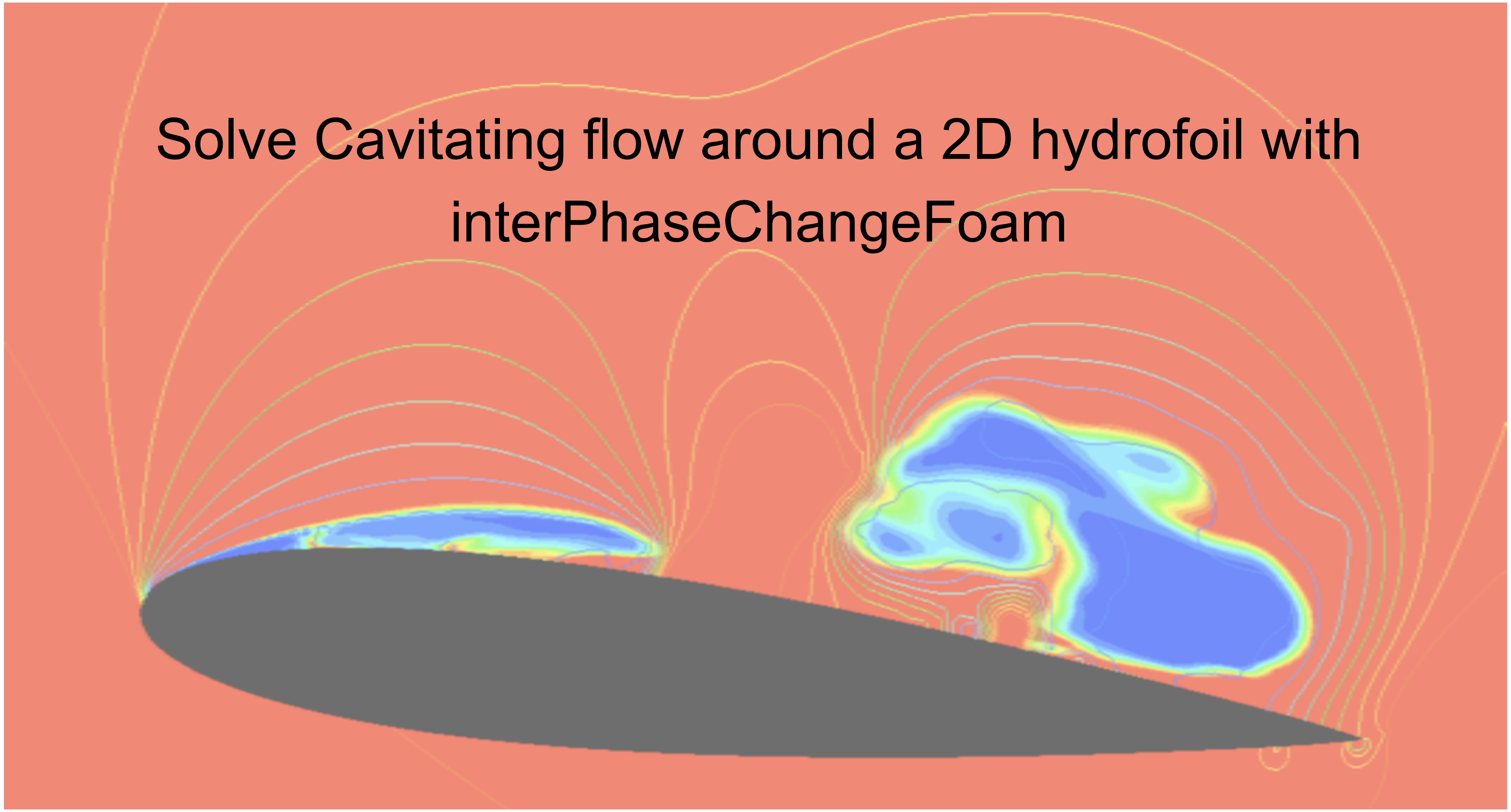
Solve Cavitating flow around a 2D hydrofoil with interPhaseChangeFoam



