AeroFoam July 28, 2008 Quickstart Guide

G. Romanelli, E. Serioli

INTRODUCTION

AeroFoam is a cell-centered Finite Volume (FV) solver developed in OpenFOAM v.1.4.1 for the accurate numerical simulation of multi-dimensional time-dependent inviscid and viscous compressible fluid flows. The toolboxes CAE2D and CAE3D, to effectively solve aeroservoelastic problems in the transonic and supersonic regimes, are also included.

REMARK: In order to make the AeroFoam solver work on the non-development OpenFOAM v.1.4.1 release, it is necessary to correct the minor bug at line 353 of file `$HOME/OpenFOAM/OpenFOAM-1.4.1/src/OpenFOAM/meshes/polyMesh/polyPatches/polyPatch/polyPatch.H` changing the minus sign with a plus sign, so that line 353 becomes: `return l + start_;`

GENERAL OPTIONS

To choose between the implemented time and space integration schemes, mesh handling strategies, etc... the following `AeroFoamSchemes` scope must be added to the `fvSchemes` input file:

```
AeroFoamSchemes
{
```

Mathematical model. *The user can choose between the following mathematical models (`flowType`): Euler equations (`Euler`) or Navier-Stokes equations (`NS`).* REMARK: `NS` option is still experimental and must be benchmarked.

```
    flowType                Euler;
```

Time integration. *The user can choose between the following time integration schemes (`timeScheme`): Explicit Euler (`EE`), explicit Runge-Kutta of order up to $p = 4$ (`RK2`, `RK3`, `RK4`) and a low storage Runge-Kutta scheme of order $p = 4$, suitable for steady-state solution only (`RK4LS`). A Local Timestepping (LT) strategy is also implemented to speed-up the convergence to the steady-state solution: the user must define the maximum local Courant number (`CourantMax`) and the number of iterations to linearly increase the local Courant number (`CourantSteps`). If `CourantMax` field is non-positive this option is ignored.*

```
    timeScheme              EE;
    CourantMax              0.0;
    CourantSteps            25;
```

Space integration. *The user can choose between the following monotone 1^{st} order accurate expressions for the numerical fluxes vector (`MonotoneFlux`): Approximate Riemann Solver (`Roe`), Advection Upstream Splitting Method (`AUSM`), Convective Upwind and Split Pressure (`CUSP`), Harten-Lax-vanLeer original (`HLL`) and corrected (`HLLC`) and Osher-Solomon (`OS`). In order to converge to a numerical solution satisfying the entropy condition some Entropy Fix (EF) strategies are implemented (`entropyFix`): Harten and Hyman (`HH1`, `HH2` and `HH2b`) for the `Roe` solver and Davies (`D1` and `D2`), Enfieldt-Roe (`ER`), LeVeque (`LV`) and Toro (`Toro`) for the `HLL` and `HLLC` solvers. To yield high resolution in presence of discontinuities the user can choose between the following 2^{nd} order accurate expressions for the numerical fluxes vector (`HighResolutionFlux`): Lax-Wendroff (`LW`) and Jameson-Schmidt-Turkel (`JST`). To prevent spurious waves a suitable flux limiter (`fluxLimiter`) should be chosen between VanLeer (`VL`), MinMod (`MinMod`), SuperBee (`SB`) and Monotonized Central (`MC`).* REMARK: The following combination is the most accurate and robust.

```
    MonotoneFlux         Roe;
    entropyFix           HH2;
    HighResolutionFlux   LW;
    fluxLimiter          VL;
```

Boundary conditions. *To set the boundary conditions in a cell-centered Finite Volume scheme the numerical solution must be extrapolated on the boundary: the user can choose between a more robust constant (`0`) or a more accurate linear (`1`) extrapolation strategies (`extrapolateBC`).*

```
    extrapolateBC        1;
```

Residual. *The user can choose between the following residual evaluation strategies (`residualNorm`):* L^2 *norm (`L2`),* L^1 *norm (`L1`) or* L^∞ *norm (`Loo`). The residual history is written runtime on the `residuals.txt` file in the `Log` folder placed at the same level of the problem folder. The numerical simulation is automatically stopped when the residual is less than the prescriped `minResidual` tolerance. If `minResidual` field is non-positive this option is ignored.*

```
    residualNorm         L1;
    minResidual          -1;
```

