Time-resolved Adaptive Direct FEM Simulation of High-lift Aircraft Configurations

Johan Jansson, Ezhilmathi Krishnasamy, Massimiliano Leoni, Niclas Jansson and Johan Hoffman

Abstract We present an adaptive finite element method for time-resolved simulation of aerodynamics without any turbulence model parameters, which is applied to a benchmark problem from the HiLiftPW-3 workshop to compute the flow past a JAXA Standard Model (JSM) aircraft model at realistic Reynolds number. The mesh is automatically constructed by the method as part of an adaptive algorithm based on a posteriori error estimation using adjoint techniques. No explicit turbulence model is used, and the effect of unresolved turbulent boundary layers is modeled by a simple parametrization of the wall shear stress in terms of a skin friction. In the case of very high Reynolds numbers we approximate the small skin friction by zero skin friction, corresponding to a free slip boundary condition, which results in a computational model without any model parameter to be tuned, and without the need for costly boundary layer resolution. We introduce a numerical tripping noise term to act as a seed for growth of perturbations, the results support that this triggers the correct physical separation at stall, and has no significant effect pre-stall. We show that the methodology quantitatively and qualitatively captures the main features of the JSM experiment - aerodynamic forces and the stall mechanism - with a much coarser mesh resolution and lower computational cost than the state of the art methods in the field, with convergence under mesh refinement by the adaptive method. Thus, the simulation methodology appears to be a possible answer to the challenge of reliably predicting turbulent-separated flows for a complete air vehicle.

Johan Jansson (Assistant Professor, e-mail: `jjan@kth.se`),
Ezhilmathi Krishnasamy (PhD candidate, e-mail: `ekrishnasamy@bcamath.org`), and
Massimiliano Leoni (PhD candidate, e-mail: `mleoni@kth.se`),
Computational Science and Technology, CSC, KTH, SE-10044 Stockholm, Sweden and
BCAM - Basque Center for Applied Mathematics, Bilbao, Spain

Niclas Jansson (Postdoctoral Researcher, e-mail: `njansson@kth.se`) and
Johan Hoffman (Professor, e-mail: `jhoffman@kth.se`),
Computational Science and Technology, CSC, KTH, SE-10044 Stockholm, Sweden

Nomenclature

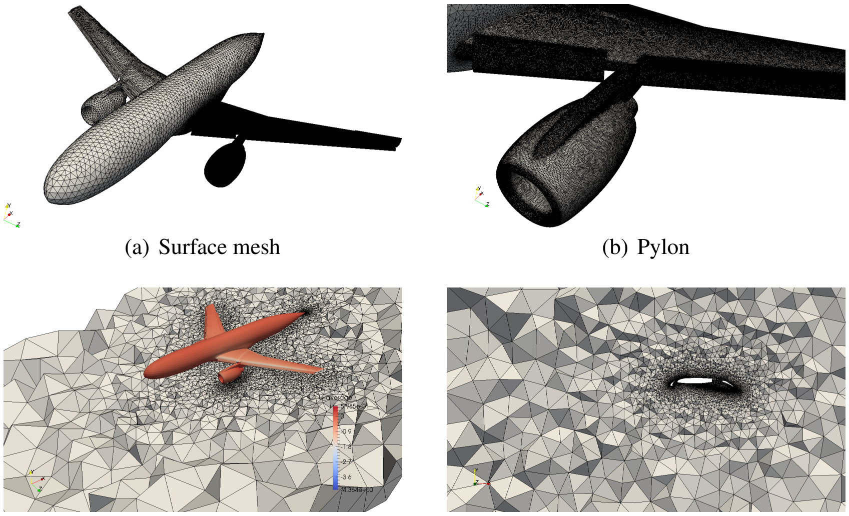
C_l	lift coefficient (dimensionless)
C_d	drag coefficient (dimensionless)
C_p	pressure coefficient (dimensionless)
h	diameter of tetrahedron in finite element mesh (m)
k	time step (s)
$\mathbf{n}$	normal unit vector (dimensionless)
P	computed pressure (Pa)
p	pressure (Pa)
q	pressure test function (Pa)
Re	Reynolds number (dimensionless)
t	time variable (s)
T	end time (s)
$\mathbf{U}$	computed velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{u}$	velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{v}$	velocity test function ($\mathrm{m\,s^{-1}}$)
$\mathbf{x}$	space variable (m)
α	angle of attack (dimensionless)
β	friction parameter ($\mathrm{kg\,m^{-2}\,s^{-1}}$)
ν	Kinematic viscosity ($\mathrm{m^2\,s^{-1}}$)
τ	tangent unit vector (dimensionless)

1 Introduction

The main challenge today in Computational Fluid Dynamics (CFD) for aerodynamics is to reliably predict turbulent-separated flows Witherden & Jameson (2017), Slotnick et al. (2014), specifically for a complete air vehicle. This is our focus in this chapter.

We present an adaptive finite element method without turbulence modeling parameters for time-resolved simulation of aerodynamics, together with results stemming from the 3rd AIAA CFD High-Lift Prediction Workshop (HiLiftPW-3) which was held in Denver, Colorado, on June 3rd–4th 2017. The benchmark was a high-lift configuration of the JSM aircraft model shown in Figure 1 at a Reynolds number realistic for flight conditions. The purpose of the workshop is to assess the capability of state of the art CFD codes and methods.

Turbulent flows present features on a range of scales, from the scale of the aircraft down to the Kolmogorov dissipation scale. Direct numerical simulation (DNS) is not feasible for a full aircraft at realistic Reynolds numbers, instead the Reynolds Averaged Navier-Stokes equations (RANS) have long been the state of the art in industry Shan et al. (2005). RANS methods do not provide a full resolution of the flow field but simulate the mean field and introduce turbulence models to make it


(a) Surface mesh

(b) Pylon

(c) Volume mesh

(d) Wing slice

Fig. 1 Overview of the JSM aircraft model and starting mesh for the adaptive method.

up for the unresolved dynamics. In particular, standard RANS do not resolve the transient flow field, but a statistical average of the turbulent flow.

In contrast, Large Eddy Simulations (LES) Sagaut (2005) resolve the dynamics of a filtered flow field, at the cost of higher mesh resolution than RANS, with subgrid models for unresolved scales. Both RANS and LES, and hybrids such as DES, introduce model parameters that need to be tuned to the problem at hand, and the results are highly sensitive to the design of the computational mesh Moin & You (2008), Huang et al. (2004), Spalart (2009), Piomelli & Balaras (2002), Mellen et al. (2003). In particular, turbulent boundary layers cannot be resolved and must be modelled. Boundary layer models require tailored boundary layer meshes, which are expensive in terms of both mesh density and manual work. Witherden and Jameson in Witherden & Jameson (2017) state that "as a community we are still far away from LES of a complete air vehicle".

The method we present is an adaptive finite element method without explicit turbulence model and boundary layer model, thus without model parameters and without the need for a boundary layer mesh. The mesh is automatically constructed by the method as part of the computation through an adaptive procedure based on a posteriori error estimation using adjoint techniques. Dissipation of turbulent kinetic energy is provided by residual-based numerical stabilization. The method is thus purely based on the Navier-Stokes equations, no other modeling assumptions are made.

We model the effect of turbulent boundary layers by a parametrization of the wall shear stress in terms of a skin friction. For very high Reynolds numbers we approximate the small skin friction by zero skin friction, corresponding to a free slip boundary condition, which results in a computational method without any model parameters that need to be tuned, and without the need for costly boundary layer resolution.

