Implementation of aeroacoustic solver for weakly compressible flows

Anandh Ramesh Babu

Department of Applied Mechanics,
Chalmers University of Technology,
Gothenburg, Sweden

Aeroacoustics

- Aeroacoustics is the study of flow induced sound.
- Pioneered in 1950's by James Lighthill.
- The sources of sound are turbulent wakes, detached boundary layers and vortex structures and so on.
- The main reason behind studying this field is to reduce noise generated by wind flow.

Computation of Aeroacoustic field

- Traditionally, computation was reliant on experimental methods.
- Lighthill's wave equation was derived by rewriting flow equations which yielded physical analytical solutions.
- Later, further extensions such as Curle's equations and FWH equations were derived from Lighthill's equation.
- These equations were computed by integrating the source terms in a Green's integral.

Computational Aeroacoustics (CAA)

With the development of CFD and computational power, CAA developed into a field on its own.
In CAA, 2 approaches are used:

1. Direct Methods : It uses the most exact and straightforward methodology where a transient simulation is performed to determine the source terms and the propagation of sound waves. Very high computational effort is required.
2. Hybrid Methods : Taking the source term from CFD analysis the propagation of sound waves are predicted. This includes scale modeling and computational effort is reduced.

Hybrid Methods

- Application of these methods would result in decoupling the flow field and acoustic field.
- This means the flow is independent of the acoustic field.
- By doing this, the problem can be divided into two:
  1. Solution of flow field.
  2. Propagation of sound waves.

Solution methodology

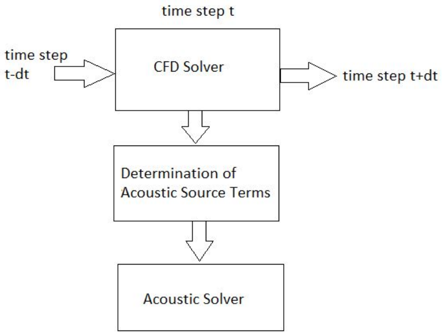

Figure: Solution methodology

Acoustic transport equation

- For weakly compressible flows or incompressible flows, transport equation with pressure takes precedence over density as fluctuations in density is negligible.
- So, a transport equation for acoustic pressure is created and coupled with fluctuations in pressure from the CFD simulations.
- The transport equation for acoustic pressure is as follows:

$$\frac{\partial^2 p_a}{\partial t^2} - c_\infty^2 \frac{\partial^2 p_a}{\partial x^2} = -\frac{\partial^2 p'}{\partial t^2} \quad (1)$$

  where p_a is the acoustic pressure, c_∞ is the speed of sound and p' is the fluctuation in pressure from the CFD simulation.
- This equation is implemented in the solver to solve to the acoustic pressure field.

rhoPimpleAdiabaticFoam

- The acoustic Solver is created as an extension of the solver, 'rhoPimpleAdiabaticFoam'.
- This solver is used in applications with low Mach number.
- The solver uses PIMPLE algorithm method for time-resolved simulations.
- Rhie-Chow interpolation is adopted.

Directory structure

Upon initializing OpenFOAM in the terminal window,

`OFv1806`

Type

`cd $FOAM_APP/solvers/compressible/rhoPimpleAdiabaticFoam`

```
rhoPimpleAdiabaticFoam
 ├── rhoPimpleAdiabaticFoam.C
 ├── UEqn.H
 ├── pEqn.H
 ├── EEqn.H
 ├── resetBoundaries.H
 ├── createFields.H
 └── Make
      ├── files
      └── options
```

Solver methodology

In the file, 'rhoPimpleAdiabaticFoam.C', the headers necessary for the solver to run are present.

```
#include "fvCFD.H"
#include "fluidThermo.H"
#include "turbulentFluidThermoModel.H"
#include "bound.H"
#include "pimpleControl.H"
#include "fvOptions.H"
#include "ddtScheme.H"
#include "fvcCorrectAlpha.H"
```

Solver methodology

Inside the main function, several classes are added for postporcessing, time and mesh control.

```
main()
{
    #include "postProcess.H"
    #include "addCheckCaseOptions.H"
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"
    #include "createControl.H"
    #include "createTimeControls.H"
    #include "createFields.H"
    #include "createFvOptions.H"
    #include "initContinuityErrs.H"
```

Solver Methodology

'runTime' object is initialized in 'createTime' header and the time loop is initiated using a while loop.