NaiXian LU (naixian.lu@chalmers.se)
Shipping and Marine Technololy, LES/Cavitation

interPhaseChangeFoam

- cavitation
- two phase flow
- flow modelled using LES
- interface captured by VOF method
- solved equations:

$$\begin{cases} \nabla \cdot \bar{v} = S_P \\ \partial_t \left(\rho \bar{v} \right) + \nabla \cdot \left(\rho \bar{v} \otimes \bar{v} \right) = -\nabla \bar{p} + \nabla \cdot \left(\bar{S} - B \right) \end{cases}$$

where $S_p = \left(\rho_l^{-1} - \rho_v^{-1} \right) \dot{m}$, $\dot{m} = \dot{m}^+ + \dot{m}^-$

⇓

modelled by mass transfer models

(Kunz, SchnerrSauer, Merkle)

Small modifications to the code

- To improve the near wall behavior
  wallViscosity.H: modify the wall viscosity according to Spalding law

- Kunz mass transfer model: $\dot{m}^{+} = A^{+}\rho_{v}/\rho_{l} \cdot \gamma \min\left[0, \overline{p} - p_{v}\right]$

  $\dot{m}^{-} = A^{-}\rho_{v} \cdot \gamma^{2}\left[1-\gamma\right]$

  Implementation: */phaseChangeTwoPhaseMixtures/Kunz/Kunz.C* because of using a negative pSat

<u>mDotAlphal()</u>

```
return Pair<tmp<volScalarField> >
 (
  mcCoeff_*sqr(limitedAlpha1)
*max(p - pSat(),p0_)/max(p - pSat(), 0.001*mag(pSat())),
//*max(p - pSat(), p0_)/max(p - pSat(), 0.01*pSat()),
mvCoeff_*min(p - pSat(), p0_)
 );
```

mDotAlphal()_c * (1-alphal)=m-

mDotAlphal()_v*alphal=m+

p>pSat: condensation

p<pSat: vaporization

<u>mDotP()</u>

```
return Pair<tmp<volScalarField> >
(
 mcCoeff_*sqr(limitedAlpha1)*(1.0 - limitedAlpha1)
 *pos(p - pSat())/max(p - pSat(),0.001*mag(pSat())),
//*pos(p - pSat())/max(p - pSat(), 0.01*pSat()),
(-mvCoeff_)*limitedAlpha1*neg(p - pSat())
  );
```

mDotP()_c=m-

mDotP()_v*(p-pSat)=m+

Compile the code

- Use the pre-installed OF-1.5.x

```
. /chalmers/sw/unsup/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc
```

- Copy the source code to your working directory

```
cp ooodlesInterPhaseChange.tar $WM_PROJECT_USER_DIR/application/solvers
tar xvf ooodlesInterPhaseChange.tar
```

- Modify the *Make/files* to write the executable in$FOAM_USER_APPBIN

```
EXE = $(FOAM_USER_APPBIN)/ooodlesInterPhaseChange
```

- Compile the code

```
wclean
rm -r Make/linux*
wmake
```

A test case

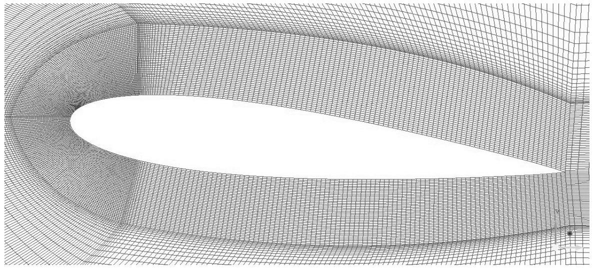
- Copy the test case to your working directory

