Implementation of Transport Model into CavitatingFoam to simulate the Cavitation in Diesel Injector Nozzle

Barış BİÇER

Energy and Fluid Science Lab.
Graduate School of Maritime Sciences, Kobe University, JAPAN
bicerbaris@hotmail.com

MSc/PhD course in CFD with OpenSource software, Dec 1-2, 2014, Chalmers

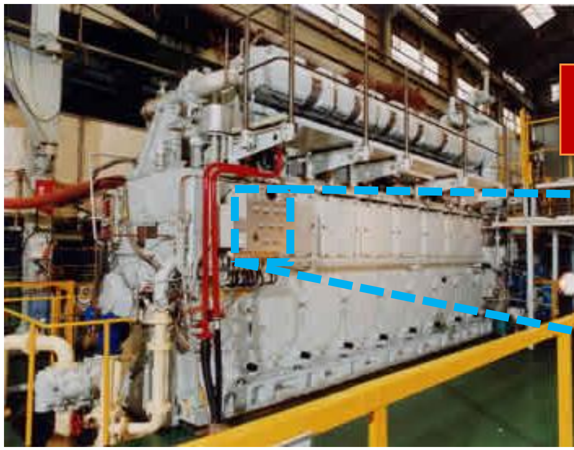
OUTLINE

1. ✱ Background
2. ✱ Cavitation models
3. ✱ Purpose
4. ✱ Description of solvers
5. ✱ Implementation Procedure
6. ✱ Test-case / Results