In this chapter we give the main components of the simulation methodology and we present our results stemming from the HiLiftPW-3, where we highlight the non-standard aspects of the methodology and discuss the results in relation to the experiments. The HiLiftPW-3 specified two variants of the JSM, one without pylon (or nacelle) and one with the pylon included in the geometry ("pylon on"). The difference in the aerodynamic forces between the two variants measured in experiments are small, typically less than 2 %. For this reason we will focus only on the "pylon on" variant with the aim of validating our methodology.

The workshop guidelines prescribed the study of these two variants either with a fixed mesh or, more interestingly, using mesh adaptation techniques. Considering the nature of our method, which intimately depends on its adaptive procedure, we concentrated on the latter study. We did not use the provided computational meshes, but instead generated more suitable ones for our methodology starting from the provided CAD files. We would like to point out that our adaptive methodology does not require any ad-hoc meshing procedure aimed at helping the solver identify flow features that are qualitatively known before starting the computations. Not only does this simplify the meshing procedure, which can now be carried out by non-specialized software (and scientists), but it also makes it faster: the only thing that we need is an initial mesh that captures the geometry of the object; this is due to the fact that the generated mesh loses memory of the underlying CAD model, and therefore the refinement of boundary triangles cannot correct a rough initial approximation of the CAD geometry. We plan to get rid of this constraint in the near future, implementing the functionality to refine boundary cells with the new vertices projected on the CAD model. Once we have a sufficiently accurate surface description, however, we can let the mesh be coarse in the volume part, which will be refined iteratively by the adaptive algorithm.

This convenient approach allows us to perform computations starting with rather coarse meshes, increasing the number of cells only where needed in order to best utilize the available computational resources. Our initial mesh for the JSM case have about 2 5M cells.

We find that the simulation results compare very well with experimental data for all the angles of attack that we studied; moreover, we show mesh-convergence by the adaptive method, while using a relatively low number of spatial degrees of freedom. The low computational cost also allows for a time-resolved simulation, which provides additional results that cannot be obtained from a stationary simulation, such as the ones based on Reynolds-averaged Navier-Stokes equations (RANS).

Thus, the simulation methodology appears to be a possible answer to the challenge of reliably predicting turbulent-separated flows for a complete air vehicle. We specifically here present simulation results reproducing the physically correct stall mechanism of large-scale separation at the wing-body juncture, which is promising for our continuing work on validating the methodology.

2 Simulation Methodology

In contrast to the statistical averages of RANS and the filtered solutions of LES, our simulation method is based on computational approximation of weak solutions to the Navier-Stokes equations (NSE), that satisfy the NSE in variational form integrated against a class of test functions.

Finite element methods (FEM) are based on a variational form of the NSE, and if the method satisfies certain conditions on stability and consistency, the FEM solutions converge towards a weak solution to the NSE as the finite element mesh is refined Hoffman & Johnson (2007). We refer to such FEM as a General Galerkin (G2) method, or a Direct Finite Element simulation (DFS).

The resolution in DFS is set by the mesh size, and no turbulence model is introduced. Dissipation of turbulent kinetic energy in under-resolved parts of the flow is provided by the numerical stabilization of G2 in the form of a weighted least squares method based on the residual of NSE.

The mesh is adaptively constructed based on a posteriori estimation of the error in chosen goal or target functionals, such as drag and lift forces. The a posteriori error estimates take the form of a residual weighted by the solution of an adjoint problem, which is computed separately using a similar stabilized FEM method Hoffman & Johnson (2007). The adaptive algorithm starts from a coarse mesh, which is locally refined each iteration based on the a posteriori error estimates.

We use a free slip boundary condition as a model for high Reynolds number turbulent boundary layers with small skin friction. This means that boundary layers are left unresolved, and that no boundary layer mesh is needed.

This methodology has been validated on a number of standard benchmark problems in the literature Hoffman (2005, 2006), Hoffman & Johnson (2006*b*), Hoffman (2009), including for an aircraft model for the HiLiftPW-2 Hoffman, Jansson, Jansson & Abreu (2015) and we find that also for the benchmark considered in this chapter the method is very efficient and provides results close to the experimental reference data.

We have used a low order finite element discretization on unstructured tetrahedral meshes, which we refer to as cG(1)cG(1), id est, continuous piecewise linear approximation in space and time.

2.1 The cG(1)cG(1) method

As the basic model for incompressible Newtonian fluid flow, we consider the NSE with constant kinematic viscosity $\nu > 0$, enclosed in $\Omega \subset \mathbb{R}^3$, with boundary Γ, over a time interval $I = [0, T]$:

$$\begin{cases} \dot{\mathbf{u}} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - 2\nu\nabla\cdot\varepsilon(\mathbf{u}) = f, & (\mathbf{x},t)\in\Omega\times I,\\ \nabla\cdot\mathbf{u} = 0, & (\mathbf{x},t)\in\Omega\times I,\\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}^0(\mathbf{x}), & \mathbf{x}\in\Omega, \end{cases} \tag{1}$$

with $\mathbf{u}(\mathbf{x},t)$ the velocity vector, $p(\mathbf{x},t)$ the pressure, $\mathbf{u}^0(\mathbf{x})$ the initial data and $f(\mathbf{x},t)$ a body force. Moreover, $\sigma_{ij} = 2\nu\varepsilon_{ij}(\mathbf{u}) - p\delta_{ij}$ is the stress tensor, with the strain rate tensor $\varepsilon_{ij}(\mathbf{u}) = 1/2(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$, and δ_{ij} the Kronecker delta. The relative importance of viscous and inertial effects in the flow is determined by the Reynolds number $Re = \mathbf{U}L/\nu$, where $\mathbf{U}$ and L are characteristic velocity and length scales.

The cG(1)cG(1) method is based on the continuous Galerkin method cG(1) in space and time. With cG(1) in time, the trial functions are continuous, piecewise linear and the test functions piecewise constant. cG(1) in space corresponds to both test functions and trial functions being continuous, piecewise linear.

Let $0 = t_0 < t_1 < ... < t_N = T$ be a sequence of discrete time steps, with associated time intervals $I_n = (t_{n-1}, t_n)$ of length $k_n = t_n - t_{n-1}$, and let $W \subset H^1(\Omega)$ be a finite element space consisting of continuous, piecewise linear functions on a tetrahedral mesh $\mathscr{T} = \{K\}$ of mesh size $h(\mathbf{x})$, with $W_{\mathbf{w}}$ the functions $\mathbf{v} \in W$ satisfying the Dirichlet boundary condition $\mathbf{v}|_\Gamma = \mathbf{w}$.