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

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

         if (pimple.nCorrPIMPLE() <= 1)
        {
            #include "rhoEqn.H"
         }
```

Solver Methodology

```
// --- Pressure-velocity PIMPLE corrector loop
  while (pimple.loop())
 {
     U.storePrevIter();
     rho.storePrevIter();
     phi.storePrevIter();
     phiByRho.storePrevIter();

     #include "UEqn.H"

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

Solver Methodology

```
            #include "EEqn.H"

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

         runTime.write();

         runTime.printExecutionTime(Info);
 }
Info<<"End \n"<<endl;
return 0;
}
```

createFields.H

- The file createFields.H is responsible for the declaration and initialization of all the fields that is used in the solver.
- All the constants and fields that are given as input to the solver are done through dictionaries.
- The variables that are initialized, obtain their initial and boundary values from the user dictionary present '0' folder in case directory.
- This file also includes moving reference frames and compressibility ratios.

Copying files to user directory

- For the solver to be implemented, the files relevant to the solver to copied to the user directory.
- In the terminal window, type `ufoam` and execute

```
cp -r --parents $FOAM_APP/solvers/
compressible/rhoPimpleAdiabaticFoam .
```

- The directory and files are renamed using the following commands :

```
cd applications/solvers/compressible
mv rhoPimpleAdiabaticFoam rhoPimpleAdiabaticAcousticFoam
cd rhoPimpleAdiabaticAcousticFoam
mv rhoPimpleAdiabaticFoam.C rhoPimpleAdiabaticAcousticFoa
sed -i s/'rhoPimpleAdiabaticFoam'/
'rhoPimpleAdiabaticAcousticFoam'/g Make/files
sed -i s/'FOAM_APPBIN'/'FOAM_USER_APPBIN'/g Make/files
```

Creating fields for acoustic solver

- In addition to the variables already declared and initialized in createFields.H, a few more variables are needed for the functioning of the acoustic solver.
- These variables are pAcoustic (acoustic pressure), pMean (mean pressure field), pFluc (fluctuating pressure field) and cInf (speed of sound).
- All the variable are of type,'volScalarField' that are defined on the mesh.

Creating fields for acoustic solver

The following lines of code are pasted in 'createFields.H' before the line 'include "createMRF.H"'.

```
Info<< "Creating field pMean\n" << endl;
volScalarField pMean
(
    IOobject
    (
        "pMean",
        runTime.timeName(),
        mesh,
        IOobject::READ_IF_PRESENT,
        IOobject::AUTO_WRITE
    ),
    mesh,
    dimensionedScalar(p.dimensions())
);
```

Creating fields for acoustic solver

```
Info<< "Creating field pFluc\n" << endl;
volScalarField pFluc
(
    IOobject
    (
        "pFluc",
        runTime.timeName(),
        mesh,
        IOobject::READ_IF_PRESENT,
        IOobject::AUTO_WRITE
    ),
    mesh,
    dimensionedScalar(p.dimensions())
);
```

Creating fields for acoustic solver

```
Info<< "Creating field pAcoustic\n" << endl;
volScalarField pAcoustic
(
    IOobject
    (
        "pAcoustic",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

Creating fields for acoustic solver

```
Info<< "Creating field cInf\n" << endl;
volScalarField cInf
(
    IOobject
    (
        "cInf",
        runTime.timeName(),
        mesh
    ),
    mesh,
    dimensionedScalar(U.dimensions())
);
cInf = sqrt(thermo.Cp()/thermo.Cv()*(thermo.Cp()
        -thermo.Cv())*T);
scalar timeIndex = 1;
```

Creating fields for acoustic solver

```
IOdictionary acousticSettings
(
    IOobject
    (
        "acousticSettings",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ_IF_MODIFIED,
        IOobject::NO_WRITE
    )
);
```

Creating fields for acoustic solver

```
dimensionedScalar tAc
(
    "tAc",
    dimTime,
    acousticSettings.lookup("tAc")
);
dimensionedScalar nPass
(
    "nPass",
    dimless,
    acousticSettings.lookup("nPass")
);
```

Once these variables are declared, the file is saved and closed.

Creating acousticSolver.H

A new file is created in the directory with the name 'acousticSolver.H' by typing `vi acousticSolver.H` and the following lines of code are pasted.

