProperties`):

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1806                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      particleProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

rhop rhop [ 1 -3  0  0  0  0  0] 1000;
e    e    [ 0  0  0  0  0  0  0] 0.8;
mu   mu   [ 0  0  0  0  0  0  0] 0.2;

// ************************************************************************* //
```

These are the particle density, and the coefficients of normal and tangential restitution as the particles hit the walls. The class does not include particle-particle collisions. You would have to use the `kinematicParticle` class or some more advanced class.

Run and animate using foamToVTK and ParaView

```
blockMesh
setFields
solidParticleInterFoam 2>&1 | tee log_solidParticleInterFoam
foamToVTK
paraview
```

- File/open: VTK/solidParticeDamBreak_..vtk
- File/open: VTK/lagrangian/defaultCloud/defaultCloud_..vtk
- For the solidParticleDamBreak object: Display: Opacity 0,3. Color By: alpha.water (cell values)
- For the defaultCloud object: Create box glyphs (Scale Mode off, Scale Factor 0.005) to visualize the particles.
- Run the animation and enjoy...

Note that with the OpenFOAM reader for Paraview distributed with v1906 it is possible to skip the VTK step and do the same procedure with the Paraview Part lagrangian/defaultCloud.