We seek $\hat{\mathbf{U}} = (\mathbf{U}, P)$, continuous piecewise linear in space and time, and the cG(1)cG(1) method for the NSE with homogeneous Dirichlet boundary conditions reads: for $n = 1,...,N$ find $(\mathbf{U}^n, P^n) \equiv (\mathbf{U}(t_n), P(t_n))$, with $\mathbf{U}^n \in V_0 \equiv [W_0]^3$ and $P^n \in W$, such that:

$$\begin{aligned} &((\mathbf{U}^n - \mathbf{U}^{n-1})k_n^{-1} + \bar{\mathbf{U}}^n\cdot\nabla\bar{\mathbf{U}}^n, \mathbf{v}) + (2\nu\varepsilon(\bar{\mathbf{U}}^n), \varepsilon(\mathbf{v})) - (P^n, \nabla\cdot\mathbf{v})\\ &+ (\nabla\cdot\bar{\mathbf{U}}^n, q) + SD_\delta^n(\bar{\mathbf{U}}^n, P^n; \mathbf{v}, q) = (f, \mathbf{v}), \quad \forall\hat{\mathbf{v}} = (\mathbf{v}, q) \in V_0 \times W, \end{aligned} \tag{2}$$

where $\bar{\mathbf{U}}^n = \frac{1}{2}(\mathbf{U}^n + \mathbf{U}^{n-1})$ is piecewise constant in time over I_n, with the stabilizing term

$$\begin{aligned} &SD_\delta^n(\bar{\mathbf{U}}^n, P^n; \mathbf{v}, q) \equiv\\ &\qquad (\delta_1(\bar{\mathbf{U}}^n\cdot\nabla\bar{\mathbf{U}}^n + \nabla P^n - f), \bar{\mathbf{U}}^n\cdot\nabla\mathbf{v} + \nabla q) + (\delta_1\nabla\cdot\bar{\mathbf{U}}^n, \nabla\cdot\mathbf{v}), \end{aligned} \tag{3}$$

and

$$(\mathbf{v}, \mathbf{w}) = \sum_{K\in\mathscr{T}} \int_K \mathbf{v}\cdot\mathbf{w}\, dx,$$

$$(\varepsilon(\mathbf{v}), \varepsilon(\mathbf{w})) = \sum_{i,j=1}^{3} (\varepsilon_{ij}(\mathbf{v}), \varepsilon_{ij}(\mathbf{w})),$$

with the stabilization parameter $\delta_1 = \kappa_1 h$, where κ_1 is a positive constant of unit size. We choose a time step size $k_n = C_{CFL}\min_{\mathbf{x}\in\Omega} h/|\mathbf{U}^{n-1}|$, with C_{CFL} typically in the range $[0.5, 20]$. The resulting non-linear algebraic equation system is solved with a robust Schur-type fixed-point iteration method Houzeaux et al. (2009).

2.2 The Adaptive Algorithm

A simple description of the adaptive algorithm, starting from $i = 0$, reads:

1. For the mesh $\mathscr{T}_i$: solve the primal and (linearized) dual problems for the primal solution $(\mathbf{U}, P)$ and the dual solution (Φ, Θ).
2. Compute the quantity $\mathscr{E}_K$ for any cell K of $\mathscr{T}_i$. If $\sum_{K \in \mathscr{T}_i} \mathscr{E}_K < TOL$ then stop, else:
3. Mark 5% of the elements with highest $\mathscr{E}_K$ for refinement.
4. Generate the refined mesh $\mathscr{T}_{i+1}$, and goto 1.

Here, $\mathscr{E}_K$ is the *error indicator* for each cell K, which we describe in Section 2.3. For now, it suffices to say that $\mathscr{E}_K$ is a function of the residual of the NSE and of the solution of a linearized dual problem. The formulation of the dual problem includes the definition of a *target functional* for the refinement, which usually enters the dual equations as a boundary condition or as a volume source term. This functional should be chosen according to the problem we are solving. In other words, one needs to ask the right question in order to obtain the correct answer from the algorithm. In this chapter our target functional is chosen to be the mean value in time of the aerodynamic forces.

The dual problem can be written as (see Hoffman & Johnson (2006*a*) for more details):

$$\begin{cases} -\dot{\varphi} - (\mathbf{u} \cdot \nabla)\varphi + \nabla \mathbf{U}^\top \varphi + \nabla \theta - \nu \Delta \varphi = \psi_1 & (\mathbf{x}, t) \in \Omega \times I \\ \nabla \cdot \varphi = \psi_2 & (\mathbf{x}, t) \in \Omega \times I \\ \varphi = \psi_3 & (\mathbf{x}, t) \in \Gamma \times I \\ \varphi(\cdot, T) = \psi_4 & \mathbf{x} \in \Omega, \end{cases} \tag{4}$$

where we find that the structure is similar to the primal NSE equations, except that the adjoint problem is linear, the transport is backward in time, and that we have a reaction term $(\nabla \mathbf{U}^\top \varphi)_j = U_{,j} \cdot \varphi$, not present in the primal NSE.

The only other input required from the user is an initial discretization of the geometry, $\mathscr{T}_0$. Since our method is designed for tetrahedral meshes that do not require any special treatment of the near-wall region (no need for a boundary-layer mesh), the initial mesh can be easily created with any standard mesh generation tool.

2.3 A posteriori error estimate for cG(1)cG(1)

The a posteriori error estimate is based on the following theorem (for a detailed proof, see chapter 30 in Hoffman & Johnson (2007)):

Theorem 1. *If $\hat{\mathbf{U}} = (\mathbf{U}, P)$ solves (2), $\hat{\mathbf{u}} = (\mathbf{u}, p)$ is a weak NSE solution, and $\hat{\varphi} = (\varphi, \theta)$ solves an associated dual problem with data $M(\cdot)$, then we have the* *following a posteriori error estimate for the target functional $M(\hat{\mathbf{U}})$ with respect to the reference functional $M(\hat{\mathbf{u}})$:*

$$|M(\hat{\mathbf{u}}) - M(\hat{\mathbf{U}})| \leq \sum_{n=1}^{N} \left[\int_{I_n} \sum_{K \in \mathscr{T}_i} |R_1(\mathbf{U},P)_K| \cdot \omega_1 \, dt \right.$$

$$\left. + \int_{I_n} \sum_{K \in \mathscr{T}_i} |R_2(\mathbf{U})_K| \; \omega_2 \, dt + \int_{I_n} \sum_{K \in \mathscr{T}_i} |SD_\delta^n(\hat{\mathbf{U}};\hat{\varphi})_K| \, dt \right] =: \sum_{K \in \mathscr{T}_i} \mathscr{E}_K$$

with

$$\begin{aligned} R_1(\mathbf{U},P) &= \dot{\mathbf{U}} + (\mathbf{U}\cdot\nabla)\mathbf{U} + \nabla P - 2\nu\nabla\cdot\varepsilon(\mathbf{u}) - f, \\ R_2(\mathbf{U}) &= \nabla\cdot\mathbf{U}, \end{aligned} \tag{5}$$

where $SD_\delta^n(\cdot;\cdot)_K$ is a local version of the stabilization form (3), and the stability weights are given by

$$\begin{aligned} \omega_1 &= C_1 h_K |\nabla\varphi|_K, \\ \omega_2 &= C_2 h_K |\nabla\theta|_K, \end{aligned}$$

where h_K is the diameter of element K in the mesh $\mathscr{T}_i$, and $C_{1,2}$ represent interpolation constants. Moreover, $|w|_K \equiv (\|w_1\|_K, \|w_2\|_K, \|w_3\|_K)$, with $\|w\|_K = (w,w)_K^{1/2}$, and the dot denotes the scalar product in $\mathbb{R}^3$.

