ExaFLOW use case for SBLI: numerical simulation of the compressible flow over a NACA-4412 airfoil at incidence

N. De Tullio and N. D. Sandham
Aerodynamics and Flight Mechanics Group
University of Southampton
Southampton SO17 1BJ, UK

1 Introduction

The test case under consideration is the compressible, subsonic flow over a NACA-4412 airfoil at 5° incidence, for a freestream Mach number $M_\infty = 0.4$ and Reynolds number based on the airfoil chord of $Re_c = 50000$. The airfoil geometry, which includes a sharp (zero thickness) trailing edge, was obtained by modifying the last coefficient in the 4-digit NACA airfoil equation (see equation 6.2 in [1]) from -0.1015 to -0.1036. This modification leads to a sharp trailing edge, with minimal changes to the overall airfoil geometry. The two-dimensional (2D) base flow over the NACA-4412 airfoil was calculated using the 2D compressible Navier-Stokes equations. The flow is unsteady and includes vortex shedding from a laminar separation bubble that forms on the suction side of the airfoil, as can be seen in figure 1. The interaction between these vortices and the trailing edge of the airfoil causes the scattering of acoustic waves and leads to the acoustic field shown in figure 2 through contours of the dilatation rate $\nabla \cdot \mathbf{u}$.

In the numerical simulations described here, the computational domain is extruded in the spanwise direction and the three-dimensional (3D) Navier-Stokes equations are advanced in time starting from a 2D solution. The flow is assumed to be periodic in the spanwise direction. Since no external

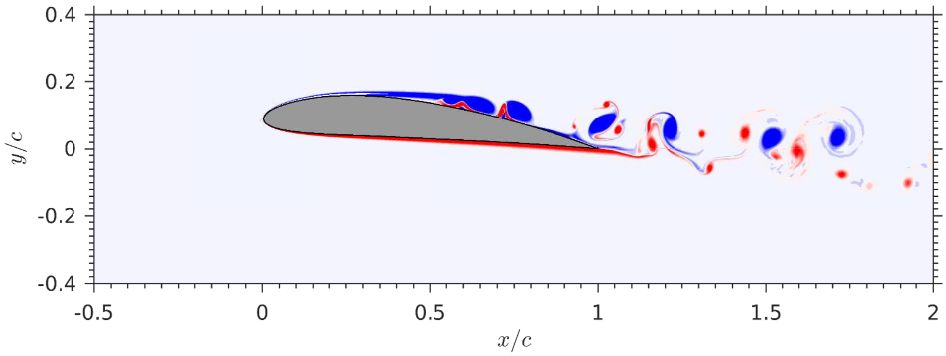

Figure 1: Instantaneous contours of spanwise vorticity ω_z. 50 contour levels plotted over a rage [-50, 50]. Blue for negative and red for positive.

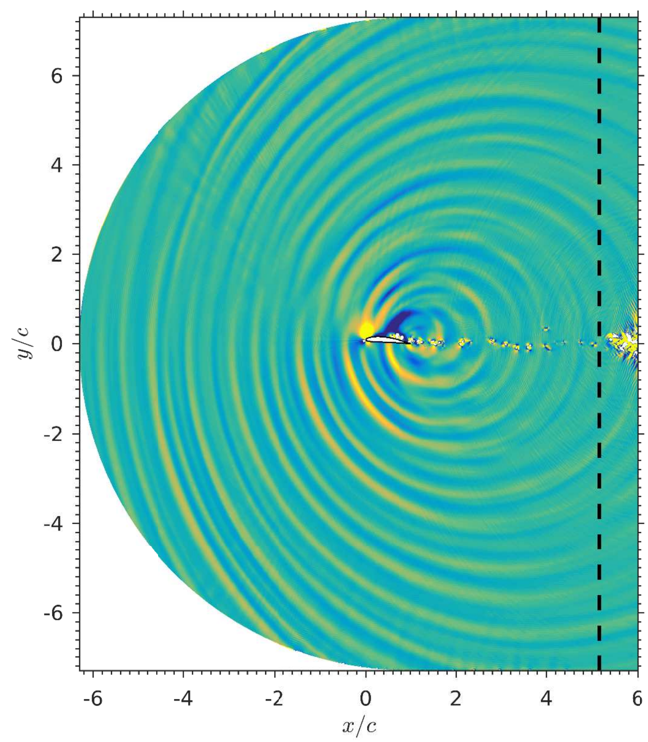

Figure 2: Instantaneous contours of dilatation rate. 50 contour levels plotted over a rage [-0.1, 0.1]. Blue for negative and yellow for positive. The dashed black line indicates the start of the zonal characteristic boundary condition.