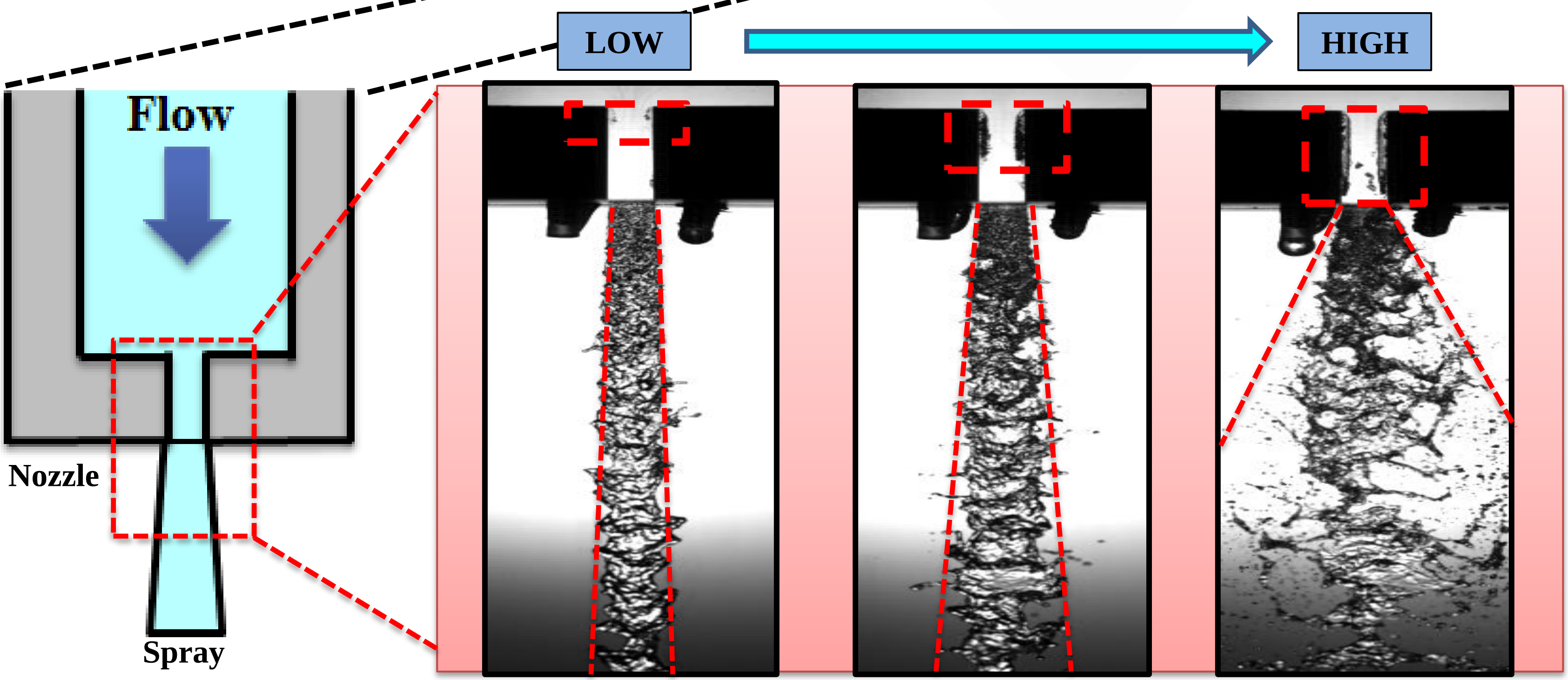
SECTION 1

Background

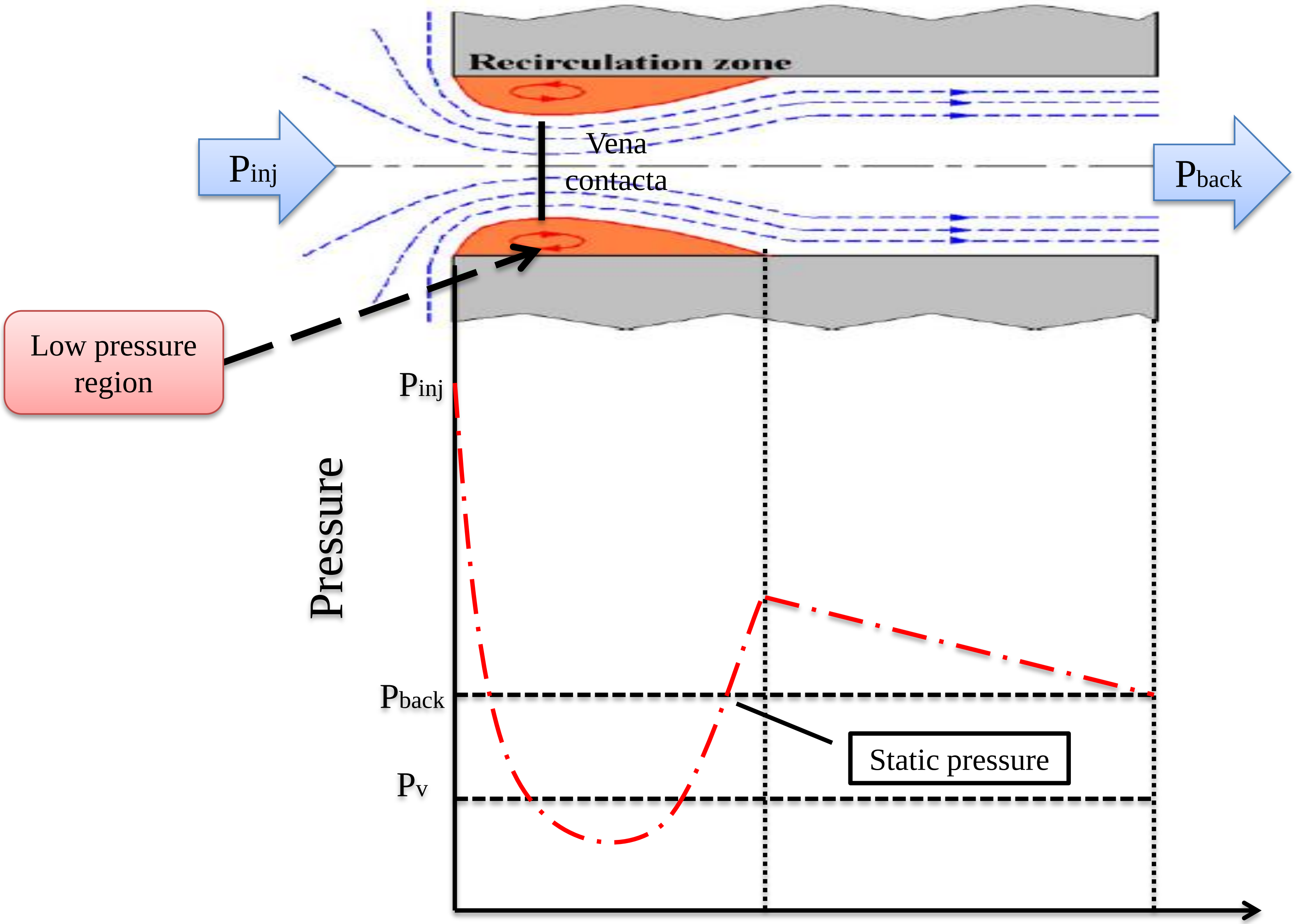
1. BACKGROUND



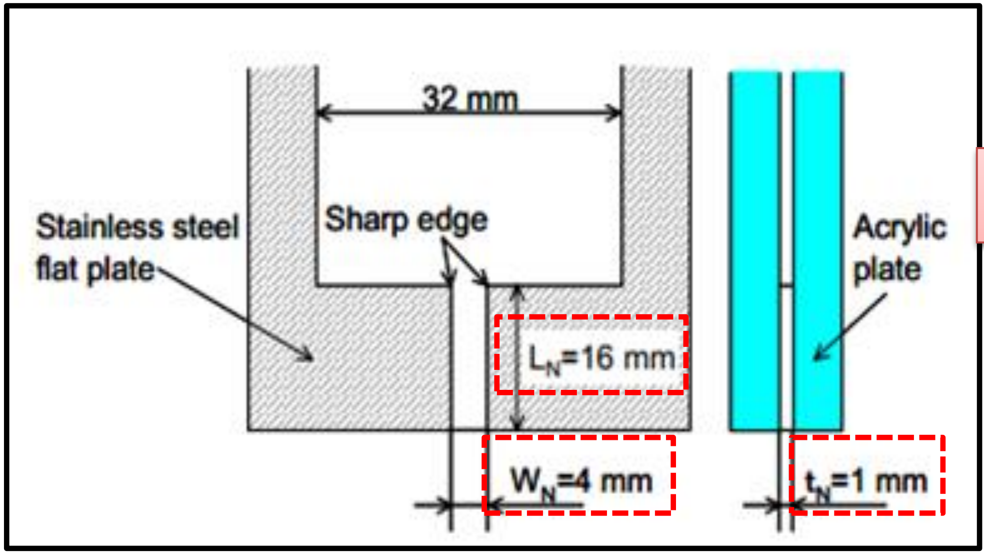
- ❑ low fuel consumption
- ❑ clean emission