For simplicity, it is here assumed that the time derivatives of the dual variables $\hat{\varphi} = (\varphi, \theta)$ can be bounded by their spatial derivatives. Given Theorem 1, we can understand the adaptive algorithm. As mentioned above, the error indicator, $\mathscr{E}_K$, is a function of the residual of the NSE and the solution of a linearized dual problem (a detailed formulation of the dual problem is given in Chapter 14 in Hoffman & Johnson (2007)). Thus, on a given mesh, we must first solve the NSE to compute the residuals, $R_1(\mathbf{U},P)$ and $R_2(\mathbf{U})$, and then a linearized dual problem to compute the weights multiplying the residuals, ω_1 and ω_2. With that information, we are able to compute $\sum_{K\in\mathscr{T}_i} \mathscr{E}_K$ and check it against the given stop criterion. This procedure of solving the forward and backward problems for the NSE is closely related to an optimization loop and can be understood as the problem of finding the "optimal mesh" for a given geometry and boundary conditions, id est, the mesh with the least possible number of degrees of freedom for computing $M(\hat{\mathbf{u}})$ within a given degree of accuracy.

2.4 The Do-nothing Error Estimate and Indicator

To minimize loss of sharpness, we also investigate an approach where the weak form is used directly in a posteriori error estimates, without integration by parts to the strong form, using the Cauchy-Schwarz inequality and interpolation estimates.

We here refer to this direct form of a posteriori error representation by duality as the "do-nothing" approach.

In terms of the exact adjoint solution $\hat{\varphi}$, the output error with respect to a weak solution $\hat{\mathbf{u}}$ can be represented as

$$|M(\hat{\mathbf{u}}) - M(\hat{\mathbf{U}})| = |(R(\hat{\mathbf{U}}), \hat{\varphi})| = |\sum_{K \in \mathscr{T}_i} (R(\hat{\mathbf{U}}), \hat{\varphi})_K| \tag{6}$$

This error representation involves no approximation or inequalities. We thus refer to the following error indicator based on the representation as the *do-nothing error indicator*:

$$e^K \equiv (R(\hat{\mathbf{U}}), \hat{\varphi})_K \tag{7}$$

A computable estimate and an error indicator are again based on the computed approximation $\hat{\varphi}_h$ of the dual solution:

$$|M(\hat{\mathbf{u}}) - M(\hat{\mathbf{U}})| \approx |(R(\hat{\mathbf{U}}), \hat{\varphi}_h)| \tag{8}$$

$$e_h^K \equiv (R(\hat{\mathbf{U}}), \hat{\varphi}_h)_K \tag{9}$$

where we may lose reliability of the global error estimate by the Galerkin orthogonality property, which states that the $(R(\hat{\mathbf{U}}), \hat{\varphi}_h)$ vanishes for a standard Galerkin finite element method if $\hat{\varphi}_h$ is chosen in the same space as the test functions. Although, in the setting of a stabilised finite element method this may not be the case, see Hoffman et al. (2016).

2.5 Turbulent boundary layers

In our work on high Reynolds number turbulent flows Hoffman & Jansson (2010), Hoffman & Johnson (Published Online First at www.springerlink.com: 10 December 2008), Vilela de Abreu et al. (2014) we have chosen to apply a skin friction stress as wall layer model. That is, we append the NSE with the following boundary conditions:

$$\mathbf{u} \cdot \mathbf{n} = 0, \tag{10}$$

$$\beta \mathbf{u} \cdot \tau_k + \mathbf{n}^T \sigma \tau_k = 0, \quad k = 1,2, \tag{11}$$

for $(\mathbf{x}, t) \in \Gamma_{solid} \times I$, with $\mathbf{n} = \mathbf{n}(\mathbf{x})$ an outward unit normal vector, and $\tau_k = \tau_k(\mathbf{x})$ orthogonal unit tangent vectors of the solid boundary Γ_{solid}. We use matrix notation with all vectors $\mathbf{v}$ being column vectors and the corresponding row vector being denoted by $\mathbf{v}^T$.

With skin friction boundary conditions, the rate of kinetic energy dissipation in cG(1)cG(1) has a contribution of the form

$$\sum_{k=1}^{2}\int_0^T\int_{\Gamma_{solid}}|\beta^{1/2}\bar{\mathbf{U}}\cdot\tau_k|^2\,ds\,dt, \tag{12}$$

from the kinetic energy which is dissipated as friction in the boundary layer. For high Re, we model $Re\to\infty$ by $\beta\to 0$, so that the dissipative effect of the boundary layer vanishes with large Re. In particular, we have found that a small β does not influence the solution Hoffman & Jansson (2010). For the present simulations we used the approximation $\beta=0$, which can be expected to be a good approximation for real high-lift configurations, where Re is very high.

2.6 Numerical tripping

The simulation setting so far is idealized in the sense that the inflow is noise-free, the surfaces have no roughness, there are no vibrations in the surface, etc. This is not a realistic setting.

In the DNS community the effect of introducing noise has been investigated in Schlatter & Orlu (2012), and it turns out that in idealized settings different DNS methods and frameworks may get different results for the same problem, but introducing a noise term has the effect of making the results more uniform.

We explore a similar idea here, where we add a volume force term of simple white in a domain approximately the bounding box of the aircraft geometry. We want the noise to only slightly perturb the solution, to act as a seed for growth of perturbations in unstable mechanisms (such as stall), but we do not want the noise to dominate the solution.

To achieve this balanced effect we scale the white noise force term by 5% of the maximum pressure gradient $|\nabla p|$.

We investigate the effect of such numerical tripping in the results section, comparing simulations with and without the tripping. We will see that especially for stall this appears to have a key effect in triggering the correct physical separation.

2.7 The FEniCS-HPC finite element computational framework

The simulations in this article have been computed using the Unicorn solver in the FEniCS-HPC automated FEM software framework.

FEniCS-HPC Hoffman, Jansson & Jansson (2015) is an open source framework for the automated solution of PDEs on massively parallel architectures, providing automated evaluation of variational forms whose description is given in a high-level mathematical notation, duality-based adaptive error control, implicit turbulence modeling by use of stabilized FEM and strong linear scaling up to thousands of cores Hoffman et al. (2013, 2012), Jansson et al. (2012), Kirby (2012), Logg, Ølgaard, Rognes & Wells (2012), Hoffman, Jansson, Jansson & Nazarov (2011), Hoffman, Jansson, Jansson, Johnson & de Abreu (2011). FEniCS-HPC is a branch of the FEniCS Logg, Mardal, Wells et al. (2012), FEniCS (2003) framework focusing on high performance on massively parallel architectures.

Unicorn is solver technology (models, methods, algorithms and software) with the goal of automated high performance simulation of realistic continuum mechanics applications, such as drag or lift computation for fixed or flexible objects (FSI) in turbulent incompressible or compressible flow. The basis for Unicorn is Unified Continuum (UC) modeling Hoffman, Jansson & Stöckli (2011) formulated in Euler (laboratory) coordinates, together with the General Galerkin (G2) adaptive stabilized finite element discretization described above.

The simulations in this chapter were run on supercomputer resources described in the Acknowledgments section, and took ca. 10h on the finest mesh for the whole time interval using ca. 1000 cores.

3 Results

We have performed simulations with the adaptive DFS methodology using the Unicorn/FEniCS-HPC framework for the JSM "pylon on" variant of the HiLiftPW-3 benchmark for the angles 4.36°, 10.58°, 18.58°, 21.57° and 22.58°. All angles except 22.58° have rich experimental data including forces, cp and oil film provided by the workshop, which we will compare against below. The angle 22.58° only has force data. The angles 21.57° and 22.58° exhibit stall in the experiment, e.g. large-scale separation leading to loss of lift force. Capturing stall quantitatively and with the correct stall mechanism is an open problem in aerodynamics, we therefore investigate both the angle 21.57°, which is the highest angle with detailed experimental data, as well as 22.58°,

The experiment is a semispan model at $Re = 1.93M$. However, "free air" computations were requested, and to avoid possible modeling errors introduced by a symmetry plane we model the entire aircraft. However, we choose the output quantity as drag and lift of the left side of the aircraft only, to save computational resources, where we expect the adaptive method to refine in the right half-volume only when there is a significant error contribution to the drag and lift on the left side.