Extended cells connectivity. *To initialize the extended cells auxiliary data structures (necessary for the* 2^{nd} *order accurate numerical fluxes) the user can choose between the following search algorithms (`adaptiveConnectivity`): incremental more suitable for structured meshes (`1`) or non-incremental more suitable for unstructured meshes (`0`).* REMARK: The extended cells internal search algorithm fails for particularly badly shaped volumes and this can affect numerical solution accuracy a little. *Otherwise the user can load the auxiliary data structures from the file `LR2LLRR.txt` in the `Data` folder placed at the same level of the problem folder. This file is organized as follows: each line corresponds to a mesh face and adjacent left (*L*), right (*R*), extended left (*LL*) and extended right (*RR*) cells ids are stored. Depending on the `loadConnectivity` field, the auxiliary data structures are only read (`1`), they are read and written (`2`) or they are only written (`-1`) in the `Log` folder placed at the same level of the problem folder. The user can check if the auxiliary data structures are correct using the option `testMesh=1` and plotting in* ParaView *the `cell T` field (adjacent* L, R, LL *and* RR *cells are set to* 100, 200, 300 *and* 400 *values respectively).* REMARK: This option creates a lot of fictitious time folders, one for each mesh cell, and it is useful for 2D applications mainly.

```
    adaptiveConnectivity 0;
    loadConnectivity     0;
    testMesh             0;
```

Tests. *The user can check if the input parameters are correct before execution using the option `testInput=1`.*

```
    testInput            1;
}
```

CAE2D TOOLBOX OPTIONS

To use the CAE2D toolbox for the aeroservoelastic analysis of wing sections with the degrees of freedom of plunging h, pitching α and trailing edge control surface rotation δ, the following `CAE2DToolbox` scope must be added to the `fvSchemes` input file:

```
CAE2DToolbox
{
```

Reference frame. *The user must define the frame of reference (`referenceFrame`) in which the aerodynamic loads are computed choosing between body axes (`body`) and wind axes (`wind`). Then the wing section dimensional reference chord (`Cref`) and the dimensional aerodynamic reference length (`Lref`) must be defined together with the dimensional coordinates of the elastic axis location (`Xref.x()` and `Xref.y()`) and of the hinge axis location (`Xhinge.x()` and `Xhinge.y()`) measured in the global frame of reference.* REMARK: The following data refer to the AGARD 702 benchmark test problem of harmonically pitching NACA64A010 wing section.

```
    referenceFrame  body;
    Cref            1.00;  // [m]
    Lref            0.50;  // [m]
    Xref.x()        0.248; // [m]
    Xref.y()        0.00;  // [m]
    Xhinge.x()      0.75;  // [m]
    Xhinge.y()      0.00;  // [m]
```

Solution type. *The user can choose between the following types of solutions (`solutionType`): wing section fixed (0), wing section forced motion (1), wing section $h-\alpha$ free motion (2) and trailing edge control surface δ free motion (3).* REMARK: The wing section movement is simulated by means of a transpiration velocity without deforming the mesh runtime. The mesh can be deformed in the post-processing stage with the utility showDisplacement.

```
    solutionType    1;
```

Forced motion. *For the wing section degrees of freedom $h-\alpha-\delta$ the user can choose between the following forced motion laws (e. g. for `Mov_h`): an harmonic forced motion (1), namely:* $h(t) = \texttt{A0\_h}+\texttt{A1\_h}\sin[\,2\,\pi\,\texttt{f\_h}\,(\,t+\texttt{tau\_h}\,)\,]$ *or a ramp forced motion (2), namely:* $h(t) = \texttt{A0\_h}+\texttt{A1\_h}\,(\,t+\texttt{tau\_h}\,)$ REMARK: The following data refer to the AGARD 702 benchmark test problem of harmonically pitching NACA64A010 wing section.

```
//.....................................Plunging of elastic axis (h)
    Mov_h           0;
    A0_h            0.0; // [m]
    A1_h            0.0; // [m]
    f_h             0.0; // [Hz]
    tau_h           0.0; // [s]
//.....................................Pitching around elastic axis (a)
    Mov_a           1;
    A0_a            0.0;      // [rad]
    A1_a            0.017628; // [rad]
    f_a             17.4166;  // [Hz]
    tau_a           -0.1;     // [s]
//.....................................Control surface rotation around hinge axis (d)
    Mov_d           0;
    A0_d            0.0; // [rad]
    A1_d            0.0; // [rad]
    f_d             0.0; // [Hz]
    tau_d           0.0; // [s]
```