3D disturbances are added to the simulations, the numerical solution should remain 2D.

2 Overview of numerical algorithm

The compressible Navier-Stokes equations are numerically solved using the SBLI code, which is a finite difference compressible Navier-Stokes solver developed at the University of Southampton. The spatial discretisation of the equations uses a standard fourth-order central difference scheme at internal points and a stable boundary treatment proposed by [2] close to boundaries, giving overall fourth-order accuracy. Time integration is based on a third-order compact Runge-Kutta method [10]. The code employs an entropy splitting approach developed by Sandham and co-workers [7], whereby the inviscid flux derivatives are split into conservative and non-conservative parts. The entropy splitting scheme, together with a Laplacian formulation of the heat transfer and viscous dissipation terms in the momentum and energy equations (which prevents the odd-even decoupling typical of central differences, see [7]), helps improve the stability of the low dissipative spatial discretisation used. The code has multi-block capabilities and is made parallel (both intra- and inter-block) using the message passing interface (MPI) library. The code has been validated extensively (see for example [5, 3, 4]).

3 Computational domain and grid arrangement

The computational domain is composed of three blocks, as can be seen in figure 3(a). Block 2 is a C-type structured grid fitted around the airfoil surface; it interfaces with the structured blocks 1 and 3, which resolve the wake of the airfoil. Since block 1 and block 3 both contain the wake line, the wake line solution at each time step is obtained by averaging between the solutions obtained in the two blocks. This is necessary because flow asymmetries near the airfoil trailing edge and/or small differences in initial conditions will cause the wake line solutions in the two blocks to diverge. For the current numerical simulations the computational domain dimensions are $W = 5.0c$ and $R = 7.3c$ (see figure 3(a)), where c is the chord length. The total domain length is $12.3c$ and the total height is $14.6c$

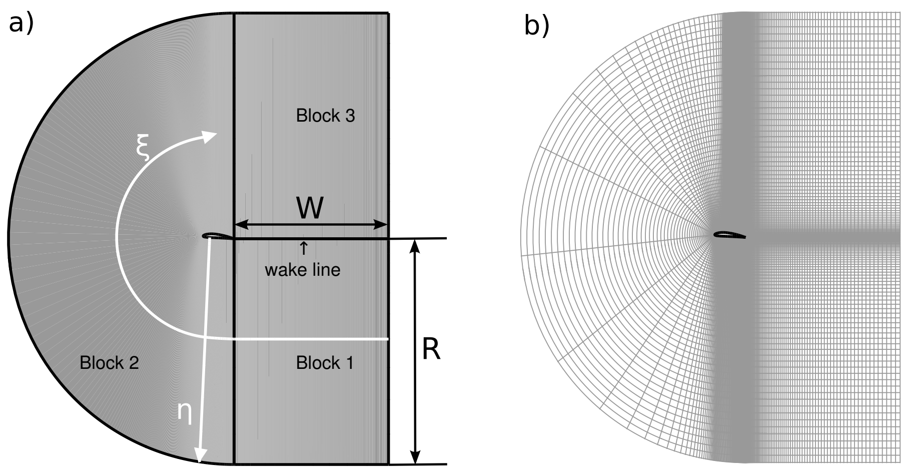

Figure 3: Computational domain arrangement. (a) multi-block domain set up, (b) computational grid, showing only one in every 10 grid points.

The numerical simulations will be carried out using characteristic conditions at all the computational domain boundaries, in order to minimize wave reflections. In particular, a zonal characteristic boundary condition [6] is applied over a distance $L_{zonal} \approx 0.85c$ near the outflow boundary of blocks 1 and 3, using 61 grid points. A standard characteristic condition [8, 9] is applied at the rest of the boundaries, where, in addition, the freestream solution is imposed at each time step. The airfoil is modelled using a no-slip, isothermal boundary condition, with the wall temperature equal to the freestream temperature.

	x	y	$\Delta\xi$	$\Delta\eta$
Stagnation point	0.004	0.086	1×10^{-3}	2.5×10^{-4}
Trailing edge	1.0	0.0	5.0×10^{-4}	2.5×10^{-4}
Exit boundary	6.0	0.0	1.5×10^{-2}	3.2×10^{-4}
Exit/Free stream boundary	6.0	7.3	1.5×10^{-2}	2.3×10^{-2}

Table 1: Grid resolution at key points in the domain.