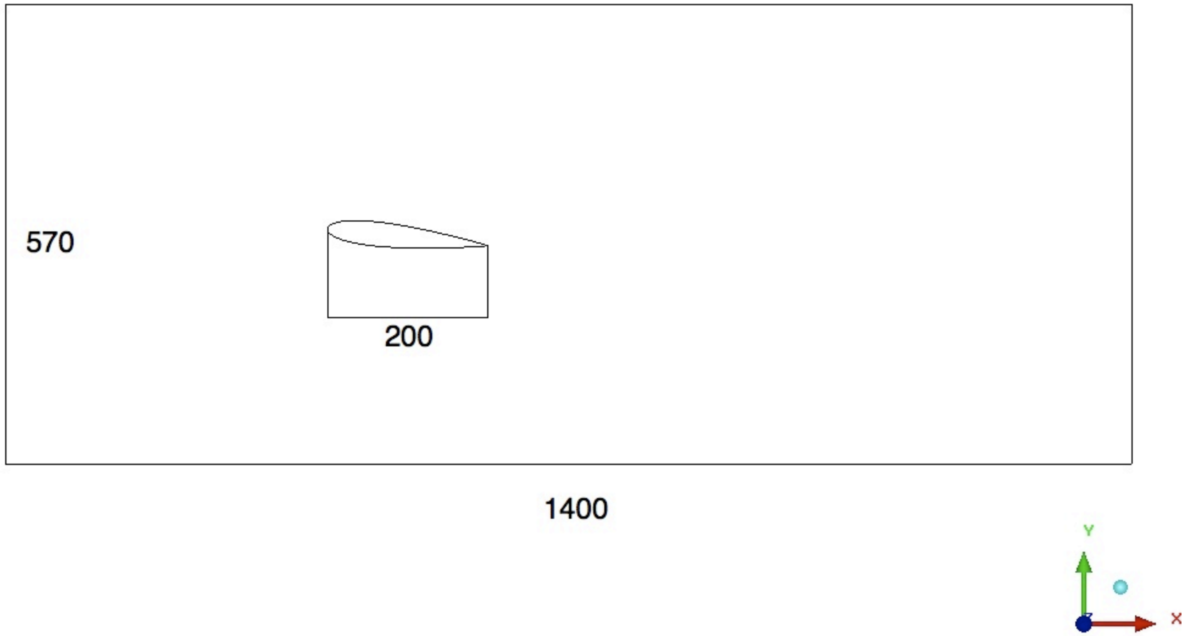
```
cp naca15_test_case.tar $WM_PROJECT_USER_DIR/run
tar xvf naca15_test_case.tar
```

Computational configurations

- geometry: 2D NACA0015
- domain: 1400mm × 570mm
- angle of attack: 6°
- Reynolds number: 1.2e+06
- cavitation number:

$$\sigma = \frac{p_\infty - p_v}{\frac{1}{2}\rho v^2} = 1$$





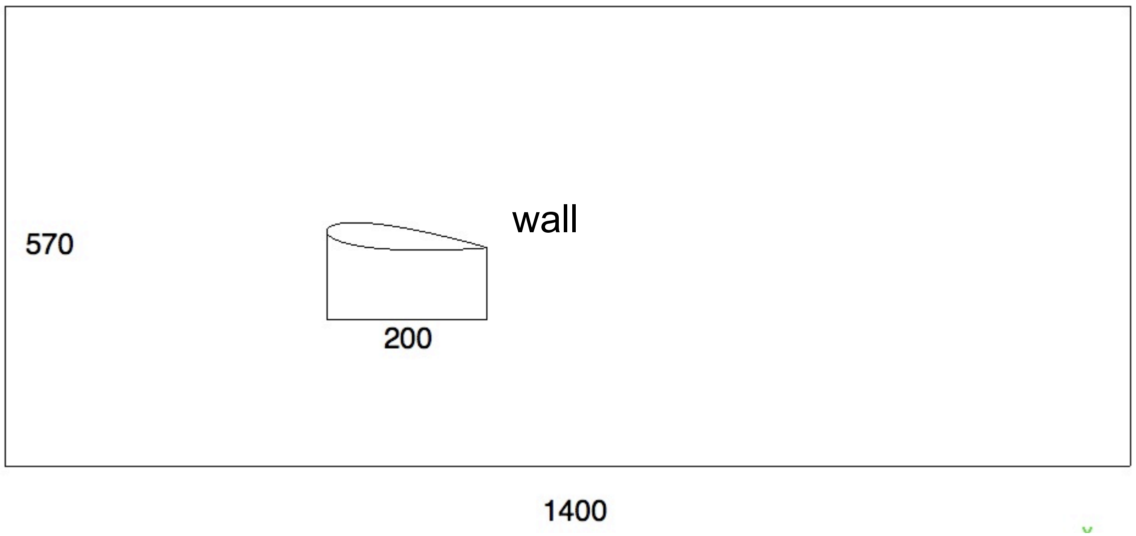
Number of cells: 0.5 millions

Computational configurations

symmetryPlane

Velocity inlet

fixedValue (6 0 0)