Pitching α and plunging h free motion. *In order to simulate the wing section $h-\alpha$ free motion the user must define the wing section mass (`m`), the distance between the elastic axis and the center of gravity locations (`d_EACG` $= x_{\mathrm{EA}} - x_{\mathrm{CG}}$), the wing section moment of inertia around the center of gravity (`I_CG`) and the translational and rotational stiffnesses (`K_hh` and `K_aa` respectively).*

```
    m               0.0; // [kg]
    d_EACG          0.0; // [m]
    I_CG            0.0; // [kgm^2]
    K_hh            0.0; // [N/m]
    K_aa            0.0; // [Nm]
```

Trailing edge control surface δ free motion. *In order to simulate the trailing edge control surface δ free motion the user must define the moment of inertia around the hinge axis (`I_dd`), the structural damping (`C_dd`), the structural stiffness (`K_dd`) and optionally the maximum trailing edge control surface rotation δ_{max} (`d_max`).*

```
    I_dd            0.0; // [kgm^2]
    C_dd            0.0; // [Nms]
    K_dd            0.0; // [Nm]
    d_max           0.0; // [rad]
}
```

CAE3D TOOLBOX OPTIONS

To use the CAE3D toolbox for the aeroservoelastic analysis of wings the following `CAE3DToolbox` scope must be added to the `fvSchemes` input file:

```
CAE3DToolbox
{
```

Aerodynamic model. *The user must define the aerodynamic dimensional reference surface (`Sref_a`) and length (`Lref_a`) together with the dimensional coordinates of a reference point location for the computation of the aerodynamic moment coefficients (`Xref_a.x()`, `Xref_a.y()` and `Xref_a.z()`).* REMARK: The following data refer to the AGARD 445.6 benchmark test problem.

```
    Sref_a          0.3513; // [m^2]
    Lref_a          0.2338; // [m]
    Xref_a.x()      0.4938; // [m]
    Xref_a.y()      0.3547; // [m]
    Xref_a.z()      0.0;    // [m]
```

Structural model. *The information about the structural model is stored in the files `StructuralModel.vertices` and `StructuralModel.elements` in the `Data` folder placed at the same level of the problem folder, together with the files `StructuralModel.mode<id>` which contain the natural frequency, the generalized mass, the generalized damping and the modal shape for each `<id>`-th mode. In order to build a Reduced Order Model (*ROM*) of the aerodynamic system, it is necessary to store the Generalized Aerodynamic Forces (*GAF*) time history consequent to a forced generalized motion law.* REMARK: The wing movement is simulated by means of a transpiration velocity without deforming the mesh runtime. The mesh can be deformed in the post-processing stage with the utility showDisplacement. *The user can choose between the following types of solutions (`solutionType`): wing fixed (`0`), wing static iterative coupled modal solution (`1`), wing dynamic coupled modal solution (`2`), wing forced modal solution with a blended real step generalized motion law (`3`), wing forced modal solution with an ideal step generalized motion law (`4`). First of all the user must define the total numer of modes to be taken into consideration (`Nmodes`) and the id of the active mode (`NactiveMode`). Then for `solutionType=3` the user must define the maximum reduced frequency dynamically important (`kMax`) and the maximum perturbation on the velocity field (`epsU`); otherwise for `solutionType=4` the user must define the maximum generalized displacement (`qMax`). Eventually the user must define the dimensional time to start the forced modal motion at (`forcingStartTime`) and the dimensional time interval to print the* GAF *time history on the files `NAEMOGenForces_m<id>.inf` and `NAEMOGenForces_m<id>.wrk` in the `Log` folder placed at the same level of the problem folder. The user can restart a previously stopped simulation using the `restart=1` option.* REMARK: The following data refer to the AGARD 445.6 benchmark test problem.

```
    solutionType     3;
    Nmodes           4;
    NactiveMode      1;
    kMax             10.0;
    epsU             0.017455;
    qMax             1.0;
    forcingStartTime 0.025; // [s]
    printInterval    4e-6;  // [s]
    restart          0;
```

