Numerical study of the airflow over a high-altitude pseudo-satellite wing

PhD update

Carlo Brunelli

October 30, 2023

PhD presentation after 1.5 years

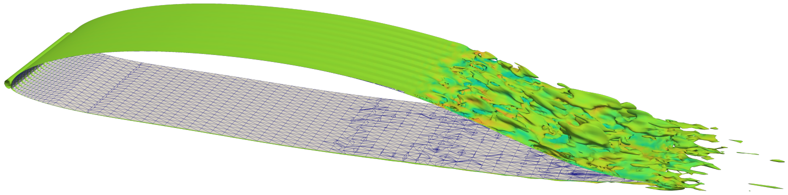


Figure: DU89 velocity contour, u_z colormap

Sections

Numerical Method implemented

The numerical method currently implemented are:

- Variational Multiscale Method (VMS)
- Streamline upwind Petrov–Galerkin (SUPG)

Different solution method are available for all of them

- Non-linear (NLIN)
- Linearized-Coupled (LC-VMS)
- Linearized-Segregated (LS-VMS)

Variational Multiscale Method

- Evolution of the SUPG
- Implict LES
- It does not need calibration
- Residual-based stabilization

Galkerkin formulation

Conservation of mass:

$$\nabla \cdot \vec{u} = 0 \tag{1}$$

Conservation of momentum:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} + \nabla p - \nu \Delta \vec{u} - f = 0 \tag{2}$$

Variational Formulation

$$\begin{aligned} B^G = \int_\Omega \frac{\partial \vec{u}}{\partial t} \cdot \vec{v}\, d\Omega + \int_\Omega (\vec{u} \cdot \nabla)\vec{u} \cdot \vec{v}\, d\Omega + \int_\Omega \nabla(p) \cdot \vec{v}\, d\Omega + \\ + \int_\Omega \nu \nabla \vec{u} \cdot \nabla \vec{v}\, d\Omega - \int_\Omega f \cdot \vec{v}\, d\Omega + \int_\Omega q(\nabla \cdot \vec{u})\, d\Omega = 0 \end{aligned} \tag{3}$$

Stabilization equations

$$B^{SUPG}(t,(\vec{u},p),(\vec{v},q)) = \int_\Omega (\tau_m(\vec{u}\cdot\nabla\vec{v} + \nabla q)\cdot\vec{R_m}\, d\Omega + \int_\Omega \tau_c(\nabla\cdot\vec{v})R_c\, d\Omega \tag{4}$$

$$B^{VMS1}(t,(\vec{u},p),(\vec{v},q)) = \int_\Omega (\vec{u}\cdot\nabla\vec{v}') \odot (\tau_m\vec{R_m})\, d\Omega \tag{5}$$

$$B^{VMS2}(t,(\vec{u},p),(\vec{v},q)) = -\int_\Omega (\nabla\vec{v} \odot (\tau_m\vec{R_m} \otimes \tau_m\vec{R_m})\, d\Omega \tag{6}$$

Stabilization parameters

$$\tau_m = \left(\frac{4}{\Delta t^2} + \vec{u} \cdot G\vec{u} + C_I \nu^2 G : G \right)^{-1/2} \tag{7}$$

$$\tau_c = (\tau_c \vec{g} \cdot \vec{g})^{-1} \tag{8}$$

Where G is the inverse of the gradient of the map cell. For a cubed shaped element, with h the edge length, $G_{ij} = \frac{1}{h^2}\delta_{ij}$, where δ_{ij} is the Kronecker delta.

Linearization

Using Taylor's expansion for velocity:

$$\vec{u}_{adv} = 2.1875u^n - 2.1875u^{n-1} + 1.3125u^{n-2} - 0.3125u^{n-3} \tag{9}$$

$$(\vec{u} \cdot \nabla)\vec{u} \Rightarrow (\vec{u}_{adv} \cdot \nabla)\vec{u} \tag{10}$$