The initial mesh in the adaptive method has ca. 2.5M vertices, and the mesh is then iteratively refined with 5% of the cells in every iteration until we observe mesh convergence in drag and lift, or as many times as we can afford. The finest adapted meshes in our computations presented here have 5M to 10M vertices.

We solve the time-dependent Navier-Stokes equations (1) with a non-dimensional unit inflow velocity over the time interval $I = [0, 10]$. For some of the cases close to stall where we observe a longer startup, we extend the time interval to $I = [0, 20]$. To compute the aerodynamic coefficients we take the mean value in the last quarter of the time interval, e.g. $[7.5, 10]$ or $[15, 20]$, respectively.

We have divided this section into three parts:

1. Detailed comparison of aerodynamic forces against the experiments including convergence of the adaptive method and analysis of stall.
2. Detailed comparison of the pressure coefficients cp against the experimental data, including analysis of cp in the stall regime.
3. Flow visualizations are presented, including dual quantities acting as weights in the error estimates, and comparison of surface velocity against oil film visualizations in the experiment.

3.1 Aerodynamic Forces

The aerodynamic force in the case of zero skin friction as we describe in section 2.5 is computed as:

$$F = \frac{1}{|I|} \int_I \int_{\Gamma_a} p\mathbf{n}\, ds dt, \tag{13}$$

with Γ_a the left half-boundary of the aircraft. The drag and lift coefficients are then simply the x and y components of F since we have unit inflow.

We use the duality-based "do-nothing" adaptive method, which iteratively refines the mesh by repeatedly solving the primal and dual problem based on the a posteriori error estimate. This generates a sequence of adapted meshes, a procedure that takes the role of the classical *mesh study*.

In Figure 2 we plot the lift coefficient, C_l, and drag coefficient, C_d, versus the angle of attack, α, for the different meshes from the iterative adaptive method. The size of the dots indicates the iteration number in the adaptive sequence, with larger dots indicating a larger number, that is more refinement. We connect the finest meshes with lines, and also plot the experimental data as lines. For the angles 18.58° and 22.58° we compute the solution both with and without the "numerical tripping" term described in Section 2.6 to assess the dependence on the angle of attack, the tripped cases are plotted in red, and the adaptive sequence shifter somewhat to the right for clarity.

We observe mesh convergence to within 1 % to 2 % for all cases, a close match to the experiments for C_l, within circa 5 %, and a small overprediction of circa 10 % for C_d, which is consistent with the majority of the participants in HiLiftPW-3 across a range of methods Rumsey (2017), suggesting a systematic error in the problem statement or the experimental data.

For the stall regime angles 18.58°, 21.57° and 22.58° we qualitatively reproduce the stall phenomenon in the experiment – a decrease in C_l with increased angle of attack past 21.57°. We observe that the stall angle occurs somewhere between 18.58°, 21.57° which is ca. 1° from the experimental stall angle.

Additionally we verify that the "numerical tripping" functions as expected: the term has no significant impact on the solution for an angle of 18.58°, which is the maximum lift angle and the maximum non-stalling angle, whereas for the stalling

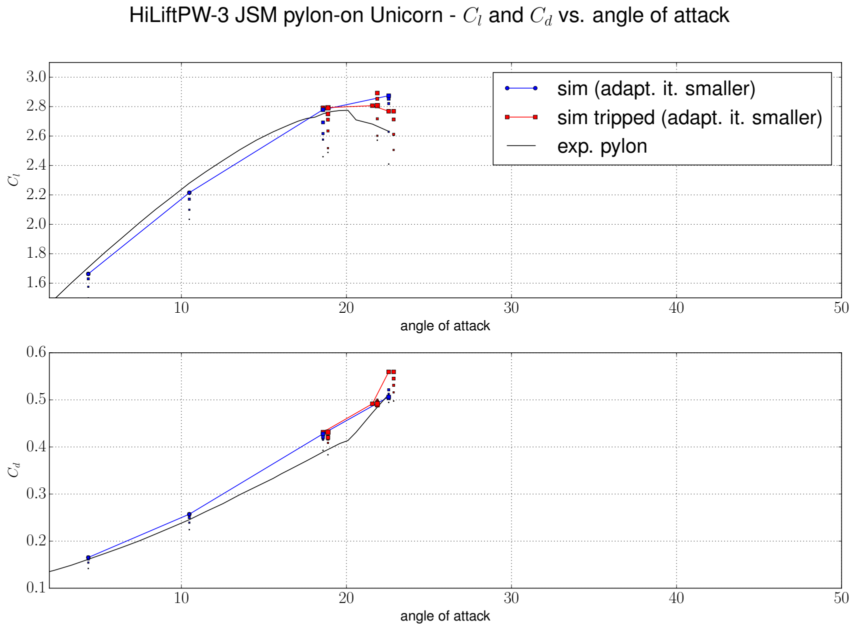


Fig. 2 Lift coefficient, C_l, and drag coefficient, C_d, versus the angle of attack, α, for the different meshes from the iterative adaptive method.

angle 22.58° we observe that the tripping has the effect of triggering a large-scale separation consistent with the stall phenomenon, whereas the untripped case appears to contain too small perturbations for the separation to occur. We analyze the stall mechanism in more detail in the surface velocity visualization below.

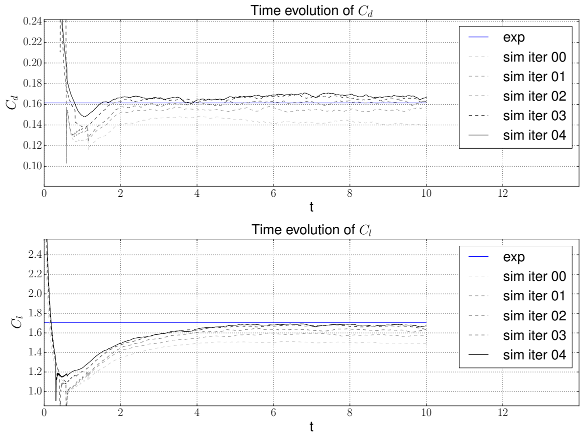
To analyze the variability in time of C_d and C_l we plot the time evolution for $\alpha = 4.36°$ in Figure 3, untripped with $\alpha = 18.58°$ in Figure 4 and tripped with $\alpha = 18.58°$ in Figure 5.

For the pre-stall cases we observe an initial "startup phase" for $t \in [0, 5]$ and then an oscillation around a stable mean value. The effect of the numerical tripping is noise in the C_d and C_l signals with amplitude of about 1 %.

3.2 Pressure coefficients

The pressure coefficients C_p from both simulation on the finest adaptive mesh and experiments are plotted in Figures 7, 8 and 9, for the wing, flap and slat respectively.