Interface procedure. *The user can check if the interface procedure between the structural and aerodynamic meshes is correct using the option `testInterface=1` and plotting in* ParaView *the `cell T` field (aerodynamic cells belonging to different structural elements are filled with red and blue colours).*

```
    testInterface    0;
}
```

UTILITIES

For aeroservoelastic applications the boundary movement is only simulated by means of a transpiration velocity, without actually deforming the mesh runtime. The mesh can be deformed in the post-processing stage with the utility showDisplacement, using the data structures created by CAE2D and CAE3D toolboxes. For the mesh motion solver to work the following files are necessary:

- `cellDisplacement` and `pointDisplacement` fields in each time folder (those files are created runtime by CAE2D and CAE3D toolboxes with the exception of `cellDisplacement` field at the initial time. REMARK: The boundary conditions on the boundary subset to be deformed must be of type `fixedValue`, otherwise the displacement field will not be stored)

- `dynamicMeshDict` file in `constant` folder, with the mesh movement algorithm options, e. g.:

```
dynamicFvMesh dynamicMotionSolverFvMesh;
motionSolverLibs ("libfvMotionSolvers.so");
solver displacementLaplacian;
diffusivity  uniform;
```

- `fvSchemes` file in `system` folder with the following laplacian scheme, e. g.:

```
laplacianSchemes
{
    default none;
    laplacian(diffusivity,cellDisplacement) Gauss linear uncorrected;
}
```

- `fvSolution` file in `system` folder with the following mesh solver, e. g.:

```
cellDisplacement PCG
{
    preconditioner   DIC;
    tolerance        1e-08;
    relTol           0;
};
```

EXAMPLE 1: *Woodward and Colella supersonic benchmark*

To futher illustrate the main AeroFoam functionalities we solve the bidimensional unsteady supersonic flow of a polytropic ideal gas with $\gamma = 1.4$ and $\overline{R} = 0.714\ \frac{J}{kg\,K}$, with a freestream thermodynamic pressure $P_\infty = 1\ Pa$, temperature $T_\infty = 1\ K$ and Mach number $M_\infty = 3$ in the channel represented in Figure 1, $L = 3\ m$ long and $h = 1\ m$ high. At $x_\mathsf{S} = 0.6\ m$ the channel is characterized by the presence of a rectangular step $h_\mathsf{S} = 0.2\ m$ high.

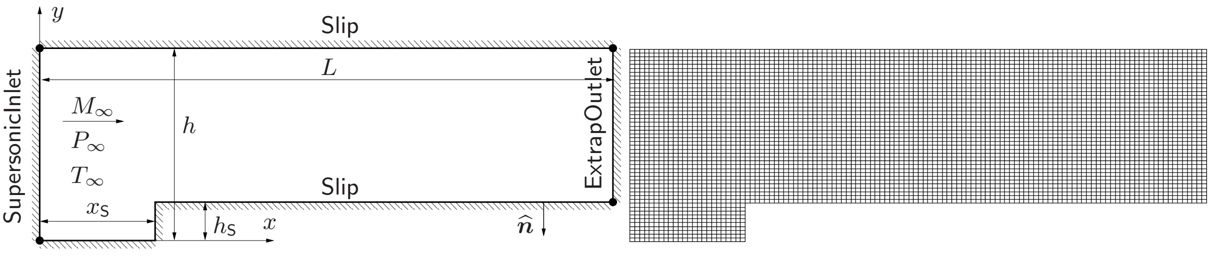


Figure 1: Qualitative definition of the Woodward and Colella supersonic benchmark test problem (left) and numerical grid created with `blockMesh` utility of $N_v = 5040$ cells with $\Delta x = 1/40\ m$ and $\Delta y = 1/50\ m$ (right).

Below the relevant sections of `controlDict` and `fvSchemes` input files are shown and the numerical solutions obtained with AeroFoam (left) at each `writeInterval` are compared with the corresponding reference numerical solutions of Woodward and Colella obtained with PPMLR on a much finer numerical grid of $N_v = 16128$ cells with $\Delta x = \Delta y = 1/80\ m$ (right).