$\vec{u}_{adv}$ used also for computing stabilization parameters

Linearization

$$B^G(t,(\vec{u},p),(\vec{v},q)) = \int_\Omega \frac{\partial \vec{u}}{\partial t}\cdot\vec{v}\,d\Omega + \int_\Omega (\vec{u}_{adv}\cdot\nabla)\vec{u}\cdot\vec{v}\,d\Omega + \int_\Omega \nabla(p)\cdot\vec{v}\,d\Omega + \\ + \int_\Omega \nu\nabla\vec{u}\cdot\nabla\vec{v}\,d\Omega - \int_\Omega f\cdot\vec{v}\,d\Omega + \int_\Omega q(\nabla\cdot\vec{u})\,d\Omega \tag{11}$$

$$B^{SUPG,lin}(t,(\vec{u},p),(\vec{v},q)) = \int_\Omega (\tau_m(\vec{u}_{adv}\cdot\nabla\vec{v} + \nabla q)\cdot \vec{R}_{m,lin}\,d\Omega + \int_\Omega \tau_c(\nabla\cdot\vec{v})R_c\,d\Omega \tag{12}$$

$$B^{VMS1,lin}(t,(\vec{u},p),(\vec{v},q)) = \int_\Omega (\vec{u}_{adv}\cdot\nabla\vec{v}') \odot (\tau_m \vec{R}_{m,lin})\,d\Omega \tag{13}$$

ODE Solver

1 Numerical Methods

Employing θ-method for time marching. $\theta = 0.5$ for velocity and $\theta = 1.0$ for pressure to avoid un-smoothed oscillations

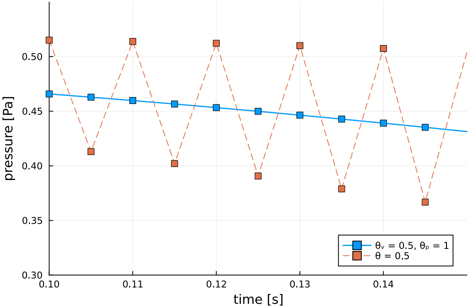


Figure: Pressure oscillations

Segregated formulation

$$T\frac{\partial \vec{x}}{\partial t} + A\vec{x} = 0 \tag{14}$$

Applying the θ-method

$$T\frac{\partial \vec{x}}{\partial t} = -A\vec{x} \tag{15}$$

$$T\frac{\vec{x}^{n+1} - \vec{x}^{n}}{\Delta t} = -\left(A\vec{x}^{n+1}\theta + A\vec{x}^{n}(1-\theta)\right) \tag{16}$$

$$\left(\frac{T}{\Delta t} + A\theta\right)(\vec{x}^{n+1} - \vec{x}^{n}) = -A\vec{x}^{n} \tag{17}$$

Matrices splitting

Splitting the problem in solving velocity and pressure field:

$$A = \begin{bmatrix} A_{pp} & A_{pu} \\ A_{up} & A_{uu} \end{bmatrix} \tag{18}$$

$$T = \begin{bmatrix} 0 & T_{pu} \\ 0 & T_{uu} \end{bmatrix} \tag{19}$$

Thanks to the stabilization the matrix A_{pp} is nonzero. This translates into nonzero diagonal elements, reducing the conditioning number of the matrix and improving the overall stability of the method.

Introducing the time step increments and the acceleration terms:

$$\Delta u = u^{n+1} - u^n$$

$$\Delta p = p^{n+1} - p^n$$

$$a = \Delta u / \Delta t$$

The linear system will be solved in an iterative manner. Two following iterations are marked as m and $m+1$, and the difference between two following iteration is:

$$\Delta a = a^{m+1} - a^m$$

$$\Delta p^{m+1} = a^{m+1} - a^m$$

$$u^m = u^n + \Delta t a^m$$

$$p^m = p^n + \sum_{i=0}^{m} \Delta p^i$$

$$
\begin{aligned}
(T_{uu} + \theta \Delta t A_{uu}) \boldsymbol{\Delta a^*} = -A_{uu} u^m - A u_{up} p^m - (T_{uu} + \theta \Delta t A_{uu}) a^m + A_{uu} \Delta t a^m + \\
+ A_{uu} \Delta t a^m + (1-\theta) A_{up} \sum_{i=0}^{m} \Delta p^i
\end{aligned}
\tag{20}
$$

$$
\begin{aligned}
\left((T_{pu} + \Delta t A_{pu})(T_{uu} + \theta \Delta t A_{uu})^{-1} \theta A_{up} - App \right) \boldsymbol{\Delta p^{m+1}} = \\
= T_{pu} \Delta a^* + A_{pu}(u^m + \Delta t \Delta a^*) + A_{pp} p^m + T_{pu} a^m
\end{aligned}
\tag{21}
$$