A representation of the computational grid employed in the current numerical simulations is shown in figure 3(b), where only one in every ten grid points are plotted, while grid resolutions at key points in the domain, for the

Block	1	2	3
N_ξ	801	1799	801
N_η	692	692	692

Table 2: Number of grid points per block.

ξ and η directions, are given in table 3. The number of grid points employed per block for a 2D slice of the grid is given in table 3. The grid was designed to resolve all the flow features around the airfoil and the far field acoustic waves generated by the flow structures, for the current Reynolds and Mach numbers. This 2D grid is extruded in the z direction (wing span) using a constant grid spacing of $\Delta z = 0.002$. The number of grid points to be used along the span N_z is a user specified parameter that can be used to modify the size of the numerical simulation. Since we impose Δz to be a constant, changing N_z means changing the spanwise extension of the computational domain.

4 Input data

The input data needed to start a simulation are: an input file (`Input.in`) with flow and numerical parameters, a binar grid file (`Airfoil_3D.bin`) that contains the computational grid coordinates and additional information regarding the multi-block domain layout, boundary conditions per block, processor distribution per block, etc. and one restart file per block (`Qfile_r1`, `Qfile_r2` and `Qfile_r3`) with flowfield and boundary data.

An excerpt of the file `Input.in` is shown below:

```
#Mach, Reynolds, Prandtl, Schmidt, Gamma, Omega
0.4, 50000, 0.72, 1.0, 1.4, 0.76
#Sutherland: temp. const. (K), ref. temp. (K)
110.4, 273.15
#CFL, dt, use CFL to calc dt ?
2.0, 0.00010, .f.
#Time step number, plot3d output step, max time
1000,1000,5000.0
#Input file, Binary input grid?, Fortran 77 input grid?
'Airfoil_3D.bin',.t.,.f.
```

```
#Restart?, Restart input file directory
.t.,'.'
#Output?, Output dir., Qfile output step (-1: at the end)
.f.,'RESULTS/',-1
#Num. of monitor points, then i,j,k, & block num. for each
5 1371 10 1 2 1524 10 1 2 1 10 1 3 191 10 1 3 191 10 4 3
```

It can be seen that the simulation is set up to run 1000 time steps. The values of the conservative variables (ρ, ρu, ρv, ρw and ρE) at five monitor points (see line 8) will be recorded every 100 time steps and saved in five `monitor_point.i` (`i` from 1 to 5) files at the end of the simulation. No additional output will be saved, unless the user modifies the first element in line 7 of the file `Input.in` to read `.t.`. The output time step period can be specified in line 4 for single precision `plot3d` files and in line 7 for double precision `Qfile` files. The `Input.in` file is located in the folder `../Test_Case/Execute/` from which the code executable `pdns3d.x` should be run. The executable can be generated using the `Makefile` in `../Test_Case/Code/`.

The file `Airfoil_3D.bin` is generated by the `BuildGridBinary.f` Fortran routine, which needs as inputs the number of grid points in the spanwise direction and the number of processors in all the three directions per block. The `BuildGridBinary.f` routine accesses the above information through the input file `Grinput.in`. It is important to note that block 2 contains about 2.25 times the number of grid points in blocks 1 and 3, meaning that block 2 should have about 2.25 times the number of processors assigned to block 1 and block 3, for load balancing purposes (blocks are not allowed to share processors). In addition, the zonal characteristic boundary condition applied near the outflow of blocks 1 and 3 (across 61 grid points) cannot cross processor boundaries, hence the maximum number of processors in the ξ direction for this two blocks is 12.