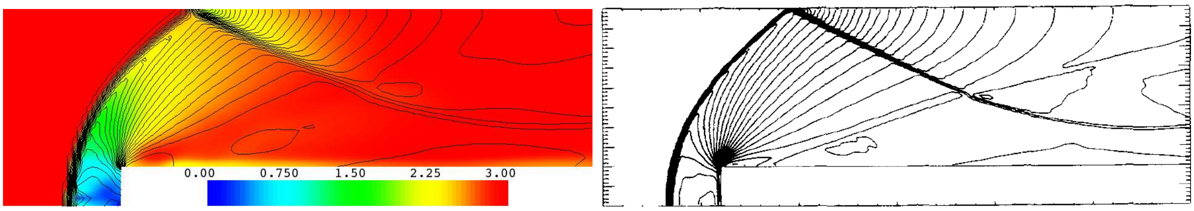
`controlDict`

```
    application          AeroFoam;
    startFrom            startTime;
    startTime            0;
    stopAt               endTime;
    endTime              4;
    deltaT               0.00125;
    writeControl         runTime;
    writeInterval        0.5;
```

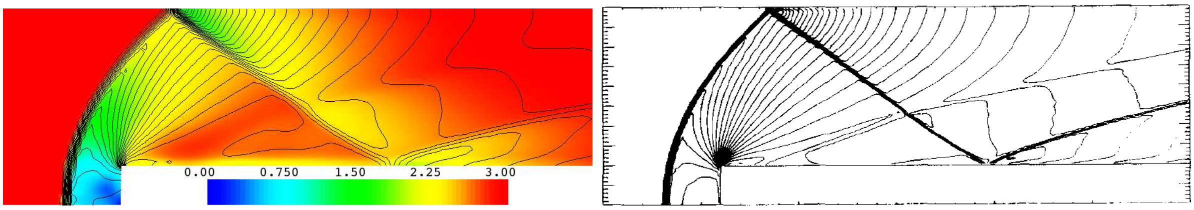
`fvSchemes`

```
AeroFoamSchemes
{
//.....................................Mathematical model
    flowType             Euler;
//.....................................Time integration
    timeScheme           RK2;
    CourantMax           0.0;
    CourantSteps         25;
//.....................................Space integration
    MonotoneFlux         Roe;
    HighResolutionFlux   LW;
    fluxLimiter          VL;
    entropyFix           HH2;
//.....................................Boundary conditions
    extrapolateBC        1;
//.....................................Residual
    residualNorm         L1;
    minResidual          -1;
//.....................................Extended cells connectivity
    loadConnectivity     0;
    adaptiveConnectivity 1;
    testMesh             0;
//.....................................Tests
    testInput            1;
}
```