It is possible to apply a multicorrector-predictor iterative scheme. At the beginning of each time step the velocity, pressure, and acceleration are initialized as:

- $u^0 = u^n$
- $p^0 = p^n$
- $a^0 = 0$
- $\sum_{i=0}^{m} \Delta p^i = 0$

Then the iterative process begins:

1. Solve linear system (20) for Δa^*
2. Solve linear system (21) for Δp^{m+1}
3. Compute $\Delta a = \Delta a^* - (T_{uu} + \theta \Delta t A_{uu})^{-1} \theta A_{up} \Delta p^{m+1}$
4. Update $a^{m+1} = a^m + \Delta a$
5. Update $u^{m+1} = u^m + \Delta a \Delta t$
6. Update $p^{m+1} = p^m + \Delta p^{m+1}$

Royal Military Academy

Sections

2 Performances

Weak Scaling

Weak scalability: the solution time almost does not change with constant problem size per processor

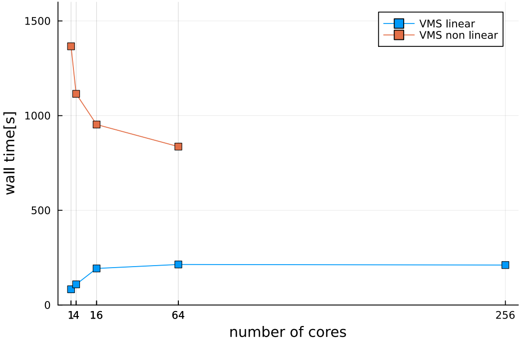


Figure: Taylor Green weak scalability

Strong Scaling

Strong scalability: doubling the number of processors halves the solution time

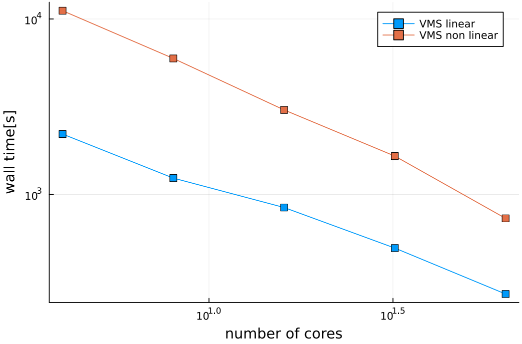


Figure: Taylor Green Strong scalability

Boundary Layer Initialization

- Avoid Instabilities (close to the leading edge)
- Avoid velocity ramping
- Allows Higher time-step

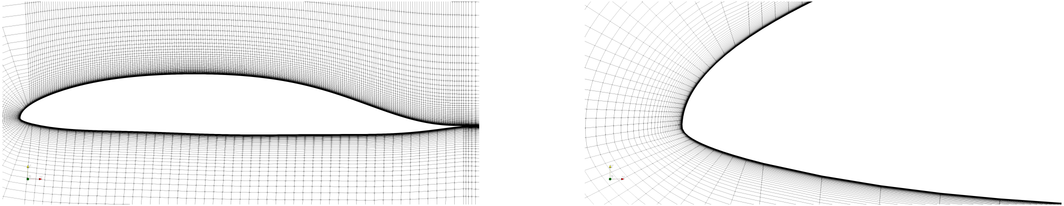


Figure: Mesh DU89

Find wall-distance

Using the function exploited by RANS solvers for the wall distance for computing turbulent parameters but now used to detect the airfoil's contour

$$\begin{cases} \nabla \cdot (|\nabla u_p|^{p-2} \nabla u_p) = -1 & x \in \Omega \\ u_p = 0 & x \in \Omega_D \end{cases} \tag{22}$$

The true-wall distance is provided by solving the equation for $p \to \infty$ Solved using a method that resembles Picard's

Find wall-distance

$$v_p = -|\nabla u_p|^{p-1} + \left(\frac{p}{p-1}u_p + |\nabla u_p|^p\right)^{\dfrac{p}{p-1}} \tag{23}$$

1. It starts solving the equation for p=2
2. $$\int \Big(\nabla(v) \cdot (\nabla u_{hp})\Big) d\Omega = \int \Big(1 \cdot v\Big) d\Omega \tag{24}$$
3. It solves the linearized equation, which becomes (25). In this case solving for p=3 the $\tilde{u}$ is uh_2
$$\int \Big(\nabla(v) \cdot (|\nabla \tilde{u}|^{p-2} \nabla u)\Big) d\Omega = \int \Big(1 \cdot v\Big) d\Omega \tag{25}$$
4. It is not possible to iterate step 3 for p > 3, the solution becomes unstable. It introduces the relaxation coefficient $\gamma = 0.5$ and it starts an internal cycle.

