Implementing a Zwart-Gerber-Belamri cavitation model

Marcus Jansson

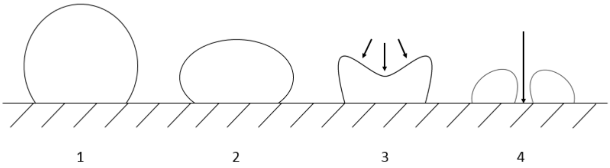
Applied Thermal and Fluid Sciences,
Linköping University,
Linköping, Sweden

2018-11-28

Agenda

- General background / cavitation models
- interPhaseChangeFoam
- The throttle tutorial
- Implementing a cavitation model
- Pre- and postprocessing (and solving)
- Study questions

Cavitation



Rayleigh-Plesset:

$$\frac{dR}{dt} = \sqrt{\frac{2}{3}\frac{p_b - p}{\rho_l}}$$

Schnerr-Sauer

"Original" Schnerr-Sauer model:

$$\dot{m} = \frac{\rho_v \rho_l}{\rho_m} \alpha_v \left(1 - \alpha_v\right) \frac{3}{R_B} \sqrt{\frac{2}{3} \frac{|p - p_v|}{\rho_l}}$$

Bubble radius:

$$\alpha_v = \frac{n_b \frac{4}{3} \pi R_B^3}{1 + n_b \frac{4}{3} \pi R_B^3}$$

Schnerr-Sauer (OpenFoam)

Mass transfer decomposition:

$$\dot{m} = \alpha_l \dot{m}_\alpha^- + (1 - \alpha_l)\, \dot{m}_\alpha^+ = \alpha_l \left(\dot{m}_\alpha^- - \dot{m}_\alpha^+ \right) + \dot{m}_\alpha^+$$

Mass transfer rates:

$$\dot{m}_\alpha^- = C_v \left(1 + \alpha_{nuc} - \alpha_l\right) \frac{3\rho_v\rho_l}{\rho_m R_B} \sqrt{\frac{2}{3\rho_l} \frac{1}{|p - p_v|}} min\left(p - p_v, 0\right)$$

$$\dot{m}_\alpha^+ = C_c \alpha_l \frac{3\rho_v\rho_l}{\rho_m R_B} \sqrt{\frac{2}{3\rho_l} \frac{1}{|p - p_v|}} max\left(p - p_v, 0\right)$$

Schnerr-Sauer (OpenFoam)

Nucleation site volume fraction:

$$\alpha_{nuc} = \frac{\frac{1}{6} n_b \pi d_{nuc}^3}{1 + \frac{1}{6} n_b \pi d_{nuc}^3}$$

Bubble radius:

$$R_B = \left(\frac{3}{4\pi n_b} \frac{1 + \alpha_{nuc} - \alpha_l}{\alpha_l} \right)^{\frac{1}{3}}$$

Zwart-Gerber-Belamri

Mass transfer decomposition:

$$\dot{m} = \alpha_l \dot{m}_\alpha^- + (1 - \alpha_l)\, \dot{m}_\alpha^+ = \alpha_l \left(\dot{m}_\alpha^- - \dot{m}_\alpha^+ \right) + \dot{m}_\alpha^+$$

Mass transfer rates:

$$\dot{m}_\alpha^- = -C_v r_{nuc} \frac{3\rho_v}{R_B} \sqrt{\frac{2}{3\rho_l} \frac{1}{|p - p_v|}} min\,(p - p_v, 0)$$

$$\dot{m}_\alpha^+ = +C_c \frac{3\rho_v}{R_B} \sqrt{\frac{2}{3\rho_l} \frac{1}{|p - p_v|}} max\,(p - p_v, 0)$$

r_{nuc} and R_B are constant

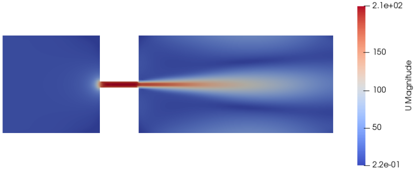
The solver

- Transient solver with generic turbulence model, based on the PIMPLE pressure correction.
- Solves for two isothermal, incompressible fluids using a Volume of Fluids (VoF) approach where momentum equations are only solved for the mixture.
- Mass transfer through cavitation with the models Merkle, Kunz, and Schnerr-Sauer.