Files `Qfile_r1`, `Qfile_r2` and `Qfile_r3` are generated by the Fortran routine `Extrude_2DField.f`, which also uses the `Grinput.in` file for input. The Fortran routines `BuildGridBinary.f` and `Extrude_2DField.f` are located in `../Test_Case/Grid/`, together with all the files needed for the generation of the computational grid. Once generated, the binary files `Airfoil_3D.bin`, `Qfile_r1`, `Qfile_r2` and `Qfile_r3` should be moved to the folder `../Test_Case/Execute/`.

5 Standard output

The set up of the current simulation is such that the SBLI code will only give five `monitor_point.i` (`i` from 1 to 5) files as output. These files will be generated in `../Test_Case/Execute/` at the end of the simulation; they contain the values of the conservative variables every 100 time steps at five different points in the domain. As an example, the contents of the file `monitor_point.1` are shown in table 5. Note that point number 5 has the same x- and y-coordinates as point 4, but different z-coordinate. However, since the flow should remain 2D, these two files should be identical. The monitor point files obtained for a case with 5 grid points in the spanwise direction can be found in the folder `../Test_Case/Execute/Monitoring_Points/`

References

[1] I. Abbot and A. von Doenhoff. *Theory of wing sections.* Dover Publications, 2^{nd} edition, 1959.

[2] M. H. Carpenter, J. Nordstrom, and D. Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148:341–365, 1999.

[3] N. De Tullio. *Receptivity and transition to turbulence of supersonic boundary layers with surface roughness .* PhD thesis, School of Engineering Sciences, University of Southampton, 2013.

[4] N. De Tullio, P. Paredes, N. D. Sandham, and V. Theofilis. Laminar-turbulent transition induced by a discrete roughness element in a supersonic boundary layer. *J. Fluid Mech.*, 735:613–646, 2013.

[5] N. De Tullio and N. D. Sandham. Direct numerical simulation of breakdown to turbulence in a Mach 6 boundary layer over a porous surface. *Physics of Fluids*, 22(094105), 2010.

[6] R. Sandberg and N. Sandham. Nonreflecting zonal characteristic boundary condition for direct numerical simulation of aerodynamic sound. *AIAA J.*, 44(2):402–405, 2006.

[7] N. D. Sandham, Q. Li, and H. C. Yee. Entropy splitting for high-order numerical simulation of compressible turbulence. *Journal of Computational Physics*, 178:307–322, 2002.

[8] K. W. Thomson. Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, 68:1–24, 1987.

[9] K. W. Thomson. Time dependent boundary conditions for hyperbolic systems, II. *Journal of Computational Physics*, 89:439–461, 1990.

[10] A. A. Wray. Minimal storage time advancement schemes for spectral methods. Rept. M.S. 202 A-1., NASA Ames Research Centre, 1990.

Step	time	ρ	ρu	ρv	ρw	ρE
1000000	0.10000000E+03	0.89329195E+00	0.22677297E+00	-0.64542249E-01	00000000E+00	0.99486173E+01
1000100	0.10001000E+03	0.89123269E+00	0.21704640E+00	-0.50826718E-01	00000000E+00	0.99261284E+01
1000200	0.10002000E+03	0.88944252E+00	0.22974307E+00	-0.33023594E-01	00000000E+00	0.99169495E+01
1000300	0.10003000E+03	0.88862728E+00	0.27104728E+00	-0.22318973E-01	00000000E+00	0.99301942E+01
1000400	0.10004000E+03	0.88904274E+00	0.34514868E+00	-0.19861198E-01	00000000E+00	0.99662313E+01
1000500	0.10005000E+03	0.89085059E+00	0.44563483E+00	-0.20426127E-01	00000000E+00	0.10025899E+02
1000600	0.10006000E+03	0.89368558E+00	0.51468416E+00	-0.35578044E-01	00000000E+00	0.10085324E+02
1000700	0.10007000E+03	0.89643240E+00	0.48810331E+00	-0.70115296E-01	00000000E+00	0.10107626E+02
1000800	0.10008000E+03	0.89978262E+00	0.39417874E+00	-0.10434086E+00	00000000E+00	0.10120412E+02
1000900	0.10009000E+03	0.90413431E+00	0.26230020E+00	-0.12572211E+00	00000000E+00	0.10142470E+02
1001000	0.10010000E+03	0.90821967E+00	0.97483446E-01	-0.13070930E+00	00000000E+00	0.10161082E+02

Table 3: Contents of the `monitor_point.1` file.