Pressure outlet

fixedValue 0

symmetryPlane

- constant/polyMesh/boundary
- 0/U, pd, gamma

LESProperties

Choose the subgrid model in *constant/LESProperties*

```
LESModel        laminar;
delta           smooth;
printCoeffs     on;

laminarCoeffs
{
}
oneEqEddyCoeffs
{
    ck          0.07;
    ce          1.05;
}
dynOneEqEddyCoeffs
{
    ce          1.05;
    filter      simple;
}
.
.
.
```

Implicit LES:

considering the action of the subgrid scale is equivalent to a strictly dissipative action, and letting the leading order truncation error in the discretization of the fluxes emulate the energy dissipation

Fluid properties and mass transfer model

constant/transportProperties

```
phaseChangeTwoPhaseMixture Kunz;

KunzCoeffs
{
   Cc         Cc        [0 0 0 0 0 0 0] 1000;
   Cv         Cv        [0 0 0 0 0 0 0] 10000;
   UInf       UInf      [0 1 -1 0 0 0 0] 6;
   tInf       tInf      [0 0 1 0 0 0 0] 1;

cavitation
{
   pSat       pSat      [1 -1 -2 0 0 0 0] -18000;
   restart    no;
   rampN      200;
   startN     10000;
}
}
twoPhase
{
   transportModel  twoPhase;
   phase1       phase1;
   phase2       phase2;
}

phase2
{
   transportModel  Newtonian;
   nu          nu [0 2 -1 0 0 0 0] 0.0000148;
   rho         rho [1 -3 0 0 0 0 0] 0.023;
   CrossPowerLawCoeffs
   {
      nu0        nu0 [0 2 -1 0 0 0 0] 1e-06;
      nuInf      nuInf [0 2 -1 0 0 0 0] 1e-06;
      m          m [0 0 1 0 0 0 0] 1;
      n          n [0 0 0 0 0 0 0] 0;
   }
   BirdCarreauCoeffs
   {
      nu0        nu0 [0 2 -1 0 0 0 0] 0.0142515;
      nuInf      nuInf [0 2 -1 0 0 0 0] 1e-06;
      k          k [0 0 1 0 0 0 0] 99.6;
      n          n [0 0 0 0 0 0 0] 0.1003;
   }
}

phase1
{
   transportModel  Newtonian;
   nu          nu [0 2 -1 0 0 0 0] 1e-6;
   rho         rho [1 -3 0 0 0 0 0] 998;
   CrossPowerLawCoeffs
   {
      nu0        nu0 [0 2 -1 0 0 0 0] 1e-06;
      nuInf      nuInf [0 2 -1 0 0 0 0]
1e-06;
      m          m [0 0 1 0 0 0 0] 1;
      n          n [0 0 0 0 0 0 0] 0;
   }
   BirdCarreauCoeffs
   {
      nu0        nu0 [0 2 -1 0 0 0 0]
0.0142515;
      nuInf      nuInf [0 2 -1 0 0 0 0]
1e-06;
      k          k [0 0 1 0 0 0 0] 99.6;
      n          n [0 0 0 0 0 0 0] 0.1003;
   }
}
```

environmentalProperties

- *contant/environmentalProperties*

  specifies the gravity acceleration vector, (in this case it is neglected)

```
g               g [0 1 -2 0 0 0 0] (0 0 0);
```

Time step control etc.

```
applicationClass interFoam;
startFrom       startTime;
startTime       0;
stopAt          endTime;
endTime         0.3;
deltaT          2e-05;
writeControl    timeStep;
writeInterval   100;
cycleWrite      0;
writeFormat     ascii;
writePrecision  6;
writeCompression uncompressed;
timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
adjustTimeStep  off;
maxCo           0.2;
maxDeltaT       1;
```

Courant number has a significant impact on the reliability and stability of the unstable flow simulation.

