

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

PhD update

Carlo Brunelli

October 31, 2023



von KARMAN INSTITUTE
FOR FLUID DYNAMICS

PhD presentation after 1.5 years

- *Is it possible to implement efficiently the Variational Multiscale Method in the Julia programming language?*
- *Is it suitable to study transitional airfoils at low-Reynolds?*

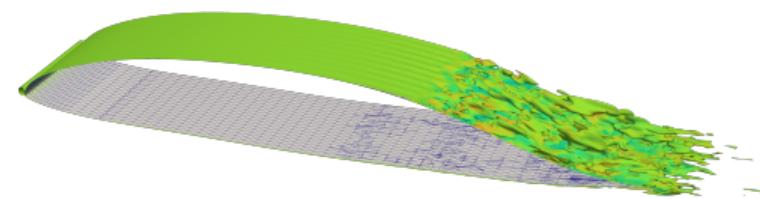


Figure: DU89 velocity contour, u_z colormap

Sections

1 Numerical Methods

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
- ▶ Synthetic Eddy Method
- ▶ Laminar Separation Bubble
- ▶ Uncertainty Quantification
- ▶ Conclusions

Numerical Method implemented

1 Numerical Methods

The numerical methods currently implemented are:

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

Different solution methods are available for all of them

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

Variational Multiscale Method

1 Numerical Methods

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

Galerkin formulation

1 Numerical Methods

Conservation of mass:

$$\nabla \cdot \vec{u} = 0 \quad (1)$$

Conservation of momentum:

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

Variational Formulation

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

Stabilization equations

1 Numerical Methods

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

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

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

Stabilization parameters

1 Numerical Methods

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

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

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

1 Numerical Methods

Using Taylor's expansion for velocity:

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

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

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

Linearization

1 Numerical Methods

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

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

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

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

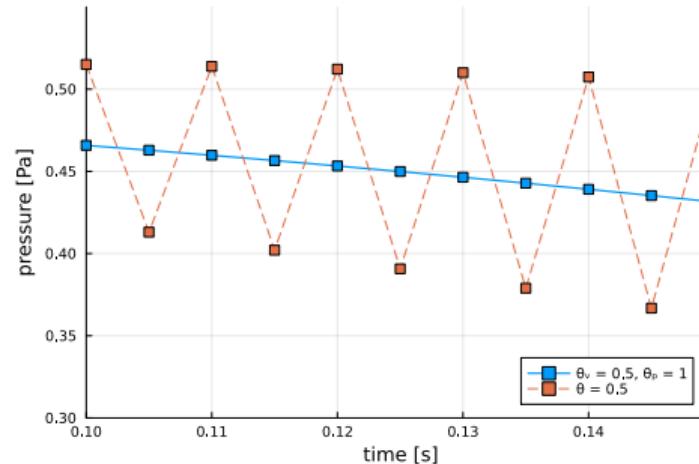


Figure: Pressure oscillations

Segregated formulation

1 Numerical Methods

$$T \frac{\partial \vec{x}}{\partial t} + A\vec{x} = 0 \quad (14)$$

Applying the θ -method

$$T \frac{\partial \vec{x}}{\partial t} = -A\vec{x} \quad (15)$$

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

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

Matrices splitting

1 Numerical Methods

Splitting the problem in solving velocity and pressure field:

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

$$T = \begin{bmatrix} 0 & T_{pu} \\ 0 & T_{uu} \end{bmatrix} \quad (19)$$

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:

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

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

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

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

Segregated Linearized Equations

1 Numerical Methods

Velocity equation:

$$(T_{uu} + \theta \Delta t A_{uu}) \Delta a^* = -A_{uu} u^m - A_{up} p^m - (T_{uu} + \theta \Delta t A_{uu}) a^m + A_{uu} \Delta t a^m + A_{uu} \Delta t a^m + (1 - \theta) A_{up} \sum_{i=0}^m \Delta p^i \quad (20)$$

Pressure equation:

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

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

Sections

2 Performances

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
- ▶ Synthetic Eddy Method
- ▶ Laminar Separation Bubble
- ▶ Uncertainty Quantification
- ▶ Conclusions