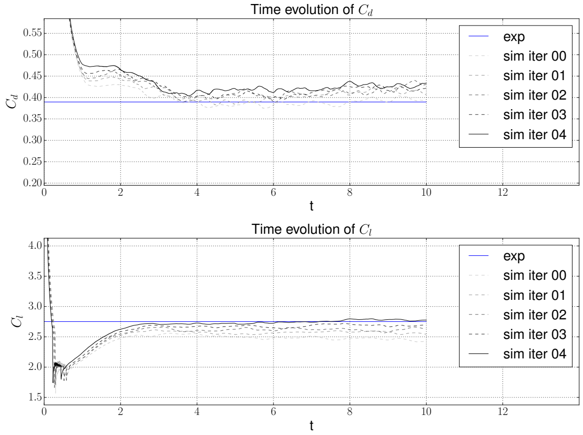
The pressure sensor locations corresponding to the plots are specified in the diagram in Figure 6.



	sim.	exp.	rel. err.
C_l	1.680	1.71	1.6%
C_d	0.168	0.161	3.9%

(a) $\alpha = 4.36^\circ$

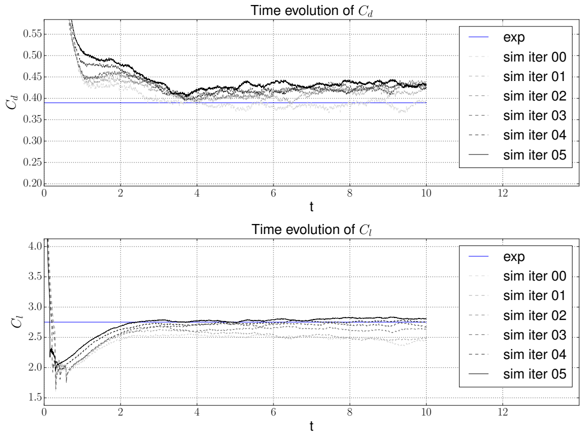
Fig. 3 Time evolution of lift coefficient, C_l, and drag coefficient, C_d, and a table of the value for the finest adaptive mesh with relative error compared to the experimental results for $\alpha = 4.36^\circ$.



	sim.	exp.	rel. err.
C_l	2.78	2.75	1.0%
C_d	0.428	0.390	9.0%

(a) $\alpha = 18.58^\circ$

Fig. 4 Time evolution of lift coefficient, C_l, and drag coefficient, C_d, and a table of the value for the finest adaptive mesh with relative error compared to the experimental results for $\alpha = 18.58^\circ$, untripped.



	sim.	exp.	rel. err.
C_l	2.80	2.75	1.6%
C_d	0.432	0.390	9.8%

(a) $\alpha = 18.58^\circ$

Fig. 5 Time evolution of lift coefficient, C_l, and drag coefficient, C_d, and a table of the value for the finest adaptive mesh with relative error compared to the experimental for $\alpha = 18.58^\circ$ with numerical tripping.

Since the aerodynamic force defined in (13) matches the experiment well, and since it consists of integrals of the pressure weighed by the normal vector, the C_p values also have to match the experiment on average. However, the C_p plots can give insight into local mechanisms such as separation patterns, an important example being the stall mechanism. These local mechanisms are what we will focus on here.

First of all, we see that for the pre-stall angles $\alpha = 10.48^\circ$ and $\alpha = 18.58^\circ$ the simulation and experiment match very well for the wing and slat, and generally well for the flap, aside from local differences. The C_p for the simulation is lower on the upper surface for the flap close to the body (the A-A station). Otherwise the curves generally match.

For the stall regime we analyze both 21.57° where experimental C_p are available and 22.56° where experimental C_p plots are not available. We compare both against the experimental C_p plots for 21.57° to have a margin for if we have stall at a higher angle in the simulation. The simulation matches the experiment very well, there is a small discrepancy for the wing close to the body (the A-A station), but considering that this is where the large-scale separation causing the stall is located, the results match acceptably.

The matching C_p curves are consistent with matching C_d and C_l from the aerodynamic force plots.

We now compare the tripped and untripped simulation with the experiment at 22.56°, as well as 22.56° in Figure 10 for the wing. We clearly see that the untripped simulation for 22.56° grossly misses the C_p on the upper surface at station A-A, near the wing-body junction where the large-scale separation mechanism causing stall is located, while the tripped simulation captures the experimental C_p curve well, aside from a slightly lower C_p near the leading edge. We conclude that the tripping acts

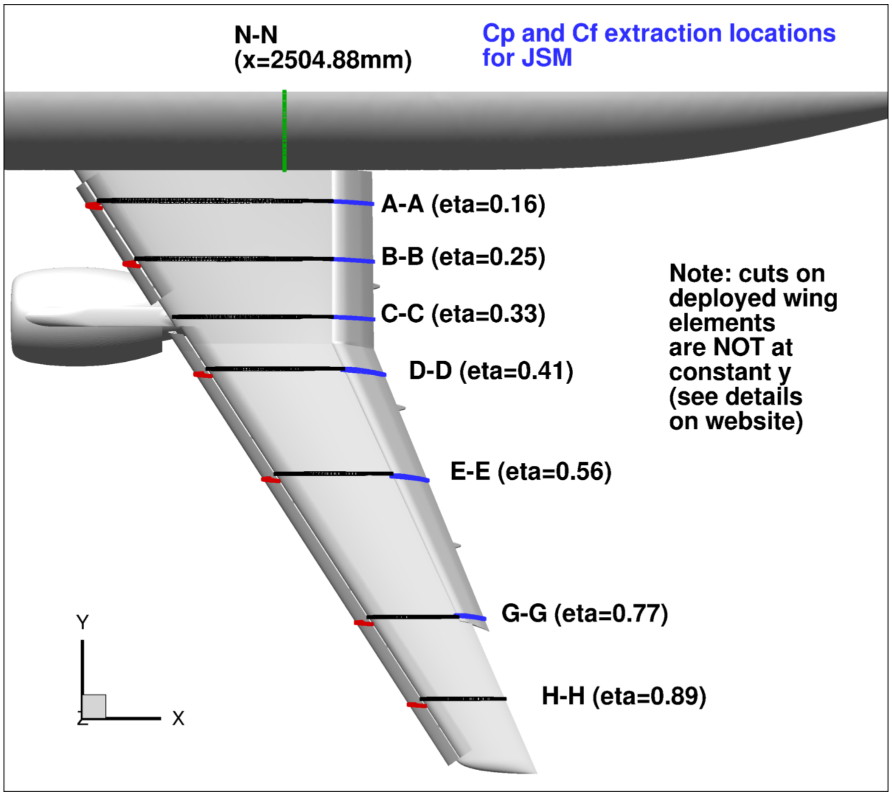


Fig. 6 Diagram of the pressure sensor layout for the JSM configuration showing where the pressure sensors are located and how they are denoted.

to trigger the physically correct separation. At the other stations, D-D and G-G, the tripped and untripped simulations are very similar, indicating that the tripping does not have a significant effect aside from the triggering of the perturbations.

The $\alpha = 21.57^\circ$ simulation is tripped and captures the experiment less well than 22.56°, but better than 22.56° untripped indicating that we may have a ca. 1° later stall angle in the simulation than in the experiment.