Boundary Layer Initialization

The velocity in the x direction in the region identified can be with a simple cubic function (26) where $dn = d/\delta_{99}$, d is the minimum distance to the airfoil, u_∞ is the free-stream flow speed.

$$f(dn) = \begin{cases} u_\infty & dn > 1 \\ (-dn^2 + 2 \cdot dn) \cdot u_\infty & dn < 1 \end{cases} \tag{26}$$

The boundary layer function has been obtained by fixing the following boundary conditions:

- Continuity with the external flow, $f(1) = 1$
- Smooth transition between boundary layer and external flow, $f'(1) = 0$
- Non-slip condition at the wall, $f(0) = 0$

Boundary Layer Initialization

It results in a low-speed zone close to the airfoil, avoiding high speed in really small cells useful for capturing the boundary layer.

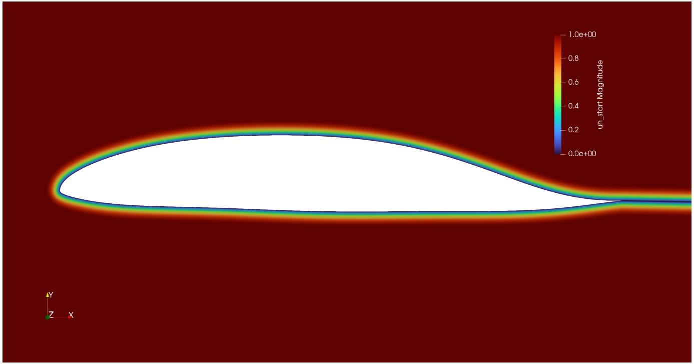


Figure: Boundary Layer Initialization

Sections

4 Synthetic Eddy Method

SyntheticEddyMethod.jl

- publication
- presented at *JuliaCon2023* at MIT

Features:

- Create fluctuations that respect the divergence-free condition (DFSEM)
- Create velocity fluctuations for inlet boundary conditions
- Create coherent eddies in 3D domain
- Define custom Reynolds Stress Tensor
- Import from file custom Reynolds Stress Tensor

Synthetic Eddy Method

Reynolds decomposition:

$$\vec{u}(\vec{x},t) = \vec{U}(\vec{x},t) + \vec{u'}(\vec{x},t) \tag{27}$$

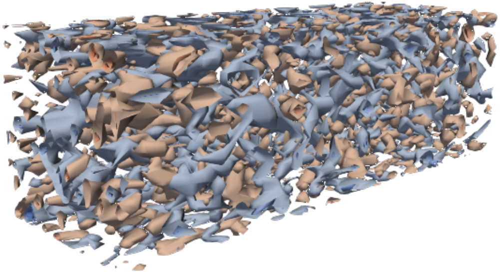
Compute velocity fluctuations, using a suitable shape function:

$$u_i(\boldsymbol{x}) = U_i(\boldsymbol{x}) + \frac{1}{\sqrt{N}} \sum_{k=1}^{N} a_{ij} \epsilon_j^k f_{\sigma(\boldsymbol{x})} \left(\boldsymbol{x} - \boldsymbol{x}^k\right) \tag{28}$$

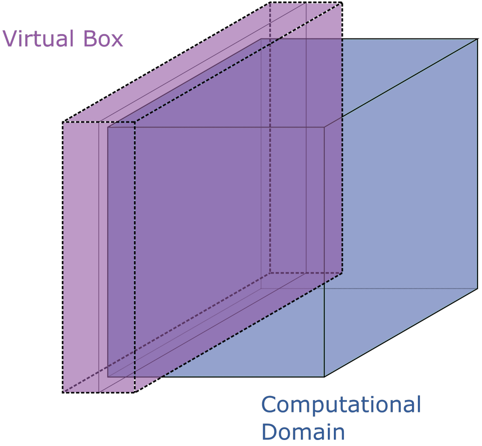
where is the shape function (tent, step or Gaussian)

$$f = \begin{cases} \sqrt{\frac{3}{2}}(1 - |x|) & |x| \leq 1 \\ 0 & |x| > 1 \end{cases} \tag{29}$$