Weak Scaling

2 Performances

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

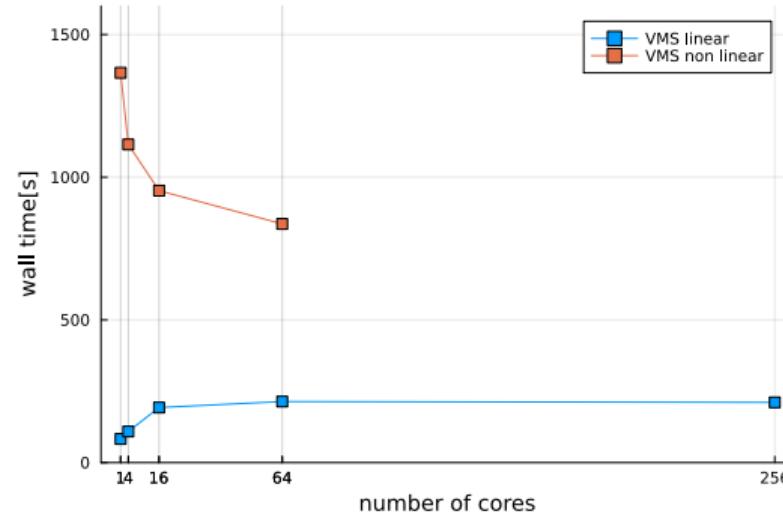


Figure: Taylor Green weak scalability

Strong Scaling

2 Performances

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

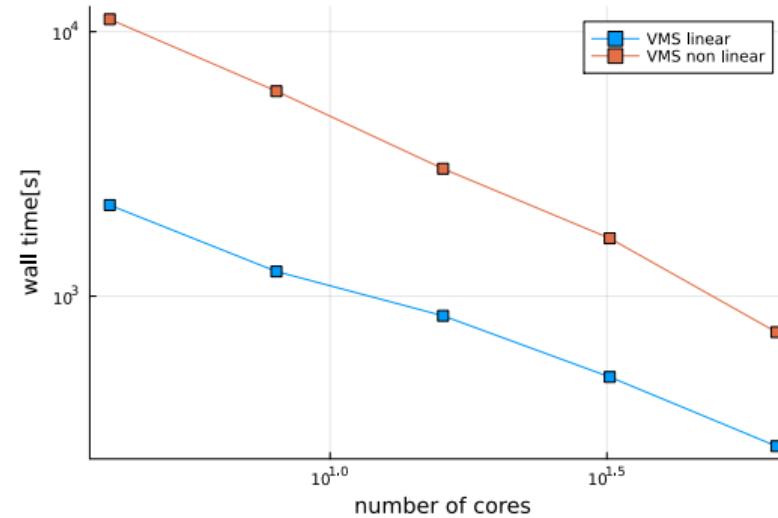


Figure: Taylor Green Strong scalability

Sections

3 Boundary Layer Initialization

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
 - ▶ Synthetic Eddy Method
 - ▶ Laminar Separation Bubble
 - ▶ Uncertainty Quantification
 - ▶ Conclusions

Boundary Layer Initialization

3 Boundary Layer Initialization

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

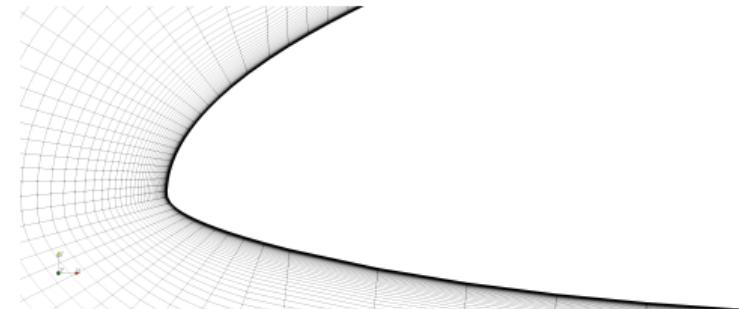
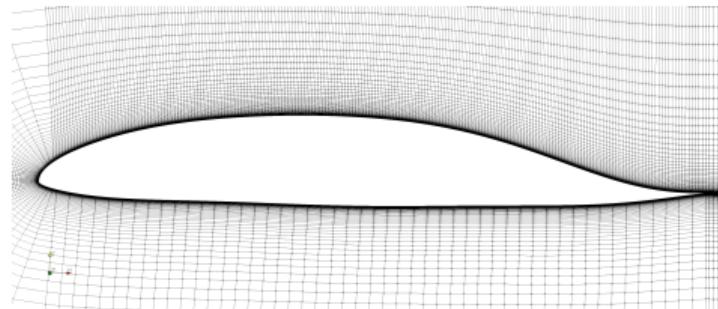


Figure: Mesh DU89

Find wall-distance

3 Boundary Layer Initialization

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

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

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

Find wall-distance

3 Boundary Layer Initialization

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

1. It starts solving the equation for $p=2$

2.

$$\int (\nabla(v) \cdot (\nabla u_{hp})) d\Omega = \int (1 \cdot v) d\Omega \quad (24)$$

3. It solves the linearized equation, which becomes (25). In this case solving for $p=3$ the \tilde{u} is uh_2

$$\int (\nabla(v) \cdot (|\nabla \tilde{u}|^{p-2} \nabla u)) d\Omega = \int (1 \cdot v) d\Omega \quad (25)$$

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

3 Boundary Layer Initialization

The velocity in the x direction in the region identified can be given 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.

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

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

3 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.

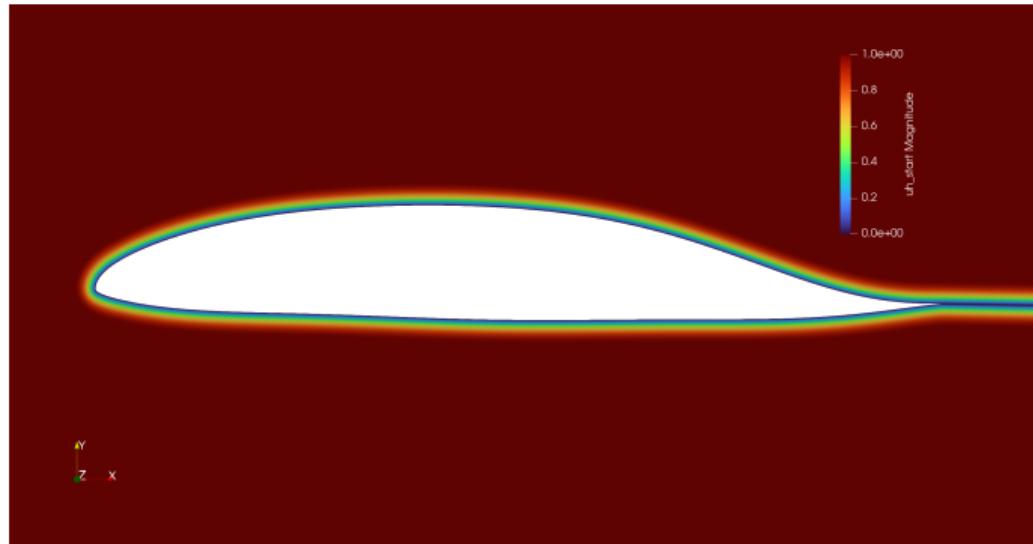


Figure: Boundary Layer Initialization

Sections

4 Synthetic Eddy Method

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
- ▶ Synthetic Eddy Method
- ▶ Laminar Separation Bubble
- ▶ Uncertainty Quantification
- ▶ Conclusions