```
//acoustic solver
if(runTime.time()>tAc)
{
if(timeIndex == 1)
{
pMean = p;
pMean.storeOldTime();
timeIndex++;
}
else
{
Info<< "Calculating fields pMean and pFluc\n" << endl;
```

Creating acousticSolver.H

```
pMean = (pMean.oldTime()*(runTime.time()-
runTime.deltaT())+p*runTime.deltaT())/(runTime.time());
pMean.storeOldTime();
if(runTime.time()>(tAc*nPass))
{
pFluc = p - pMean;
Info<< "Solving the wave equation for pAcoustic\n" << endl;
fvScalarMatrix pAcousticEqn
(
  fvm::d2dt2(pAcoustic) - sqr(cInf)*fvm::laplacian(pAcoustic)
  + fvc::d2dt2(pFluc)
 );

solve(pAcousticEqn);
} }
}
```

Compiling the solver

- The acousticSolver.H file needs to be added in the *.C file. This is done by typing `#include "acousticSolver.H"` in 'rhoPimpleAdiabaticAcousticFoam.C' file before the line runTime.write() at the end of the time loop.
- After this step, the solver can be compiled using the command, `wmake`.

Importing test case

- The case tested is flow past a wedge at a freestream velocity of 95m/s at 1 atm stagnation pressure and 297.3K stagnation temperature.
- The case files are found with the name,'prism', in sonicFoam tutorial under RAS folder.
- The files are copied to the user run directory using the command, `cp -r $FOAM_TUTORIALS/compressible/sonicFoam/RAS/prism $FOAM_RUN`
- The folder is accessed by typing, `cd $FOAM_RUN/prism`

Case geometry

The case geometry is modified in blockMeshDict. The file is opened using the command, `vi system/blockMeshDict` and the vertices are modified to obtain the domain shown in Figure 2.

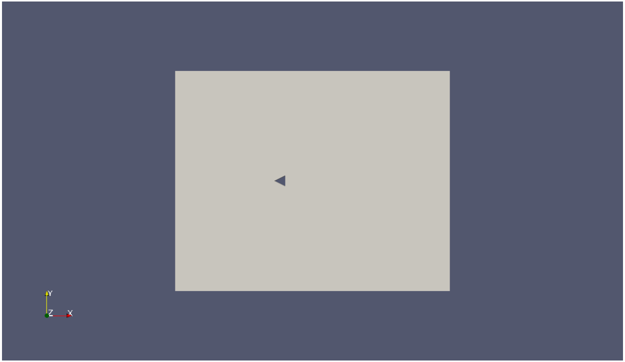

Figure: 2. Test domain

Modifications to controlDict

Open controlDict using the command, `vi system/controlDict`

- Under application, the solver is renamed to 'rhoPimpleAdiabaticAcousticFoam'.
- 'endTime' is set to 0.15.
- 'deltaT' is set to 5e-06.
- 'writeInterval' is set to 1e-04.

It is noted that that the deltaT value is set in consideration of mesh and courant criterion. The file is saved and closed.

Modifications to fvSchemes

Open fvSchemes using the command, `vi system/fvSchemes`.

- The newly added acoustic equation as mentioned in eqn. 1, contains a second time derivative term and a laplacian term.
- The laplacian term by default is set to 'Gauss linear corrected'.
- The following code is copied to the file after ddtScheme,

```
d2dt2Schemes
{
    default Euler;
}
```

The file is saved and closed.

Modifications to fvSolution

Open fvSolution using the command, `vi system/fvSolution`.

- A solver needs to be set for the variable 'pAcoustic'.
- This variable solved for requires a solver capable of handling non-symmetric matrices.
- So PBiCGStab solver with a DILU preconditioner was set with a tolerence of 1e-06.