Synthetic Eddy Method



(a) Isovelocity contour



(b) Virtual Box

Synthetic Eddy Method

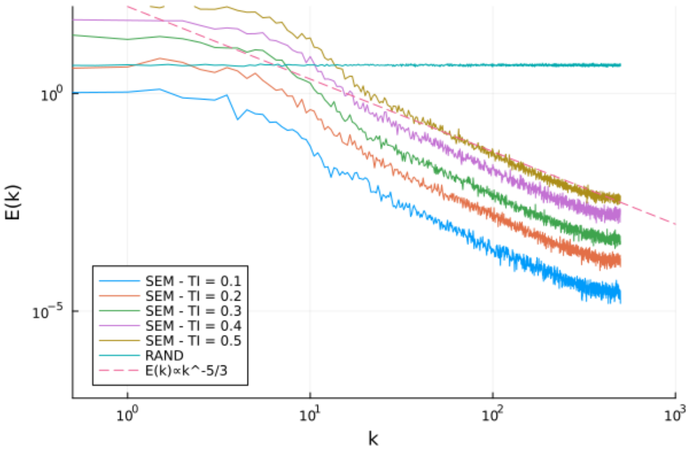


Figure: Power Spectral Density of the turbulent kinetic energy obtained using tent function

Royal Military Academy

Sections

Simulations

5 Laminar Separation Bubble

- VMS linearized coupled sd7003s - Re $60\,000$ - AoA 4°
- VMS linearized segregated DU89 - Re $250\,000 - 500\,000$ - AoA $1^\circ - 5^\circ$

Models

sd7003s - Re 60 000 - AoA 4°

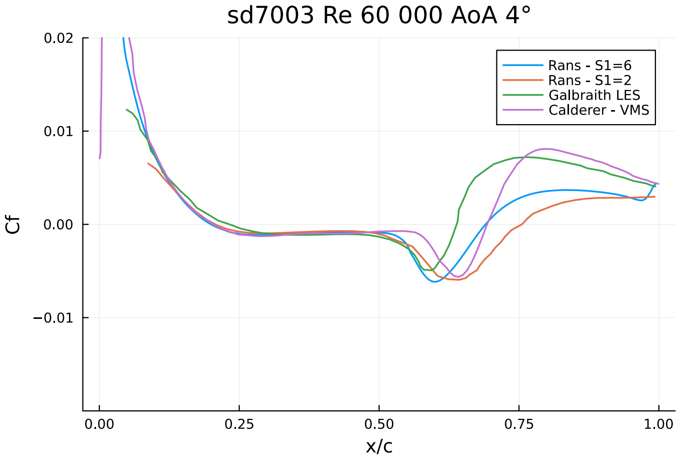
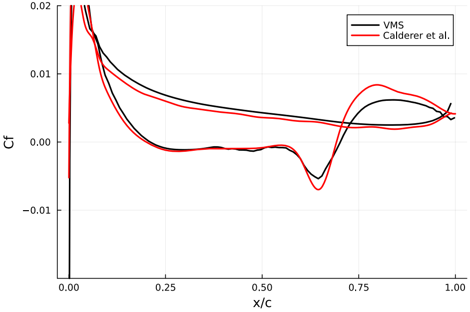


Figure: Different model provides different results. RANs use $\gamma - Re_\theta$ model.