SyntheticEddyMethod.jl

4 Synthetic Eddy Method

- 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

4 Synthetic Eddy Method

Reynolds decomposition:

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

Compute velocity fluctuations, using a suitable shape function:

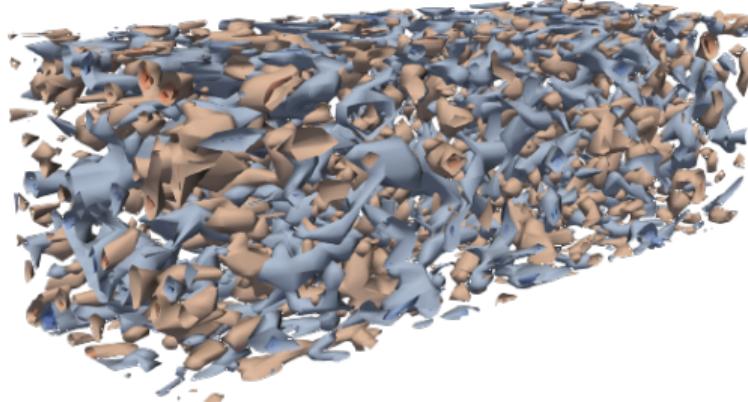
$$u_i(\mathbf{x}) = U_i(\mathbf{x}) + \frac{1}{\sqrt{N}} \sum_{k=1}^N a_{ij} \epsilon_j^k f_{\sigma(\mathbf{x})} (\mathbf{x} - \mathbf{x}^k) \quad (28)$$

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

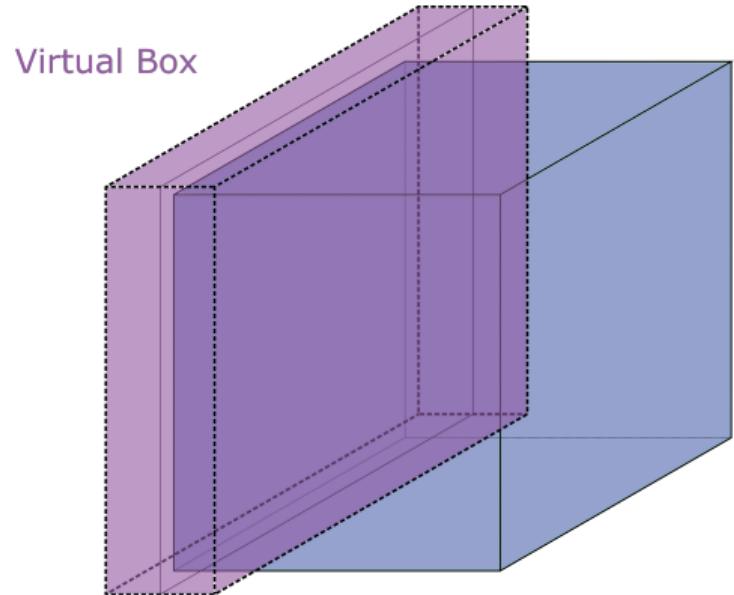
$$f = \begin{cases} \sqrt{\frac{3}{2}}(1 - |x|) & |x| \leq 1 \\ 0 & |x| > 1 \end{cases} \quad (29)$$

Synthetic Eddy Method

4 Synthetic Eddy Method



(a) Isovoltage contour



Computational
Domain

(b) Virtual Box

Synthetic Eddy Method

4 Synthetic Eddy Method

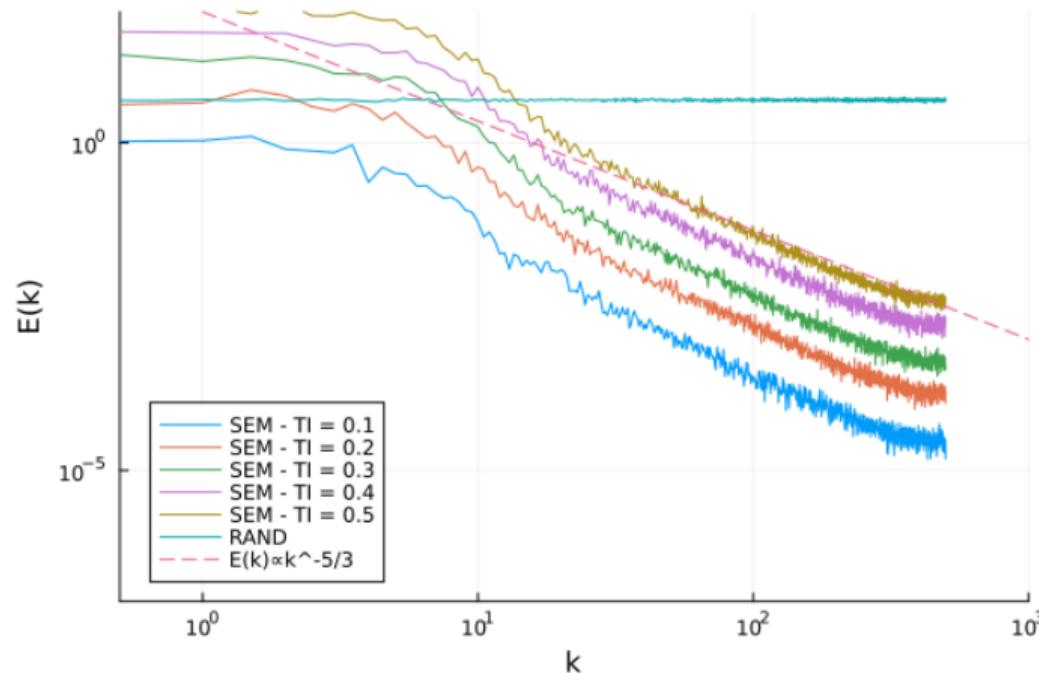


Figure: Power Spectral Density of the turbulent kinetic energy obtained using tent function