3.3 *Flow and Adaptive Mesh Refinement Visualization*

Here we concentrate on presenting effective visualization of the flow and the adaptive mesh refinement procedure. Our aim is to provide information on the properties and features of the approximated solution and, more importantly, of the approximating procedure, most of which cannot be discerned from one dimensional plots of the

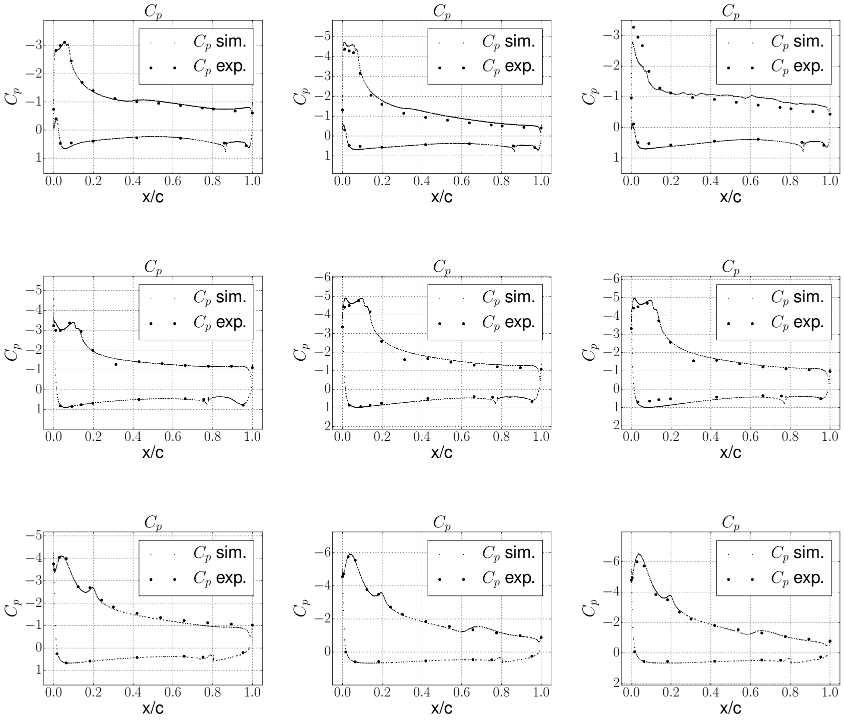


Fig. 7 Pressure coefficients, C_p, versus normalized local chord, x/c, for the angles of attack $\alpha = 10.48^\circ$ (left), $\alpha = 18.58^\circ$ (middle) and $\alpha = 22.56^\circ$ (right) at locations A-A (top), D-D (middle) and G-G (bottom) for the wing of JSM pylon on.

pressure coefficient and the aerodynamic forces. Sometimes these more complex visualizations cannot be directly compared to experiments, but still they constitute a qualitative validation of the results.

The first plots that we show are the surface plots of the velocity magnitude on the upper side of the wing. Together with the velocity magnitude surface plots we also report pictures of the oil film experiment that was provided by the organizers as a validation tool. These serve as comparison tools, and we report such comparison in Figure 11.

Some common features intrinsic of the geometry of the JSM aircraft are revealed by the oil film experiment and reproduced by the velocity plots. A pattern of low velocity streaks, alternating with areas of higher velocity, is seen on the suction side of the fixed wing for all angles of attack. This is caused by separation at the slat tracks upstream, which is correctly captured by the numerical solution.

Another characteristic feature of the flow is the turbulent separation near the tip of the wing. This is particularly evident in the case $\alpha = 18.59°$,

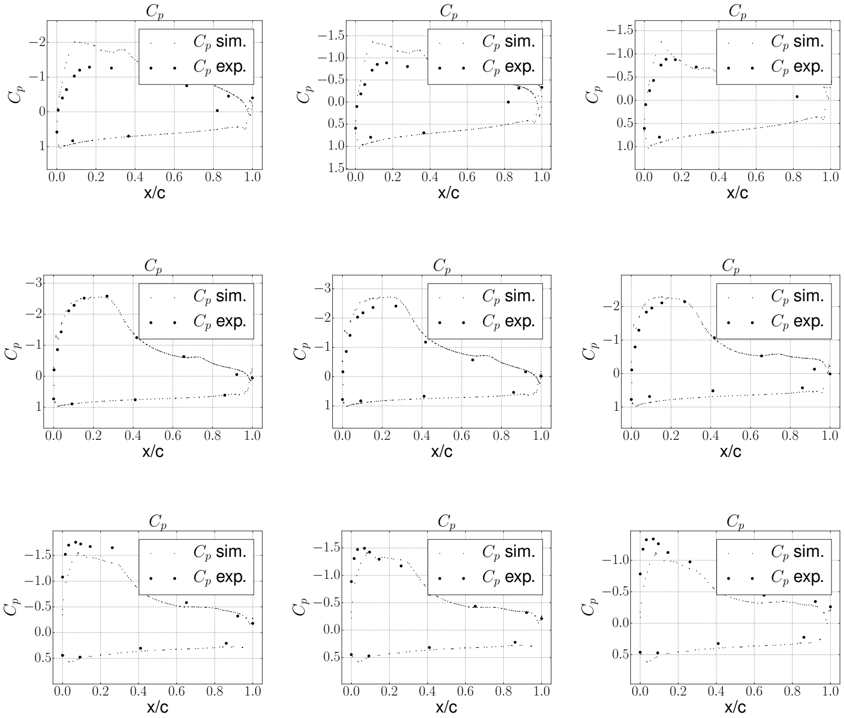


Fig. 8 Pressure coefficients, C_p, versus normalized local chord, x/c, for the angles of attack $\alpha = 10.48°$ (left), $\alpha = 18.58°$ (middle) and $\alpha = 22.56°$ (right) at locations A-A (top), D-D (middle) and G-G (bottom) for the flap of JSM pylon on.

Areas that exhibit this kind of flow behavior influence the aerodynamic forces on the aircraft, and indeed in our experimentation we found that computations done on some meshes resulted in wrong predictions of the target functionals, usually yielding lower lift coefficients than the experimental ones. We were able to overcome this intermediate obstacle by refining the surface mesh were the original geometry had a higher curvature. We later interpreted the effectiveness of this workaround as a symptom that the original meshes were unable to capture the surface geometry to a sufficient degree of accuracy, and were for this reason failing at reproducing these complex patterns.

Another interesting visualization technique, which we are about to present, is more closely related to turbulence itself: the Q-criterion Hunt et al. (1988). The Q-criterion was widely used in the literature to visualize turbulent features of fluid flows. The main idea is that it is possible to define a quantity, commonly denoted by the letter Q, whose value is related to the vorticity and thus the visualization of the

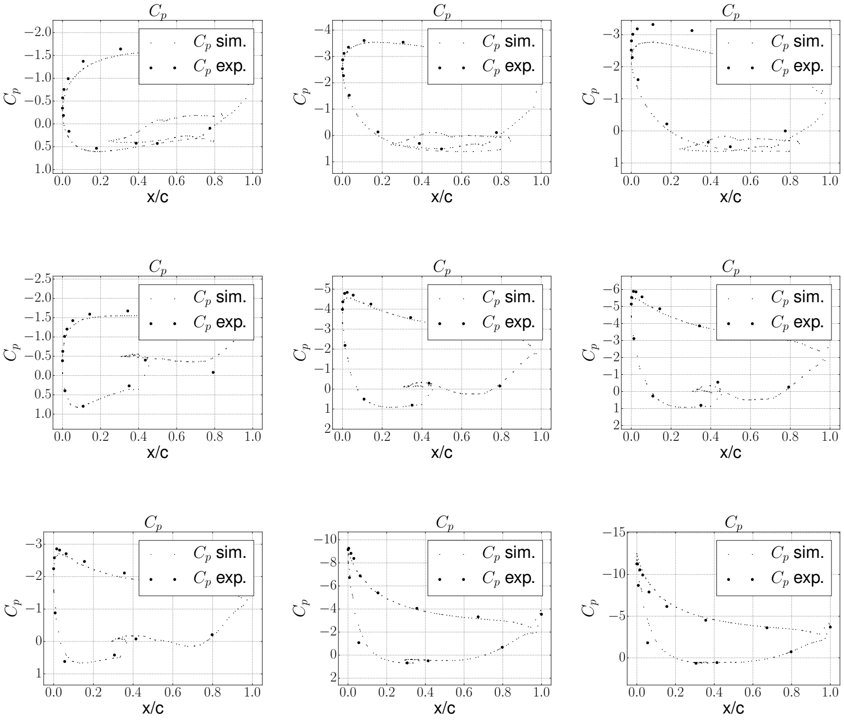


Fig. 9 Pressure coefficients, C_p, versus normalized local chord, x/c, in the stall regime for the angles of attack $\alpha = 10.48°$ (left), $\alpha = 18.58°$ (middle) and $\alpha = 22.56°$ (right) at locations A-A (top), D-D (middle) and G-G (bottom) for the slat of JSM pylon on.