throttle tutorial

"Standard" tutorial for cavitatingFoam
Inlet- and outlet pressure

- interPhaseChangeFoam solves for the liquid fraction
- Includes effects of g

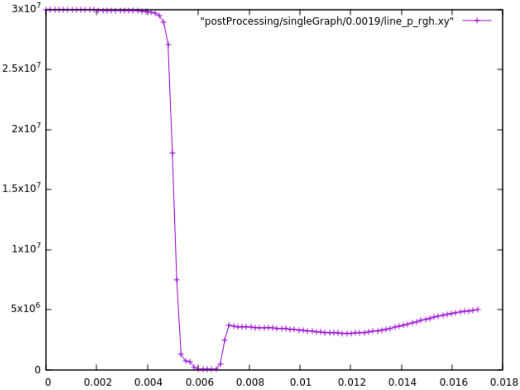


Velocity magnitude

More results

Liquid fraction and pressure

Water fraction



Pressure

phaseChangeTwoPhaseMixtures and mDotAlphal

- interPhaseChangeFoam uses alphaEqnSubCycle to correct the phase interfaces.
- alphaEqnSubCycle (and alphaEqn) includes vDotAlphal which is part of the phaseChangeTwoPhaseMixtures library
- vDotAlphal includes mDotAlphal which is different for every cavitation model

mDotAlphal in Schnerr-Sauer

Important member functions

- rRb
- alphaNuc
- pCoeff
- mDotAlphal
- mDotP

mDotAlphal

```
Foam::Pair<Foam::tmp<Foam::volScalarField> >
Foam::phaseChangeTwoPhaseMixtures::SchnerrSauer::mDotAlphal() const
{
    const volScalarField& p = alpha1_.db().lookupObject<volScalarField>("p");
    volScalarField limitedAlpha1(min(max(alpha1_, scalar(0)), scalar(1)));

    volScalarField pCoeff(this->pCoeff(p));

    return Pair<tmp<volScalarField> >
    (
        Cc_*limitedAlpha1*pCoeff*max(p - pSat(), p0_),

        Cv_*(1.0 + alphaNuc() - limitedAlpha1)*pCoeff*min(p - pSat(), p0_)
    );
}
```

pCoeff

```
Foam::tmp<Foam::volScalarField>
Foam::phaseChangeTwoPhaseMixtures::SchnerrSauer::pCoeff
(
    const volScalarField& p
) const
{
    volScalarField limitedAlpha1(min(max(alpha1_, scalar(0)), scalar(1)));
    volScalarField rho
    (
        limitedAlpha1*rho1() + (scalar(1) - limitedAlpha1)*rho2()
    );

    return
        (3*rho1()*rho2())*sqrt(2/(3*rho1()))
       *rRb(limitedAlpha1)/(rho*sqrt(mag(p - pSat()) + 0.01*pSat()));
}
```

Add a new model

Start by copying the existing cavitation models in `phaseChangeTwoPhaseMixtures` to your user directory. Copy the SS cavitation model and rename the files.

```
cp -r $FOAM_APP/solvers/multiphase/interPhaseChangeFoam/phaseChangeTwoPhaseMixtures \
$WM_PROJECT_USER_DIR/src/phaseChangeTwoPhaseMixtures
cd $WM_PROJECT_USER_DIR/src/phaseChangeTwoPhaseMixtures
cp -r SchnerrSauer Zwart
cd Zwart
mv SchnerrSauer.C Zwart.C
mv SchnerrSauer.H Zwart.H
```

Replace each 'SchnerrSauer' with 'Zwart' in Zwart.C and Zwart.H.

```
sed -i s/"SchnerrSauer"/"Zwart"/g Zwart.*
```

Make sure that `Make/files` ends with the following two lines:

```
...
Zwart/Zwart.C

LIB = $(FOAM_USER_LIBBIN)/libphaseChangeTwoPhaseMixtures
```

Zwart.H

Remove the private member data for `n_`, `dNuc_`, `rRb_` and `alphaNuc` in Zwart.H Add the new member data between `Cv_` and `p0_`.

```
    //- Nucleation site volume
    dimensionedScalar rNuc_;

    //- Nucleation site radius
    dimensionedScalar Rb_;
```

Zwart.C

Remove the constructors for `n_` and `dNuc_`. Add in the constructor, between `Cv_` and `p0_`:

```
rNuc_("rNuc", dimless, phaseChangeTwoPhaseMixtureCoeffs_),
Rb_("Rb", dimLength, phaseChangeTwoPhaseMixtureCoeffs_),
```