Sections

5 Laminar Separation Bubble

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
- ▶ Synthetic Eddy Method
- ▶ Laminar Separation Bubble
- ▶ Uncertainty Quantification
- ▶ Conclusions

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° – 5°

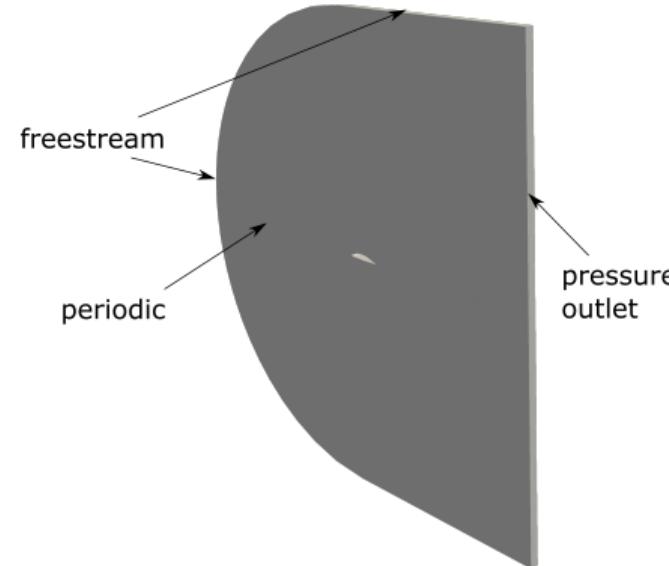
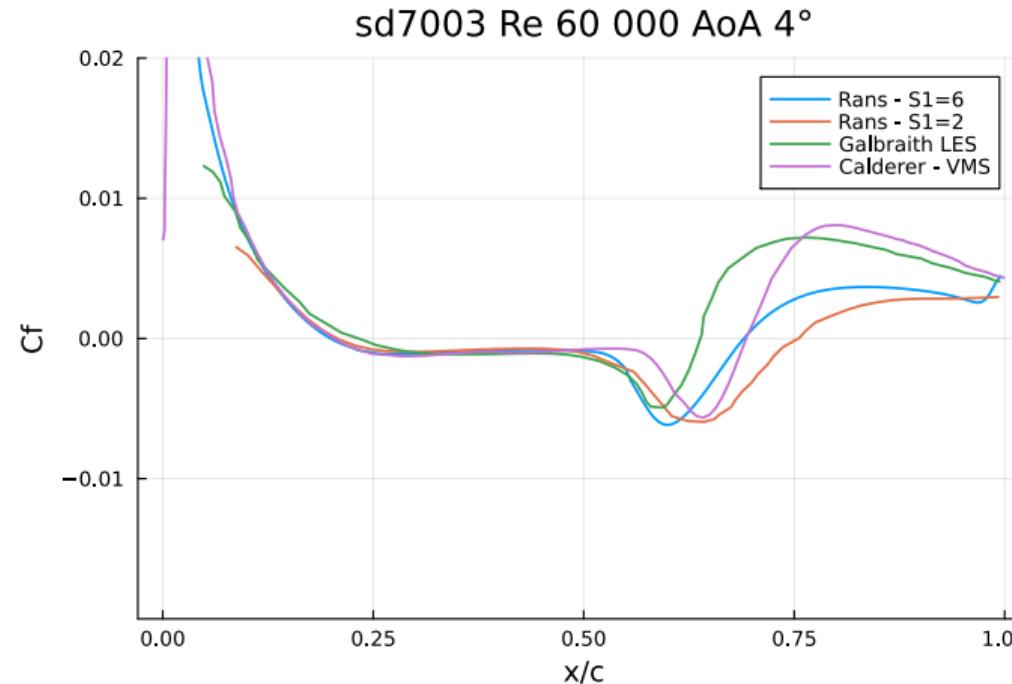


Figure: Airfoils simulations domain

Models

5 Laminar Separation Bubble

sd7003s - Re 60 000 - AoA 4°



VMS linearized coupled sd7003s

5 Laminar Separation Bubble

Coupled: velocity and pressure are solved at the same time. Re 60 000 - AoA 4° .