```
 pAcoustic
{
 solver PBiCGStab;
 preconditioner DILU;
 tolerance  1e-6;
 relTol 0;
}
```

The file is saved and closed.

Modifications in constant folder

One modification is made in 'thermophysicalProperties' and 'turbulenceProperties' each in the constant folder. Additionally, a new file 'acousticSettings' is created.

- ***thermophysicalProperties***

  The following code is added to the mixture section.

  ```
   equationOfState
  {
     p0            101325;
     T0            297.3;
  }
  ```

- ***turbulenceProperties***

  The RAS model is changed to 'RNGkEpsilon' model.

Modifications in constant folder

- *acousticSettings*

  The following settings are mentioned.

```
tAc     0.0105;
nPass   2;
```

Creating fields

- pAcoustic field must be created in the 0/ folder
- pAcoustic field is created by typing `vi 0/pAcoustic`.
  internalField : 0
  boundary condition : waveTransmissive
  boundary value : 0.
  This non-reflecting boundary condition is used on all the outer boundaries and 'zeroGradient' boundary condition is on the prism.

Modifying existing fields

- The velocity is set to 95m/s in the file 0/U.
- By using isentropic flow relations, the corresponding pressure was determined to be 96319.74 Pa and the temperature was 293K. These are correspondingly set in 0/p and 0/T.

Running case

- The case can be run using the commands, `blockMesh` and `rhoPimpleAdiabaticFoam` or by executing these two commands in an Allrun script.
- The solution of the case takes a while to complete as the speed of the flow is quite high and to view wake structures, atleast 4-5 flow passes must be completed. The wake structures can be captured with a reasonable mesh refinement while satisfying the courant number criterion for transient simulation.

Solution

On completion, the results can be viewed using the command `paraFoam`. The field is set to pAcoustic and played. The initial solution is disregarded as the flow is undeveloped. Figure 3 denoted the acoustic pressure field at time t=0.065s.

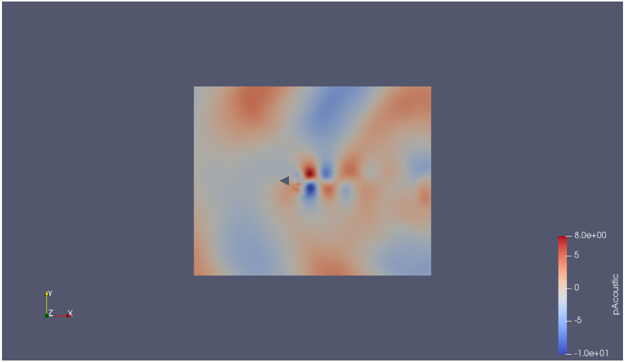

Figure: pAcoustic Field at t=0.065s

Solution

Figure 4 denoted the acoustic pressure field at time t=0.15s.

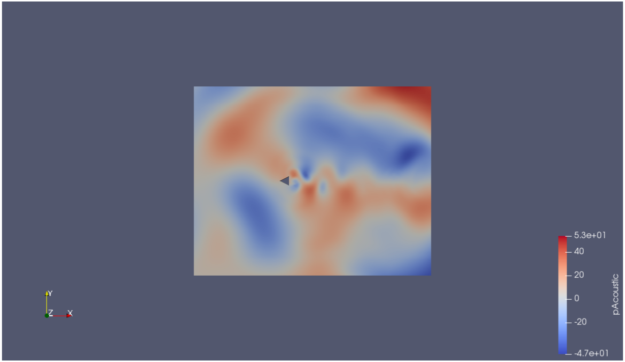

Figure: pAcoustic Field at t=0.15s

Results

- The pAcoustic field pattern is synonymous to pFluc and in turn p.
- As seen from the results, a maximum acoustic pressure level of around 53 Pa is seen.
- The corresponding sound level intensity is 128 dB which seems reasonable. But this implementation must be validated.
- This implementation is suitable only for weakly compressible and incompressible regimes and beyond that would yield unphysical results.

Conclusion

Questions?

Thank You