Cavitation phenomena in a contraction



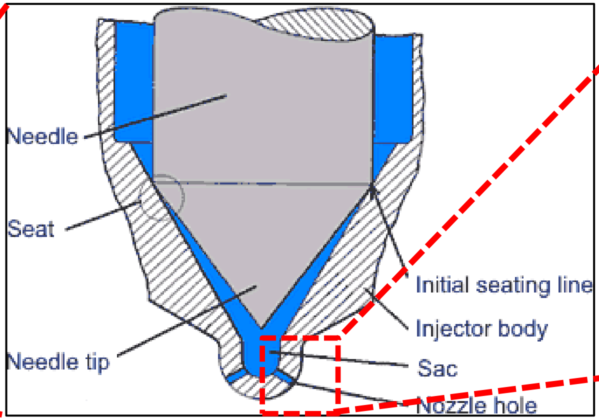
Schematic illustration of cavitation formation inside nozzle hole



shows

- ❑ Rectangular
- ❑ Large-scale transparent nozzle

Real nozzle



- ❑ Length is about 1 mm
- ❑ Diameter is about 0.1~0.2 mm
- ❑ Operate at very high injection pressure

Numerical analyses are necessary

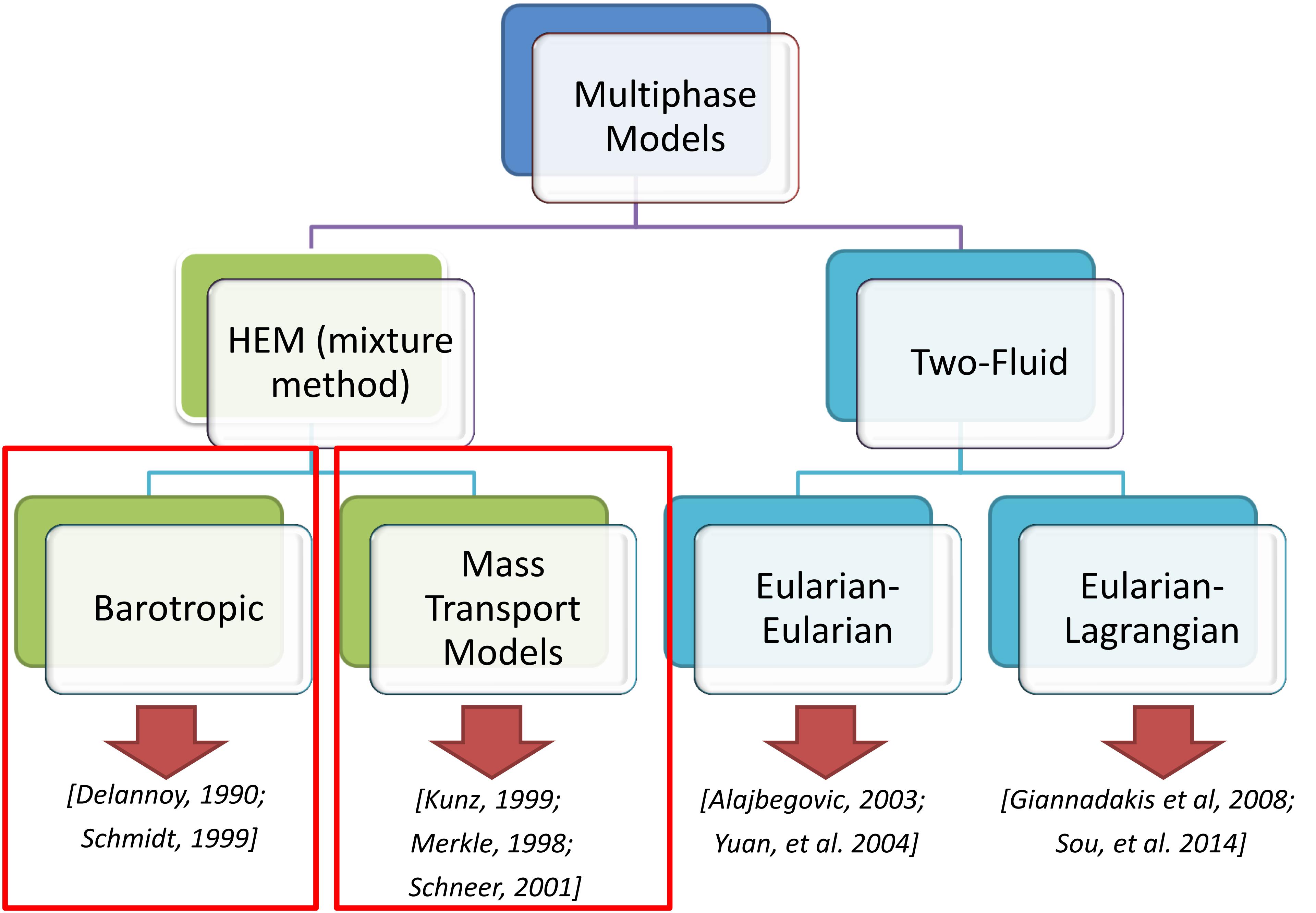
SECTION
2

Cavitation Models

2. CAVITATION MODELS

In order to model the cavitation flow we need first to specify;

- the two-phase treatment of the liquid and vapor and,
- as well as phase transition between them as source term.