Initialization with velocity-ramping

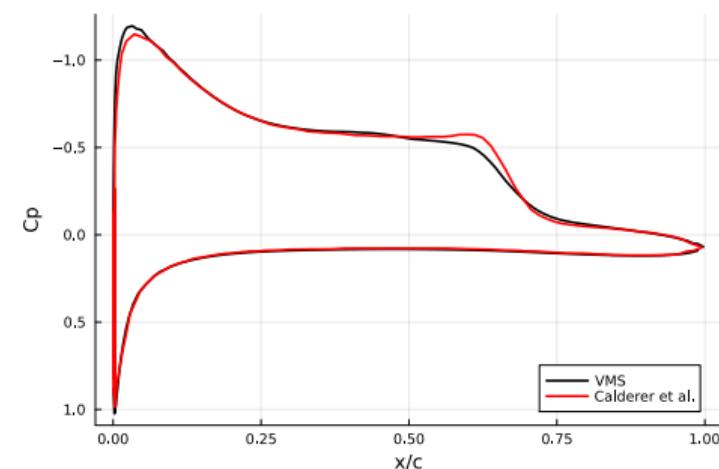
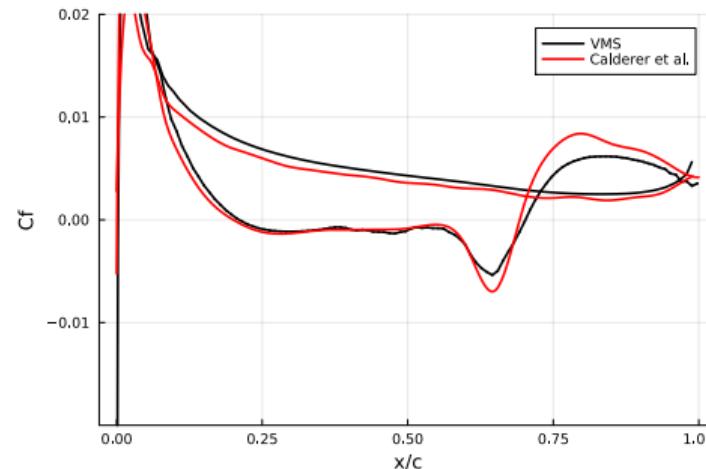


Figure: Comparison with VMS literature results

VMS linearized coupled sd7003s

5 Laminar Separation Bubble

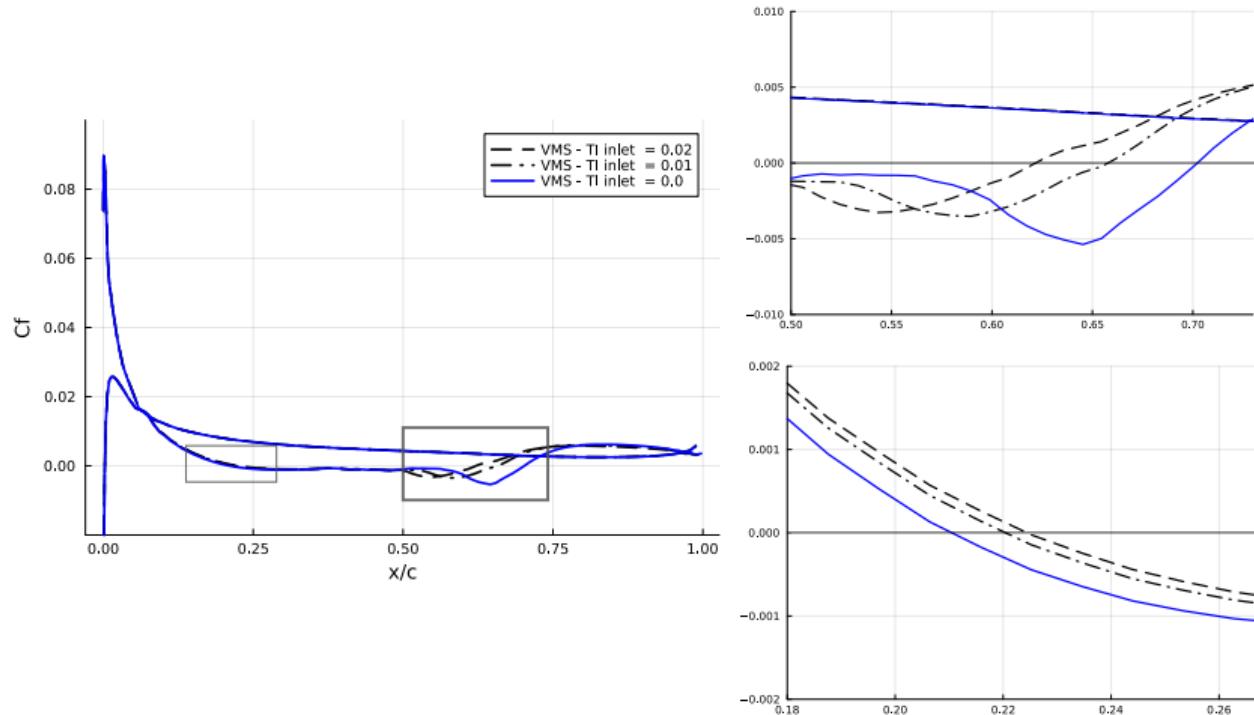
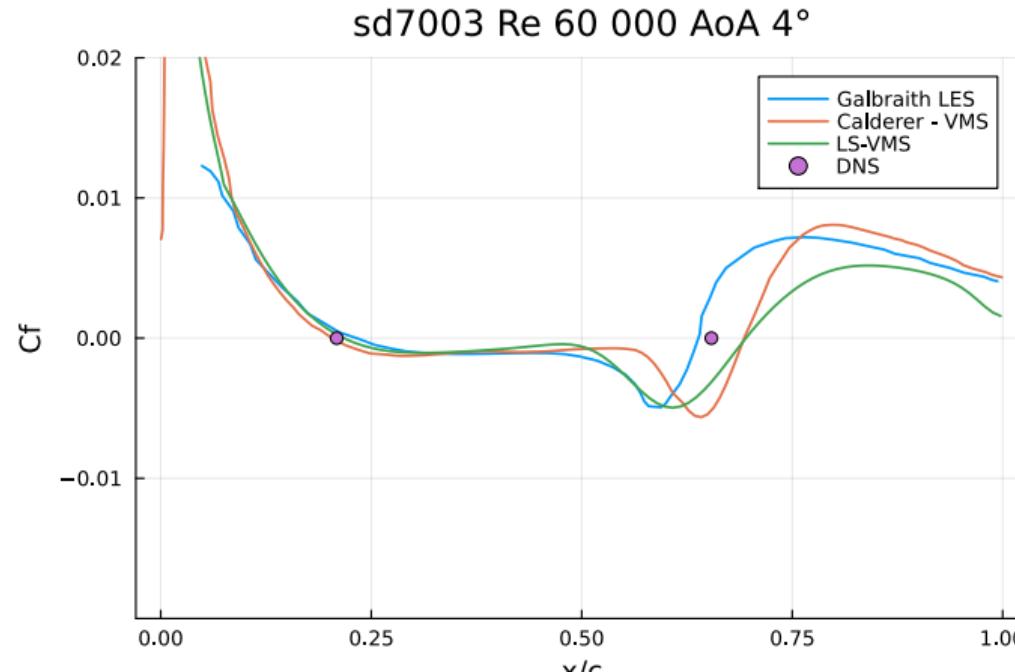