VMS linearized coupled sd7003s

Copuled: velocity and pressure are solved at the same time. Re 60 000 - AoA 4° .
Initialization with velocity-ramping



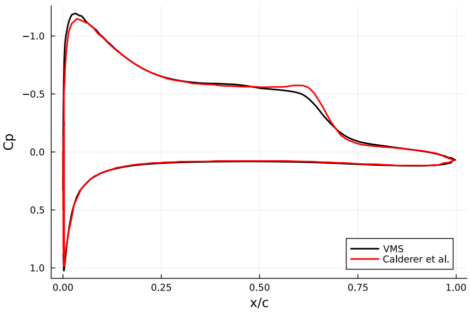
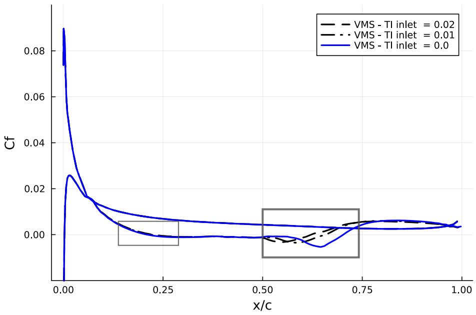
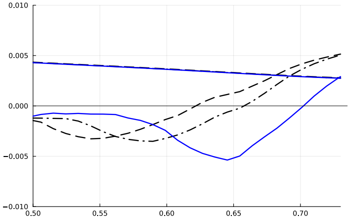


Figure: Comparison with VMS literature results

VMS linearized coupled sd7003s





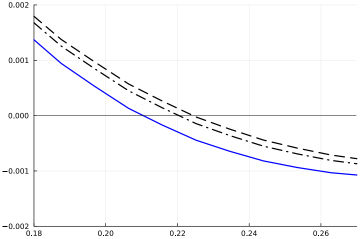
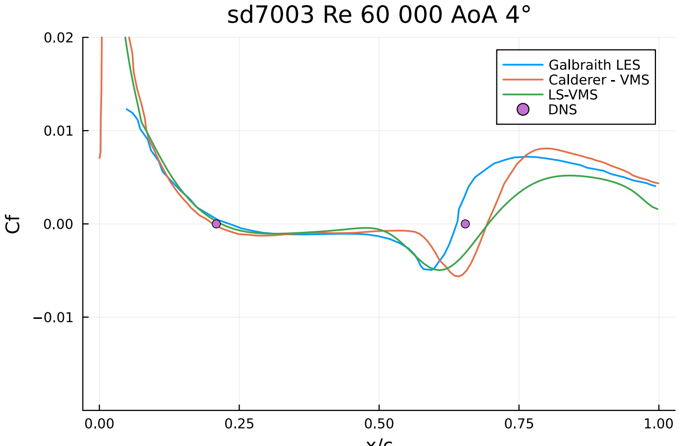


Figure: Bubble position function of freestream turbulence intensity

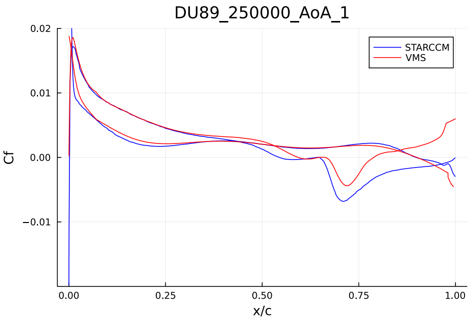
Segregated: each time step pressure and velocity system are solved one after the other multiple times. It is possible to re-use the matrices and preconditioner. It is an iterative method.

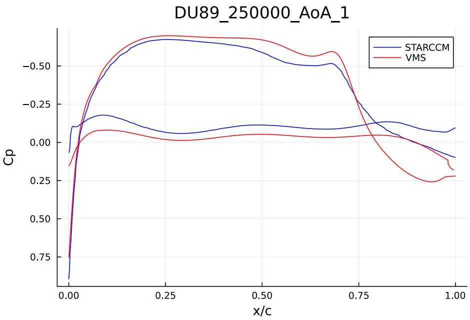


VMS Linearized-Segregated

5 Laminar Separation Bubble

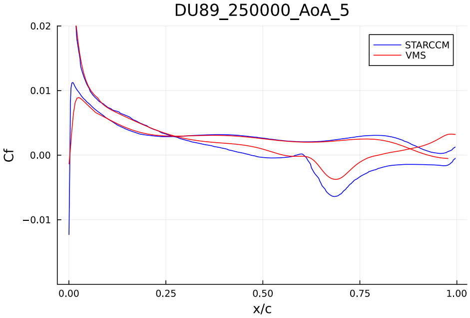
PhD research aims to simulate a new airfoil. Re 250 000 - AoA 1°