1. Sou, A., Biçer, B., & Tomiyama, A. (2014). Numerical simulation of incipient cavitation flow in a nozzle of fuel injector. Computers & Fluids, 103, 42-48.
2. BICER, B., TANAKA, A., FUKUDA, T., & SOU, A. NUMERICAL SIMULATION OF CAVITATION PHENOMENA IN DIESEL INJECTOR NOZZLES, ILASS-ASIA, 2013.

Mass Transport based Equation Model

Three key points cardinally should be chosen regarding to MTM:

1. Selection of an appropriate mass transfer model (means source term of transport equation)
2. A solution strategy for the advection equation

VOF (Volume of fluid)

Available multiphase solvers in OpenFoam

- ❑cavitatingFoam
- ❑compressibleInterFoam
- ❑compressibleTwoPhaseEulerFoam
- ❑interFoam
- ❑interMixingFoam
- ❑LTSInterFoam

- ❑multiphaseEulerFoam
- ❑multiphaseInterFoam
- ❑interPhaseChangeFoam
- ❑twoPhaseEulerFoam
- ❑twoLiquidMixingFoam
- ❑compressibleTwoPhaseEulerFoam

* HEM model
* Barotropic equation with compressibility

* VOF model
* Transport equation with phase change
* incompressible

SECTION 3

Purpose

3. PURPOSE

- Describe the two cavitation solvers included in OpenFOAM-2.3.x such as cavitatingFoam and interPhaseChangeFoam,
- Briefly explain the implementation of the transport equation model into cavitatingFoam, which is called as ”TransportCavitatingFoam”, to simulate the cavitation phenomena inside injector nozzle,
- Test the performance and applicability of the new solver by simulating the turbulent cavitating flow inside the enlarge rectangular nozzle,
- Verify the calculated results through the experimental data.

SECTION 4

Description of solvers

"cavitatingFoam" solver

"Transient cavitation code based on the homogeneous equilibrium model from which the compressibility of the liquid/vapour "mixture" is obtained.

- HEM model with the barotropic closure: $$\frac{D\rho}{Dt} = \Psi \frac{DP}{Dt}$$

ρ: mixture density
P: pressure
t: time
Ψ: compressibility

- Continuity equation: $$\frac{\partial \rho}{\partial t} + \nabla \bullet (\rho U) = 0$$

- Momentum equation: $$\frac{\partial \rho U}{\partial t} + \nabla \bullet (\rho U U) = -\nabla P + \nabla \bullet \left[\left(\mu_{eff} \left(\nabla U + (\nabla U)^T \right) \right] \right.$$

μ_{eff}: effective viscosity

- Vapor mass faction (γ) : $$\gamma = \frac{\rho - \rho_{l,sat}}{\rho_{v,sat} - \rho_{l,sat}}$$

$\rho_{v,sat}$: vapor density at saturation
$\rho_{l,sat}$: liquid density at saturation

- An iterative PIMPLE algorithm is employed to solve P:

$$\frac{\partial (\Psi P)}{\partial t} - \left(\rho_l^0 + (\Psi_l - \Psi_v) P_{sat}\right) \frac{\partial \Psi}{\partial t} - P_{sat} \frac{\partial \Psi}{\partial t} + \nabla \bullet (\rho U) = 0$$

P_{sat} :pressure at saturation
ρ_l^0 : the liquid density at given temperature

cavitatingFoam members

“$FOAM_SOLVERS/multiphase/cavitatingFoam

- alphavPsi.H
- cavitatingFoam.C
- continuityErrs.H
- setDeltaT.H
- setInitialDeltaT.H
- readControls.H
- readThermodynamicsProperties.H
- createFields.H
- pEqn.H
- rhoEqn.H
- UEqn.H
- Make
  - files
  - options

Solves vapor mass fraction eqn.

Main source code shows the flow chart of solver

Described according to compressibility model

“$FOAM_TUTORIALS/multiphase/cavitatingFoam/constant/thermodynamicProperties

```
{
version     2.0;
format      ascii;
class       dictionary;
location    "constant";
object      thermodynamicProperties;
}
// * * * ** * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
barotropicCompressibilityModel linear;
psiv            psiv [ 0 -2 2 0 0 ] 5.6-06;
rholSat         rholSat [ 1 -3 0 0 0 ] 1000;
psil            psil [ 0 -2 2 0 0 ] 4.54e-07;
pSat            pSat [ 1 -1 -2 0 0 ] 2300;
rhoMin          rhoMin [ 1 -3 0 0 0 ] 0.001;
// *****                                                              //
```

rhoMin represents min density which is used to keep the density positive and can be set as 0.001.

cavitatingFoam.C file and solver flowchart

```
Info<< "\nStarting time loop\n" << endl;

while (runTime.run())
{
    #include "readControls.H"
    #include "CourantNo.H"
    #include "setDeltaT.H"

    runTime++;
    Info<< "Time = " << runTime.timeName() << nl << endl;

    // --- Pressure-velocity PIMPLE corrector loop
    while (pimple.loop())
    {
        #include "rhoEqn.H"
        #include "alphavPsi.H"
        #include "UEqn.H"

        // --- Pressure corrector loop
        while (pimple.correct())
        {
            #include "pEqn.H"
        }

        if (pimple.turbCorr())
        {
            turbulence->correct();
        }
    }

    runTime.write();
```

Reading maxAcousticCo number

Calculates and outputs the mean and max. Co numbers

Reset the timestep to maintain a constant max. Co number

Calculates the mixture density:

$$\frac{\partial \rho_m}{\partial t} + \nabla \bullet (\rho_m U) = 0$$

Calculates vapor mass fraction

$$\gamma = \frac{\rho - \rho_{l,sat}}{\rho_{v,sat} - \rho_{l,sat}}$$

Solves momentum equation

PIMPLE loop to solve pressure correction and turbulence equations

BICER, B., TANAKA, A., FUKUDA, T., & SOU, A. NUMERICAL SIMULATION OF CAVITATION PHENOMENA IN DIESEL INJECTOR NOZZLES, ILASS-ASIA, 2013.