Recommended by OpenFOAM, the upper limit of the Co should be around 0.2

Solution algorithm: *system/fvSolution*

Discretization schemes: *system/fvSchemes*

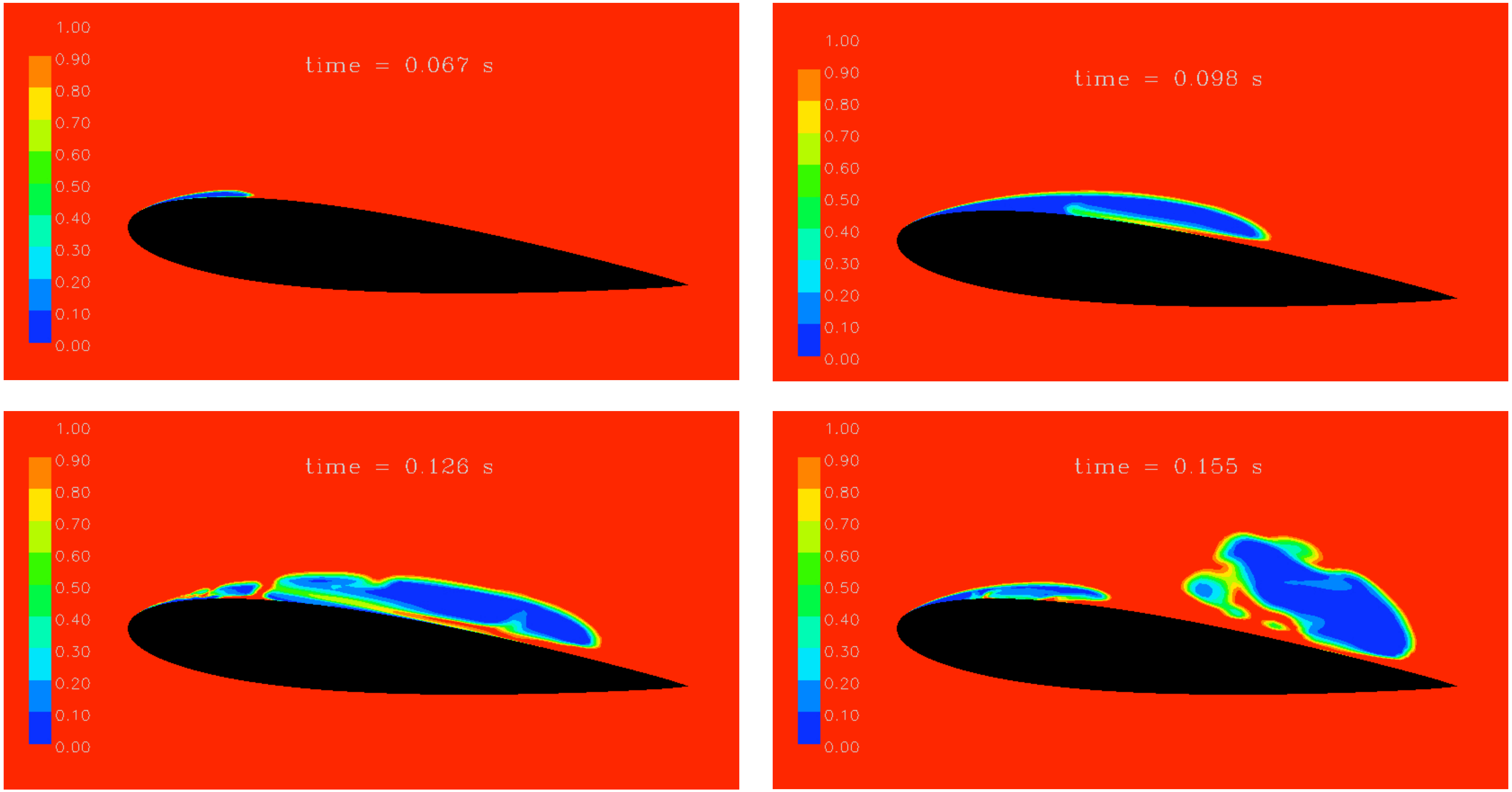
Run the case

```
cd naca15_test_case
ooodlesInterPhaseChange &> log &
tail –f log
```

Note:

simulation of cavitating flow should be started from converged wetted flow result since stabilized pressure distribution is critical for cavitating flow computation.

Result


1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
time = 0.067 s
time = 0.098 s
time = 0.126 s
time = 0.155 s