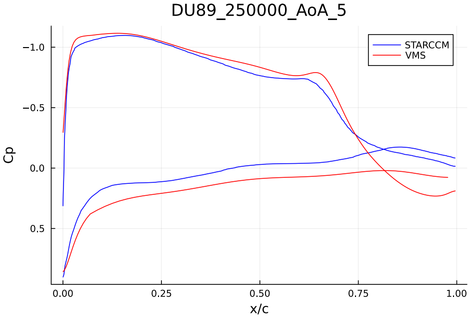




VMS Linearized-Segregated

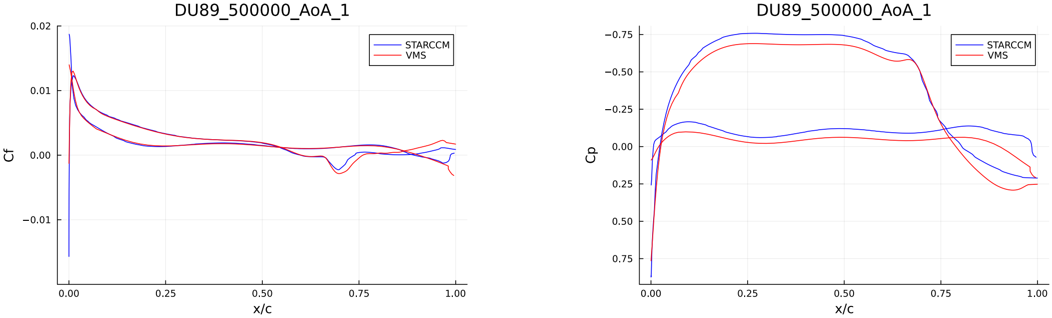
Re 250 000 - AoA 5°





VMS Linearized-Segregated

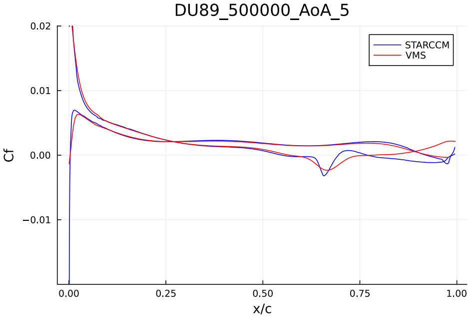
Re 500 000 - AoA 1°

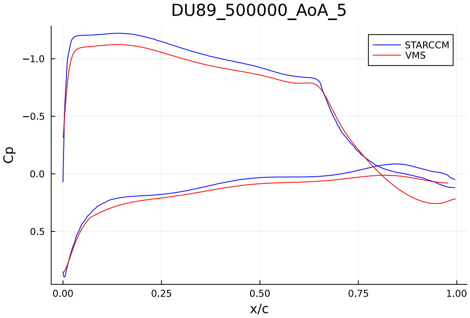


VMS Linearized-Segregated

5 Laminar Separation Bubble

Re 500 000 - AoA 5°





Mesh Sensitivity

Mesh settings	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{F}$	$\mathcal{SF}$
Airfoil divisions	150	200	200	300
Z divisions	12	16	22	16
First cell height [m]	4.8e-6	2.8e-6	1.6e-6	1.6e-6
Number of Cells	4.1e5	6.6e5	8.5e6	8.9e6
CL	0.3539	0.3514	0.3504	0.3519
CD	0.00915	0.00950	0.00929	0.00947
Separation (x/c)	0.60	0.60	0.60	0.60

Table: Mesh sensitivity analysis DU89, Reynolds 500 000, Aoa 1°

Time Sensitivity

Mesh	$\mathcal{C}$	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{M}$
Time average[s]	10	20	10	20
CL	0.3538	0.3539	0.3516	0.3514
CD	0.00910	0.00915	0.00939	0.00950
Separation (x/c)	0.60	0.60	0.60	0.60

Table: Time average sensitivity analysis DU89, Reynolds 500 000, Aoa 1°

dt$[s]$	$2\ 10^{-3}$	$1\ 10^{-3}$	$5\ 10^{-4}$
CL	diverged	0.3514	0.3511
CD	diverged	0.00950	0.00911
Separation (x/c)	diverged	0.60	0.60