“interPhaseChangeFoam” solver

“Solver for 2 incompressible, isothermal immiscible fluids with phase-change (e.g. cavitation). Uses a VOF (volume of fluid) phase-fraction based interface capturing approach.

The momentum and other fluid properties are of the "mixture" and a single momentum equation is solved.”

- Continuity equation: $$\frac{\partial \rho_m}{\partial t} + \nabla \bullet (\rho_m \mathbf{U}) = 0$$

ρ_m: *mixture density*

- Momentum equation: $$\frac{\partial \rho_m U}{\partial t} + \nabla \bullet (\rho_m \mathbf{U}\mathbf{U}) = -\nabla P + \nabla \bullet \left[\left(\mu_{eff} (\nabla \mathbf{U} + (\nabla \mathbf{U})^T \right) \right] + f_\sigma$$

f_σ: *surface tension force*

- Transport equation: $$\frac{\partial (\alpha \rho_l)}{\partial t} + \nabla \bullet (\alpha \rho_l \mathbf{U}) + \underbrace{\nabla \bullet [\alpha \mathbf{U}_c (1-\alpha)]} = R_c - R_e$$

$\mathbf{U}_c$ is called as artificial compression term, which is not zero only at the interface. It explains the shrinkage of the phase-interphase towards a sharper one

- Mixture viscosity and density: $\mu_m = (1-\alpha)\mu_v + \alpha \mu_l$ $\rho_m = (1-\alpha)\rho_v + \alpha \rho_l$

“interPhaseChangeFoam” members

“$FOAM_SOLVERS/multiphase/interPhaseChangeFoam

- alphaEqn.H
- alphaEqnSubCycle.H
- createFields.H
- interPhaseChangeFoam.C
- UEqn.H
- pEqn.H
- phaseChangeTwoPhaseMixtures Folder
  - Kunz
    - Kunz.C
    - Kunz.H
  - Merkle
    - Merkle.C
    - Merkle.H
  - SchnerrSauer
    - SchnerrSauer C
    - SchnerrSauer.H
  - phaseChangeTwoPhaseMixture
    - newPhaseChangeTwoPhaseMixture.C
    - phaseChangeTwoPhaseMixture.C
    - phaseChangeTwoPhaseMixture.H
- Make
  - files
  - options

Solves transport alpha eqn.

Main source code shows the flow chart of solver

Implemented cavitation models

Definitions for mixture two-phase

interPhaseChangeFoam.C file and solver flowchart

```
Info<< "\nStarting time loop\n" << endl;

while (runTime.run())
{
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "setDeltaT.H"

    runTime++;

    Info<< "Time = " << runTime.timeName() << nl << endl;

    // --- Pressure-velocity PIMPLE corrector loop
    while (pimple.loop())
    {
        #include "alphaControls.H"

        surfaceScalarField rhoPhi
        (
            IOobject
            (
                "rhoPhi",
                runTime.timeName(),
                mesh
            ),
            mesh,
            dimensionedScalar("0", dimMass/dimTime, 0)
        );

        mixture->correct();

        #include "alphaEqnSubCycle.H"
        interface.correct();

        #include "UEqn.H"

        // --- Pressure corrector loop
        while (pimple.correct())
        {
            #include "pEqn.H"
        }

        if (pimple.turbCorr())
        {
            turbulence->correct();
        }
    }

    runTime.write();
```

Reading the control parameters used by setDeltaT

Calculates and outputs the mean and max. Co numbers

Reset the timestep to maintain a constant max. Co number

Reading the control parameters for alpha equation

"$FOAM_TUTORIALS/multiphase/ interPhaseChangeFoam/system/ fvSolution

```
solvers
{
    "alpha.water.*"
    {
        cAlpha          0;
        nAlphaCorr      2;
        nAlphaSubCycles 1;

        MULESCorr       yes;
```

Solves alpha transport equation, and obtain new distribution

Correct the alpha boundary condition and also interface curvature

Solves the momentum equation

PIMPLE loop to solve pressure correction and turbulence equations

- For the detailed code explanation of the solver, refer to previous reports of this course:

A. Asnaghi, interPhaseChangeFoam tutorial and PANS turbulence model, MSc/PhD course in CFD with OpenSource software, (2013).

N. Lu, Tutorial: Solve cavitating flow around a 2D hydrofoil using a user modified version of interPhaseChangeFoam, MSc/PhD course in CFD with OpenSource software, (2008).