Figure: Bubble position function of freestream turbulence intensity

VMS Linearized-Segregated

5 Laminar Separation Bubble

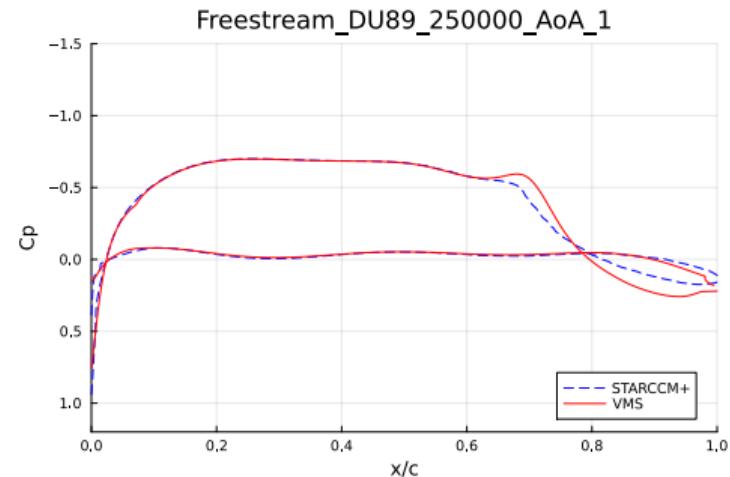
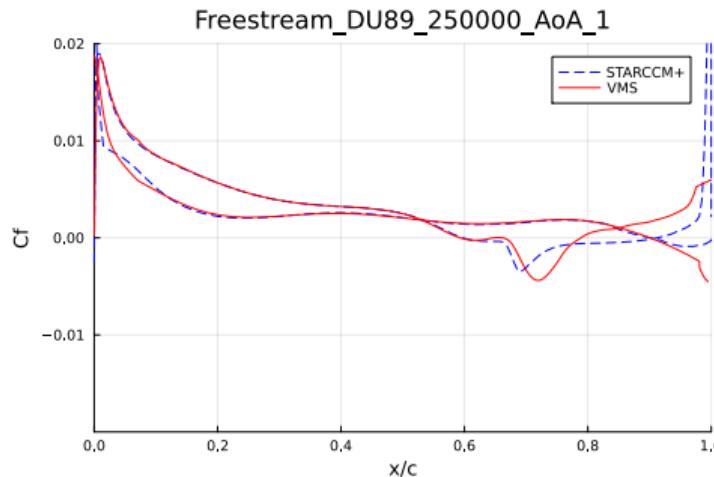
Segregated: each time step pressure and velocity system are solved one after the other multiple times. It is possible to re-use the matrices and preconditioner. It is an iterative method.



VMS Linearized-Segregated

5 Laminar Separation Bubble

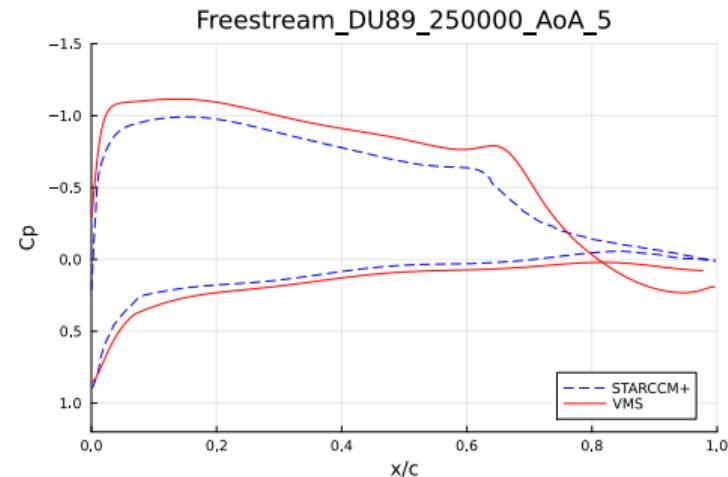
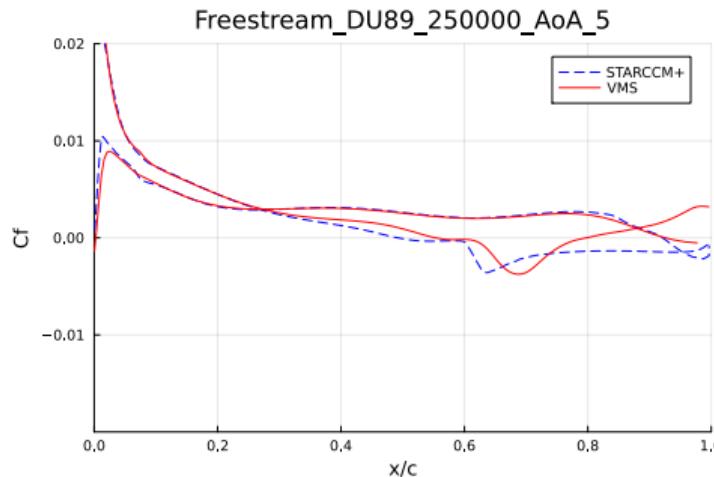
PhD research aims to simulate a new airfoil. Re 250 000 - AoA 1°