Table: Time sensitivity analysis DU89, Reynolds 500 000, Aoa 1°

VMS airfoil in wind tunnel

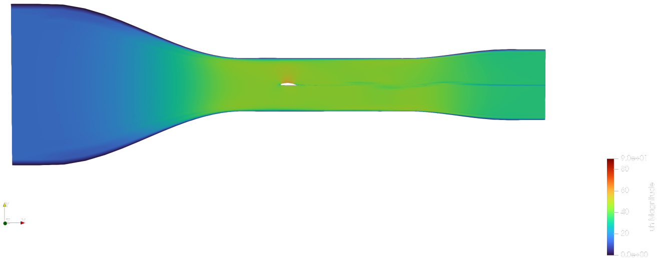


Figure: 3D VMS Wind Tunnel simulation with DU89

Royal Military Academy

Sections

6 Uncertainty Quantification

Model Variables

RANS Re 60 000 - AoA 4° sd7003.

- TI $(\mu = 5.0 \cdot 10^{-4}; \sigma = 2 \cdot 10^{-4})$- turbulence intensity
- $\mu_r \in [1, 10]$ - turbulent viscosity ratio

$k - \omega$ parameters:

- $\sigma\omega 1$ $(\mu = 0.5; \sigma = 0.0667)$
- $\alpha 1$ $(\mu = 0.31; \sigma = 0.03)$
- $\beta*$ $(\mu = 0.09; \sigma = 0.0041)$

γRe_{θ} parameters

- $s1 \in [2, 11]$
- $C1 \in [2, 4]$

Sobol Indexes

6 Uncertainty Quantification

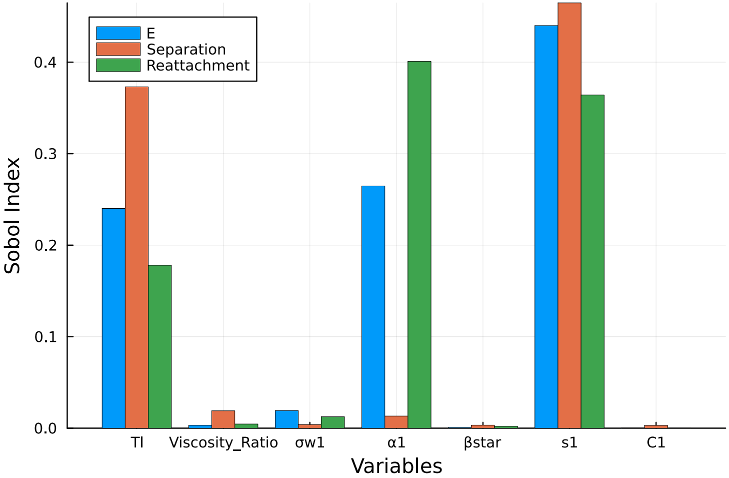


Figure: Sobol' first order indices for 3D RANS variables

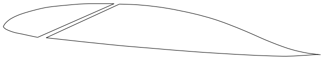
Royal Military Academy

Sections

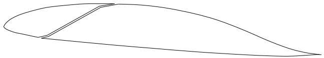
7 Conclusions

Other activities

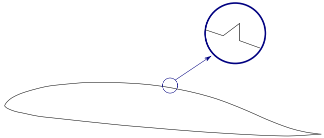
- Synthetic Eddy Method publication
- JuliaCon 2023 conference
- Co-author AIAA 2023 conference paper
- Testing standard passive flow controls (no improvement in aerodynamic efficiency)



(a) Slot



(b) Slot curved



(c) Riblet

Figure: Passive Flow Controls

Elements of novelty

- Systematic usage of Julia in fluid-dynamics
- Usage of VMS for high Reynolds airfoil
- First LES code in Julia - fully parallelized - working with 3D airfoils up to Re 500 000
- Synthetic Eddy Method coded in Julia and coupled with the VMS

It has been challenging, but it seems we are on the right path!

Next steps

Expected papers:

- VMS paper (prof. Janssens reading)
- Experimental validation of the LS-VMS - publish paper
- LC-VMS - (publish paper?)

Expected conferences:

- AIAA2024 conferences (LS-VMS, LC-VMS)
- DLES14 (VMS usage, validation test cases)
- ICAS2024 (Passive Flow Controls)

Expected research:

- Uncertainty Quantification using Polynomials Chaos Transformation on $\gamma - Re_\theta$ and VMS
- Start coding the adjoint optimization
- Test a passive flow control with the VMS