- For the example of test-case of this solver, refer to previous report of this course:

M. Andersen, A interPhaseChangeFoam tutorial, MSc/PhD course in CFD with OpenSource software, (2011).

Kunz Cavitation Model

$FOAM_SOLVERS/multiphase/interPhaseChangeFoam/ phaseChangeTwoPhaseMixures/Kunz/

- Transport equation: $$\frac{\partial(\alpha\rho_l)}{\partial t} + \nabla \bullet (\alpha\rho_l \mathbf{U}) + \nabla \bullet [\alpha \mathbf{U}_c (1-\alpha)] = \underbrace{R_c - R_e}$$

Evaporation source term

$$R_e = C_v \frac{\rho_v \alpha}{t_\infty (1/2 \rho_l U_\infty{}^2)} \min[0, P - P_v]$$

Condensation source term

$$R_c = C_c \frac{\rho_v \alpha^2 (1-\alpha)}{t_\infty}$$

- U_∞ is mean stream velocity (V_{inlet})
- t_∞ is mean flow time scale = L/ U_∞
- L is the characteristic length scale, (nozzle length)
- C_v and C_c are two-empirical

"$FOAM_TUTORIALS/multiphase/interPhaseChageFoam/constant/ transportProperties"

```
KunzCoeffs
{
    UInf            UInf    [0 1 -1 0 0 0 0]     20.0;
    tInf            tInf    [0 0 1 0 0 0 0]      0.005;
    Cc              Cc      [0 0 0 0 0 0 0]      1000;
    Cv              Cv      [0 0 0 0 0 0 0]      1000;
}
```

SECTION 5

Implementation Procedure

"TransportCavitatingFoam" implementation

"cavitatingFoam"

- Compressible barotropic model (equation of state)

$$\frac{D\rho_m}{Dt} = \Psi \frac{DP}{Dt}$$

$$\gamma = \frac{\rho_m - \rho_{l,sat}}{\rho_{v,sat} - \rho_{l,sat}}$$

alphavPsi.H

$$\rho_m = (1-\gamma)\rho_l^0 + (\gamma\Psi_v + (1-\gamma)\Psi_l)P_{sat} + \Psi(P - P_{sat})$$

"TransportCavitatingFoam"

- Incompressible model, with transport alpha equation

$$\frac{\partial(\alpha\rho_l)}{\partial t} + \nabla \bullet (\alpha\rho_l \mathbf{U}) + \nabla \bullet [\alpha \mathbf{U}_c (1-\alpha)] = R_c - R_e$$

alphaEqn.H + AlphaEqnSubCycle.H

$$\rho_m = (1-\alpha)\rho_v + \alpha\rho_l$$

- phaseChangeTwoPhaseMixtures
  Kunz.C
  Kunz.H

- So, first, all the files and addictions related barotropic compressibility model were removed, and then incompressible alpha transport equation with the Kunz cavitation model properly implemented. Look at the report for the intensive details about implementation.