Remove the entire member functions `rRb` and `alphaNuc`. Replace the member functions `pCoeff` and `mDotAlpha`

pCoeff:

```
Foam::tmp<Foam::volScalarField>
Foam::phaseChangeTwoPhaseMixtures::Zwart::pCoeff
(
    const volScalarField& p
) const
{
    return
        (3*rho2())*sqrt(2/(3*rho1()))
        /(Rb_*sqrt(mag(p - pSat()) + 0.01*pSat()));
}
```

Zwart.C

mDotAlphal:

```
Foam::Pair<Foam::tmp<Foam::volScalarField> >
Foam::phaseChangeTwoPhaseMixtures::Zwart::mDotAlphal() const
{
    const volScalarField& p = alpha1_.db().lookupObject<volScalarField>("p");
    volScalarField limitedAlpha1(min(max(alpha1_, scalar(0)), scalar(1)));

    volScalarField pCoeff(this->pCoeff(p));

    return Pair<tmp<volScalarField> >
    (
        Cc_*pCoeff*max(p - pSat(), p0_),

        Cv_*rNuc_*pCoeff*min(p - pSat(), p0_)
    );
}
```

Zwart.C

mDotP:

```
Foam::Pair<Foam::tmp<Foam::volScalarField> >
Foam::phaseChangeTwoPhaseMixtures::Zwart::mDotP() const
{
    const volScalarField& p = alpha1_.db().lookupObject<volScalarField>("p");
    volScalarField pCoeff(this->pCoeff(p));

    volScalarField limitedAlpha1(min(max(alpha1_, scalar(0)), scalar(1)));

    return Pair<tmp<volScalarField> >
    (
        Cc_*(1.0 - limitedAlpha1)*pos(p - pSat())*pCoeff,

        (-Cv_)*rNuc_*limitedAlpha1*neg(p - pSat())*pCoeff
    );
}
```

Zwart.C

Finally, remove n_ and dNuc_ from the read function and add rNuc_ and Rb_. Compile the class again using wmake.

```
bool Foam::phaseChangeTwoPhaseMixtures::Zwart::read()
{
    if (phaseChangeTwoPhaseMixture::read())
    {
        phaseChangeTwoPhaseMixtureCoeffs_ = optionalSubDict(type() + "Coeffs");

    phaseChangeTwoPhaseMixtureCoeffs_.lookup("Cc") >> Cc_;
        phaseChangeTwoPhaseMixtureCoeffs_.lookup("Cv") >> Cv_;
        phaseChangeTwoPhaseMixtureCoeffs_.lookup("rNuc") >> rNuc_;
        phaseChangeTwoPhaseMixtureCoeffs_.lookup("Rb") >> Rb_;

        return true;
    }
    else
    {
        return false;
    }
}
```

Pre-processing

Go to the run directory and copy the throttleSchnerr tutorial (found in caseFiles). Remove the old time directories.

```
run
cp -r throttleSchnerr throttleZwart
cd throttleZwart
rm -r 0.*
rm -r postProcessing
```

In `constant/transportProperties`, change `phaseChangeTwoPhaseMixture` from `SchnerrSauer` to `Zwart` to use the new cavitation model. Add the new model coefficients after the section with `SchnerrSauerCoeffs`.

```
ZwartCoeffs
{
    Cc      0.01;
    Cv      50;
    rNuc    5.0e-04;
    Rb      1.0e-06;
}
```

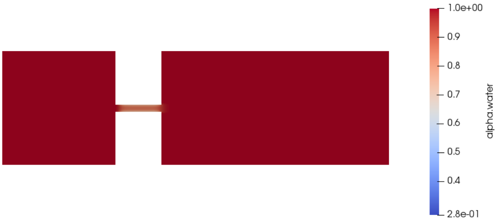
Note the addition of `rNuc` and `rRb` and the values for `Cc` and `Cv`

Results and post-processing

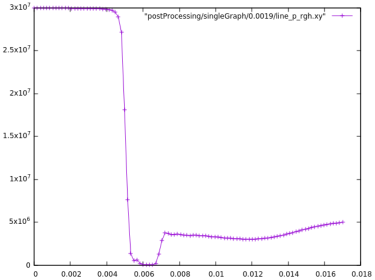
Solve the case, now with the new ZGB cavitation model.

```
interPhaseChangeFoam >& log_run&
```

Compare the results to the previous model



Water fraction



Pressure

Summary

- Cavitation models
- Solving the throttle with Schnerr-Sauer cavitation model
- Implementation of a new cavitation model
- Solving the throttle with Zwart et. al. cavitation model
- Comparing models
- Questions?