VMS Linearized-Segregated

5 Laminar Separation Bubble

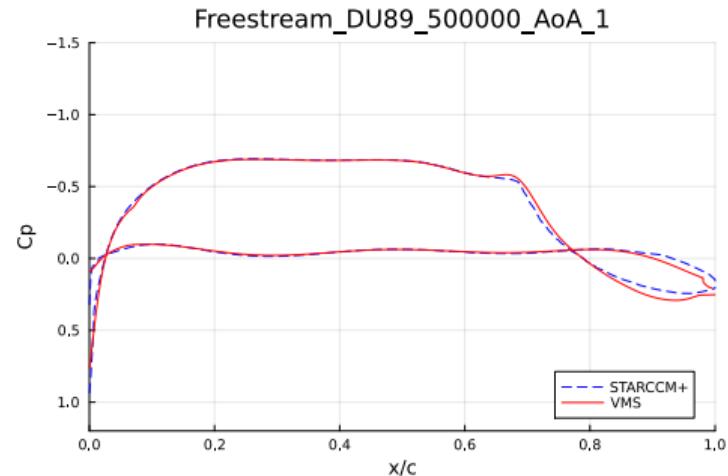
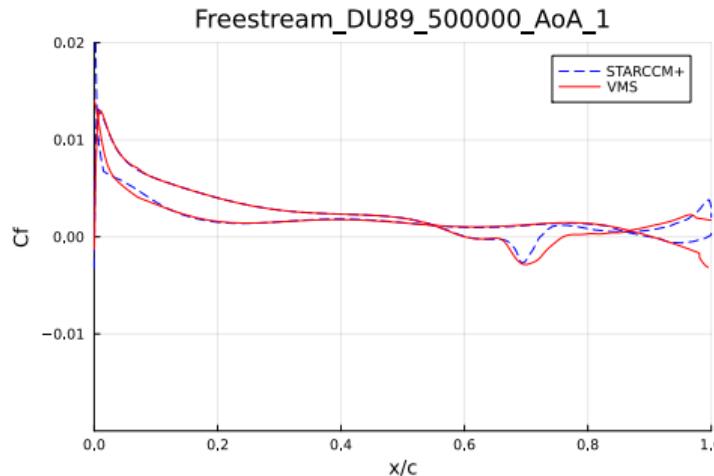
Re 250 000 - AoA 5°



VMS Linearized-Segregated

5 Laminar Separation Bubble

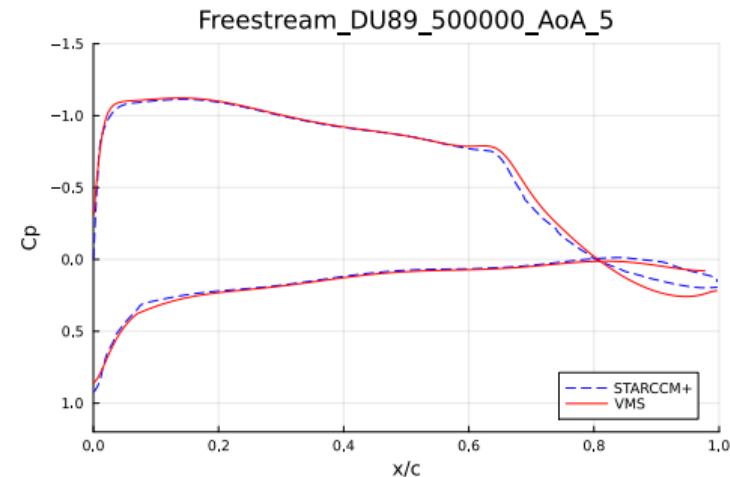
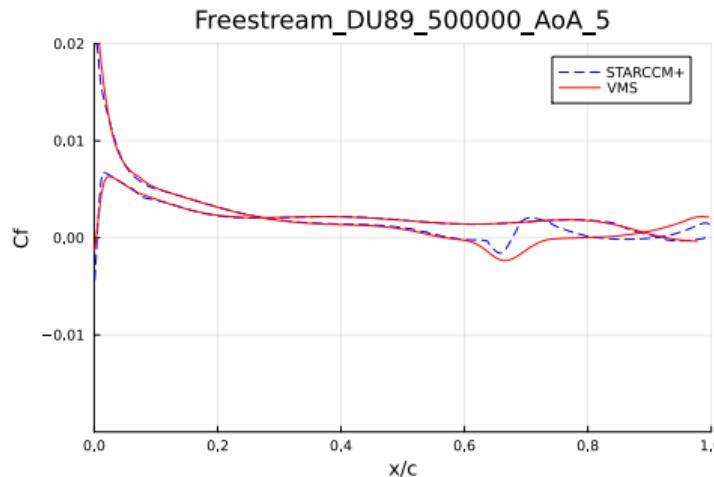
Re 500 000 - AoA 1°



VMS Linearized-Segregated

5 Laminar Separation Bubble

Re 500 000 - AoA 5°



Mesh Sensitivity

5 Laminar Separation Bubble

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

5 Laminar Separation Bubble

Mesh	C_L	C_D	M_{CL}	M_{CD}
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°

$dt[s]$	$2 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-4}$
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 Linearized-Segregated