SECTION 6

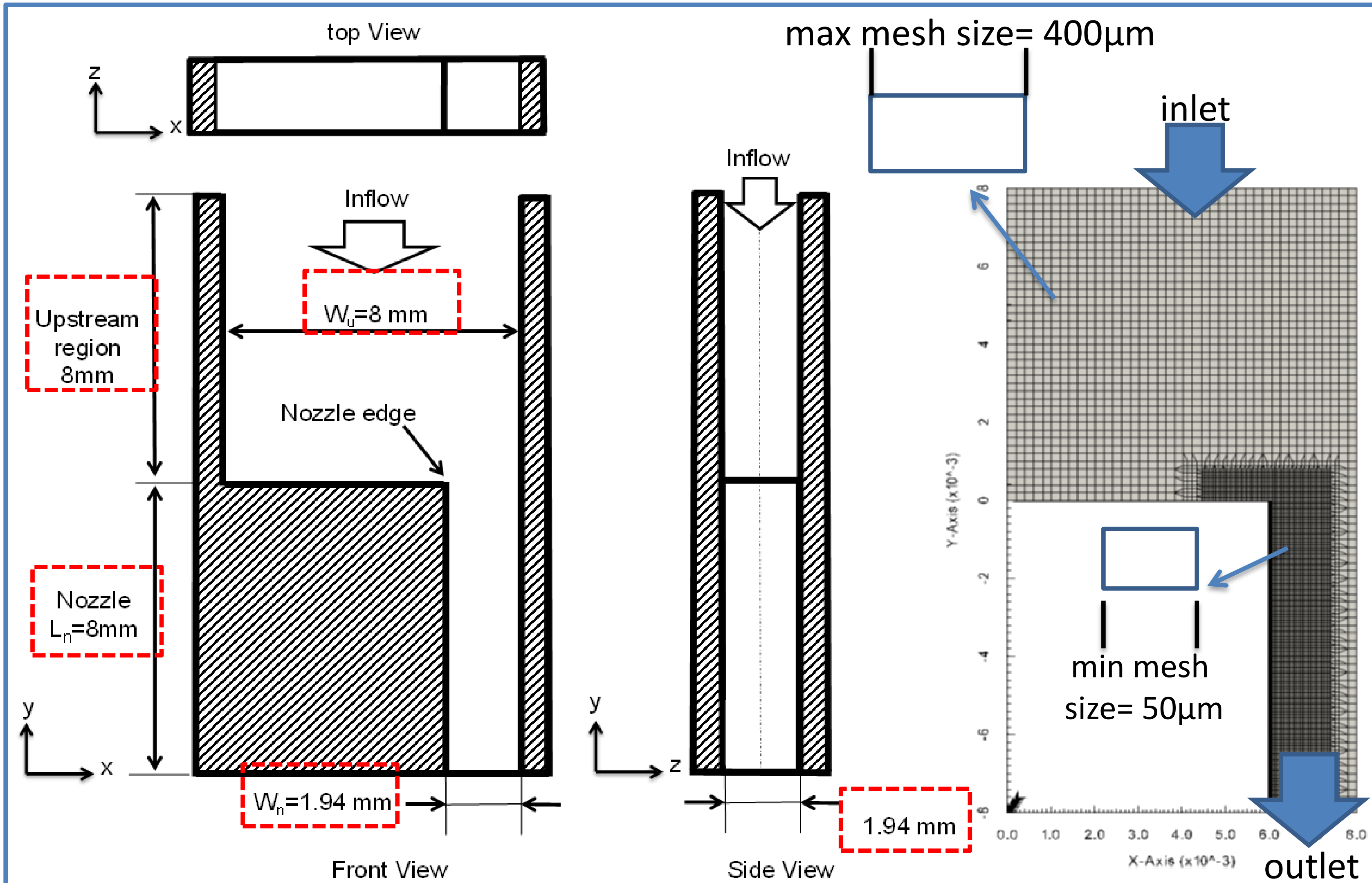
Test-case / Results

Test-case

- The cavitating turbulent flow inside a rectangular nozzle has been simulated using *TransportCavitatingFoam* new solver for the test case.
- A test case named *rectangular_nozzle_test_case*, which contains *0/*, *constant/* and *system/* folders, is already provided to users.
- Validation of the test case has been done using our previous experimental data in terms of the cavitation profile in the nozzle.

- The nozzle flow is considered as turbulent since Reynolds number is higher than 20,000. Therefore, turbulence effects have been introduced using RANS methods such as RNG k-ε model.

- Additionally, Kunz cavitation model empirical constants Cc and Cv are set to 1000.



(a) Nozzle geometry (b) 3-D Computational grid

Mesh was created via blockMesh and refineMesh

Mesh number: 73,100

$\Delta t = 10^{-7} \sim 10^{-8}$s, Co = 0.1

CPU = 1.5 days

Boundary Conditions

	Inlet	Outlet	Walls
U	fixedValue (=3.2 m/s)	inletOutlet	fixedValue (0 0 0) no-slip
P	zeroGradient	fixedValue (0.1MPa)	zeroGradient
k	fixedValue	inletOutlet	wallFunction
ε	fixedValue	inletOutlet	wallFunction

Before running the code, go to system/conrolDict file and change the application name to:

```
application      TransportCavitatingFoam
```

then go to test-case directory and run the code:

```
TransportCavitatingFoam &>log
```