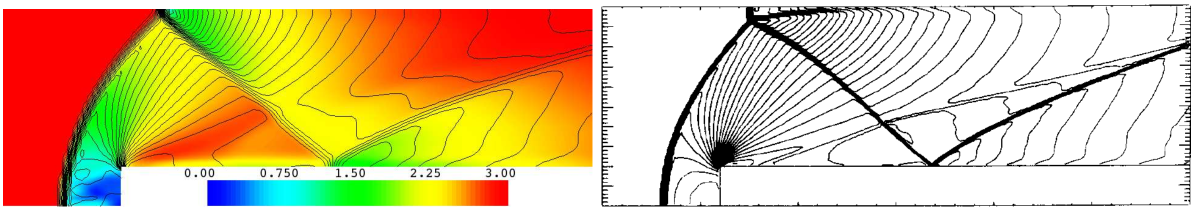
$t = 0.5\ s$



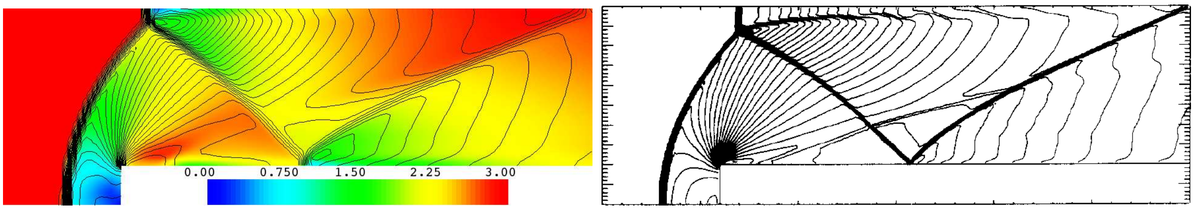
$t = 1.0\ s$



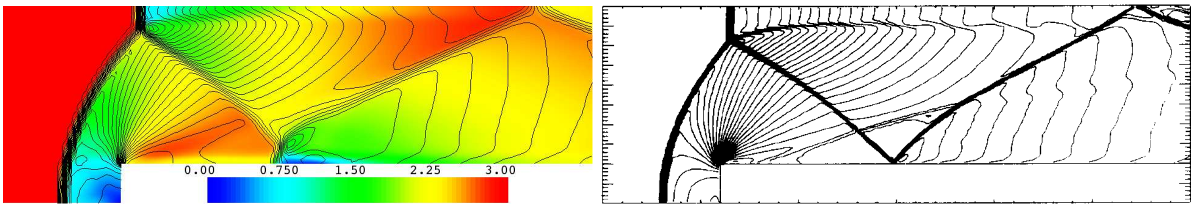
$t = 1.5\ s$



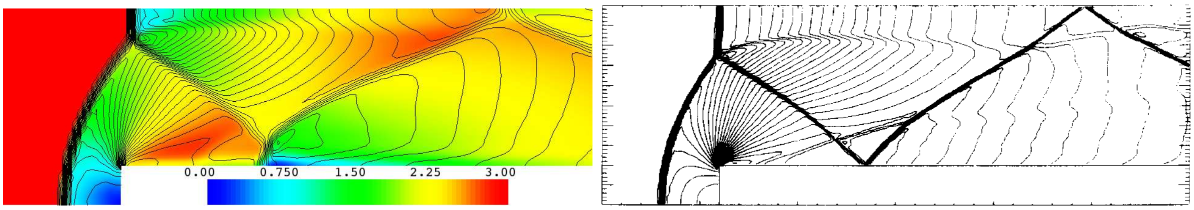
$t = 2.0\ s$



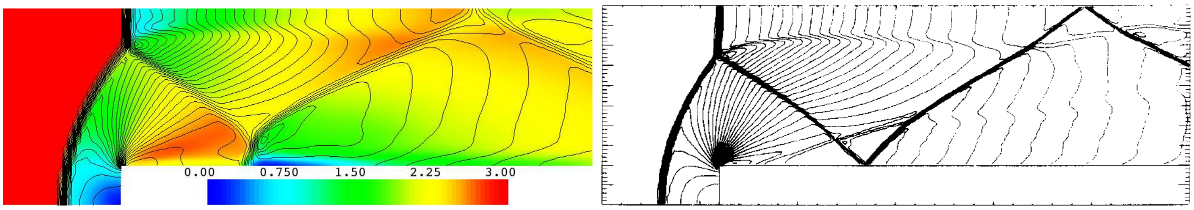
$t = 2.5\ s$



$t = 3.0\ s$



$t = 3.5\ s$



$t = 4.0\ s$

CONTACT AND REFERENCES

To report bugs or errors please contact `giulio.romanelli@gmail.com` or `elisa.serioli@gmail.com`. A lot of useful information and data about the mathematical models, the numerical schemes and the benchmark test problems can be found in the following references:

- R. L. Bisplinghoff, H. Ashley and R. L. Halfman. *Aeroelasticity.* Dover. 1996
- M. Feistauer, J. Felcman and I. Straškraba. *Mathematical and Computational Methods for Compressible Flow.* Oxford University Press. 2003
- A. Jameson. "Analysis and Design of Numerical Schemes for Gas Dynamics 1: Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence". *NASA Contractor Report 196477.* 1994
- A. Jameson. "Analysis and Design of Numerical Schemes for Gas Dynamics 2: Artificial Diffusion and Discrete Shock Structure". *NASA Contractor Report 196476.* 1994
- R. J. LeVeque. *Numerical Methods for Conservation Laws.* Birkäuser Verlag. 1992
- Various authors. "Experimental Data Base for Computer Program Assessment". *AGARD Advisory Report 138.* 1979
- P. Woodward and P. Colella. "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks". *Journal of Computational Physics.* 1984