5 Laminar Separation Bubble

VMS airfoil in wind tunnel

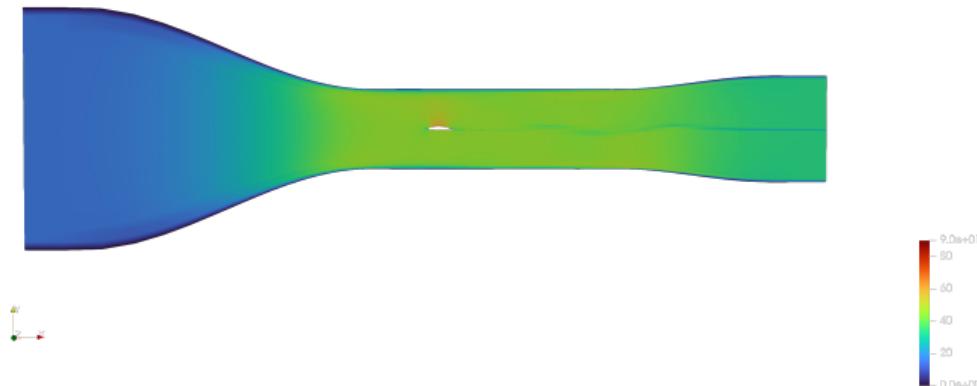


Figure: 3D VMS Wind Tunnel simulation with DU89

Sections

6 Uncertainty Quantification

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
- ▶ Synthetic Eddy Method
- ▶ Laminar Separation Bubble
- ▶ Uncertainty Quantification
- ▶ Conclusions

Model Variables

6 Uncertainty Quantification

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:

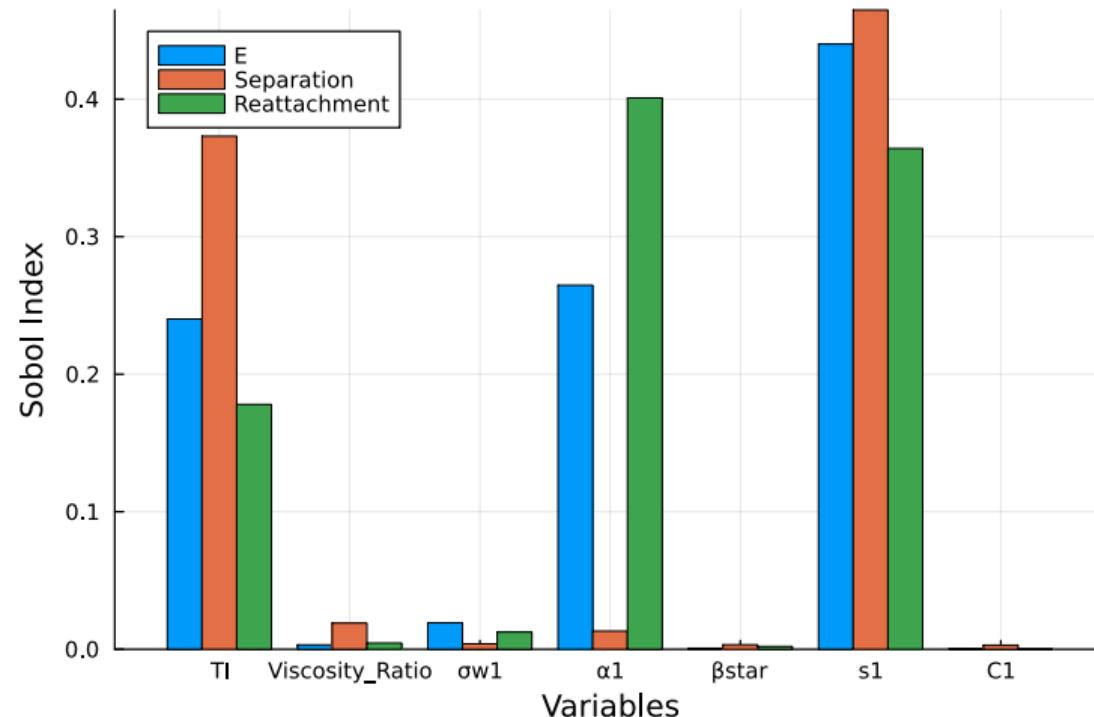
- $\sigma\omega 1 (\mu = 0.5; \sigma = 0.0667)$
- $\alpha 1 (\mu = 0.31; \sigma = 0.03)$
- $\beta * (\mu = 0.09; \sigma = 0.0041)$

γRe_θ parameters

- $s 1 \in [2, 11]$
- $C 1 \in [2, 4]$

Sobol Indexes

6 Uncertainty Quantification



Sections

7 Conclusions

- ▶ Numerical Methods
- ▶ Performances
- ▶ Boundary Layer Initialization
- ▶ Synthetic Eddy Method
- ▶ Laminar Separation Bubble
- ▶ Uncertainty Quantification
- ▶ Conclusions

Other activities

7 Conclusions

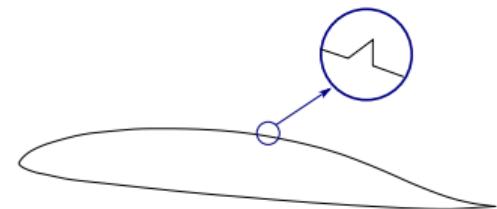
- Synthetic Eddy Method publication
- JuliaCon 2023 conference
- Co-author AIAA 2023 conference paper
- Testing standard passive flow controls (no improvement in aerodynamic efficiency)



(a) Slot



(b) Slot curved



(c) Riblet

Figure: Passive Flow Controls

Elements of novelty

7 Conclusions

- 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

7 Conclusions

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