Results

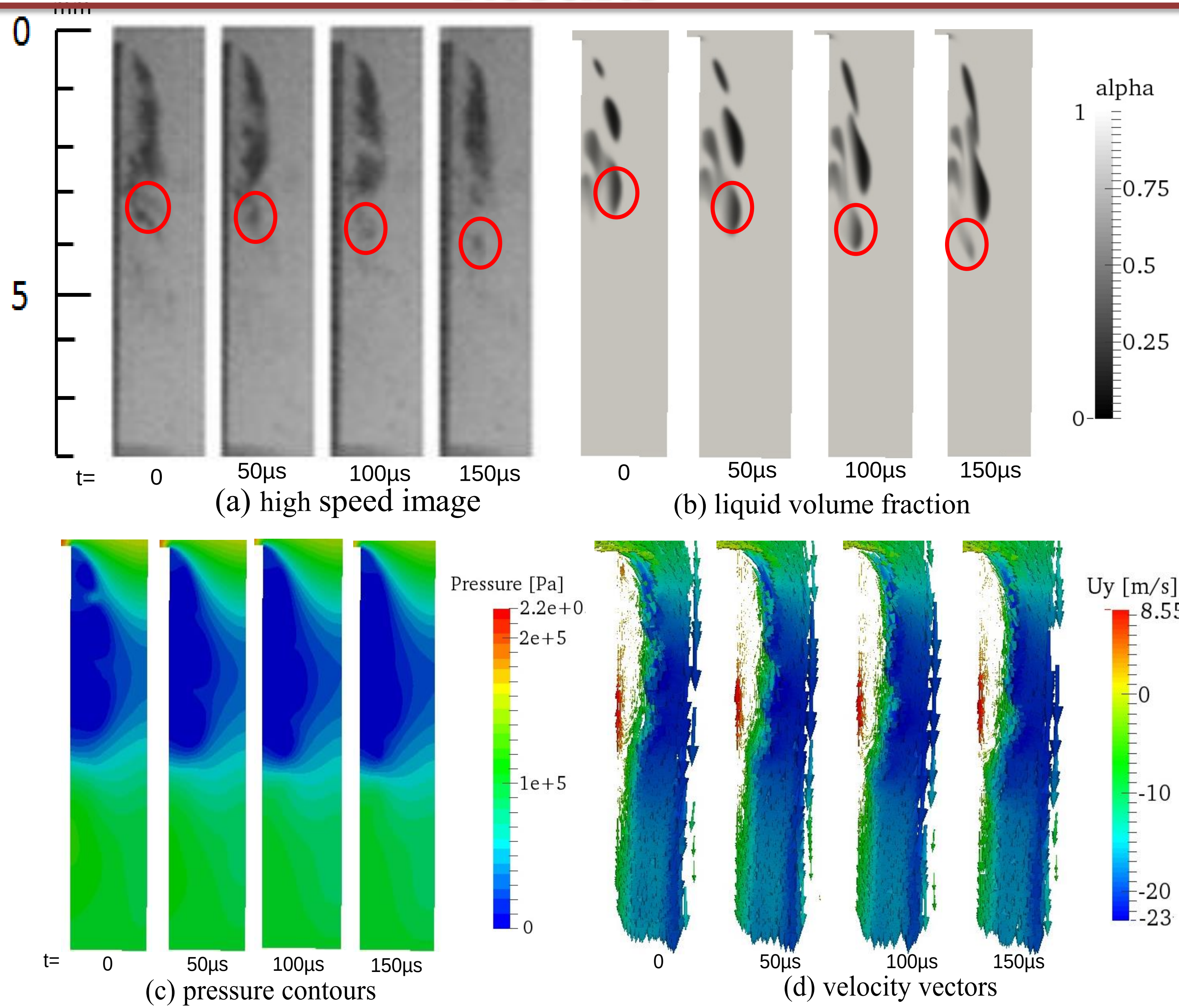
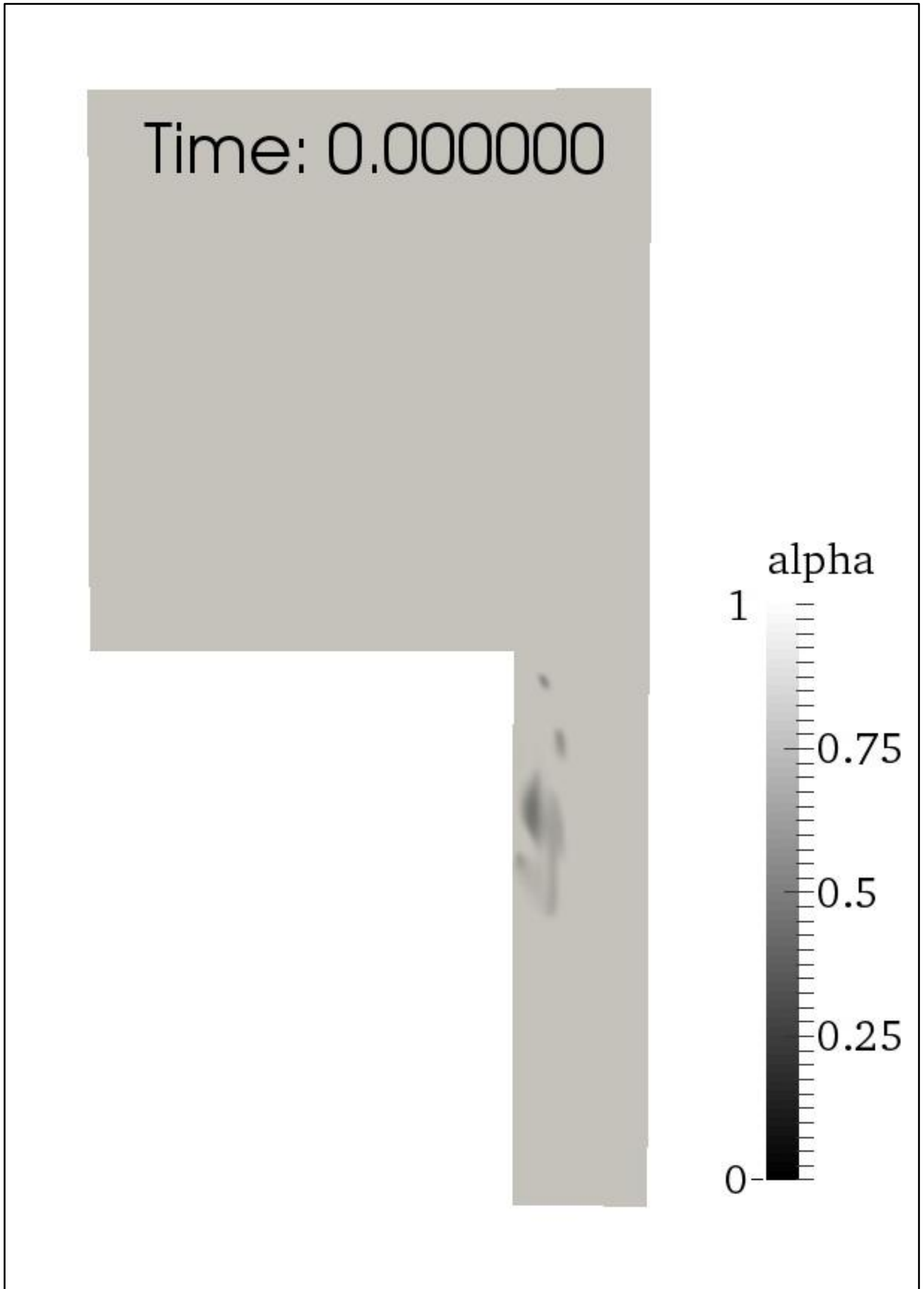


Fig. 2. Experimental and calculated results (P_{in} = 0.22 MPa, results are shown at every 50 μs)



Time: 0.000000
alpha
1
0.75
0.5
0.25
0

Energy & Fluid Science Lab.

THANKS FOR YOUR KIND LISTENING

ありがとうございました