isocontours of Q is claimed to give visual information on the presence and location of vortexes within the flow field.

The Q-criterion for the case of the airplane with pylon is displayed in Figure 12 for three different angles of attack.

Once again, the visualization technique highlights the same pattern as in the previous case: the isosurfaces assume a characteristic V shape along the interfaces between the fast and slow velocity regions on the suction side of the wing. Not only that, but we can also clearly distinguish a clustering of these isosurfaces near the tip of the wing, matching the position of the turbulent separation zone that we mentioned above. The Q-criterion visualizations are consistent with the surface velocity plots, and this internal coherence increases our trust in the computational results.

Let us now turn our attention to the adaptive procedure which produces the successive approximations of the fluid flow. As we described above, the mesh refinement solution is driven by the residual of the Navier-Stokes equations and the solu-

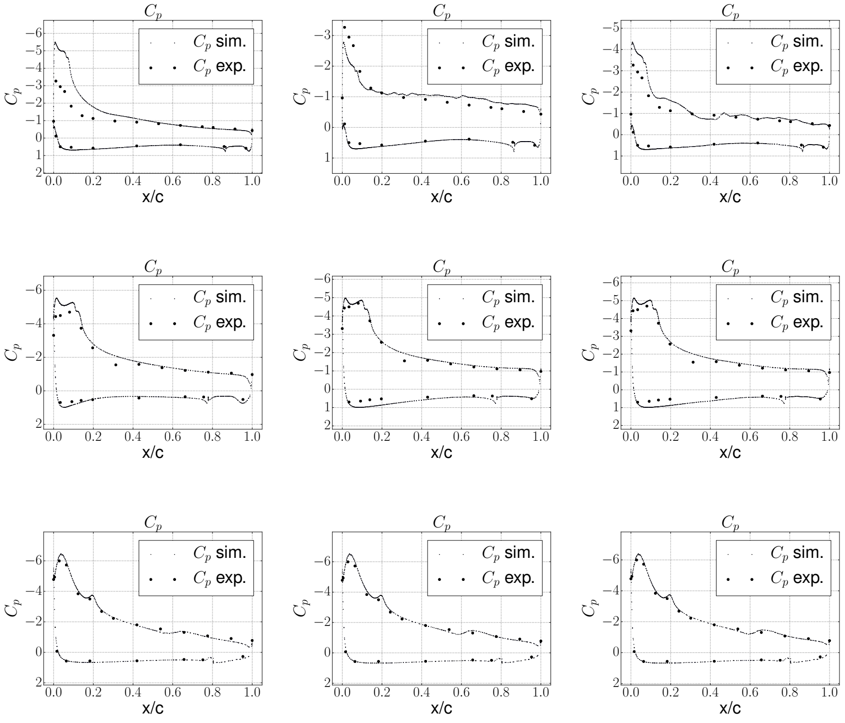
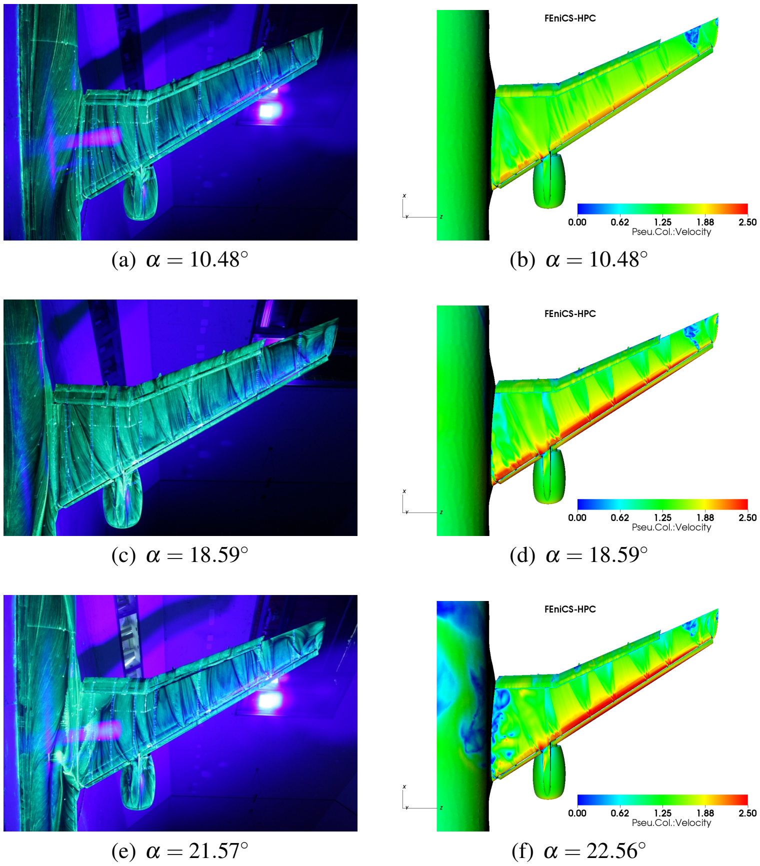


Fig. 10 Pressure coefficients, C_p, versus normalized local chord, x/c, for the angle of attack $\alpha = 22.56^\circ$ untripped (left), the same angle $\alpha = 22.56^\circ$ tripped (middle) and $\alpha = 21.57^\circ$ tripped at locations A-A (top), D-D (middle) and G-G (bottom) for the wing of JSM pylon on.

tion of the dual Navier-Stokes equations. We begin by showing a plot of a volume rendering of the dual solution, see Figure 13.

What is worth noting here is that the adjoint velocity flows backwards in time and, consequently, it appears to be flowing in the opposite direction of the primal velocity. We observe that the part of the mesh where the dual velocity has higher values is *upstream* to the airplane. Because of the way the do-nothing error estimator is designed, we expect that the refinement will happen where both the residual and the dual solution are large. Indeed, this has the important implication that the mesh refinement will not only happen on the wing, where the forces are computed, but also upstream, splitting cells that, a priori, are unrelated to the computation of the aerodynamic forces.

This feature is unique for our methodology: while other methods tend to refine the mesh in zones where *intuitively* higher accuracy would yield better approximation of the aerodynamic forces, namely around the body and downstream, the adaptive algorithm provides an automatic procedure that knows nothing about the



(a) $\alpha = 10.48°$

(b) $\alpha = 10.48°$

(c) $\alpha = 18.59°$

(d) $\alpha = 18.59°$

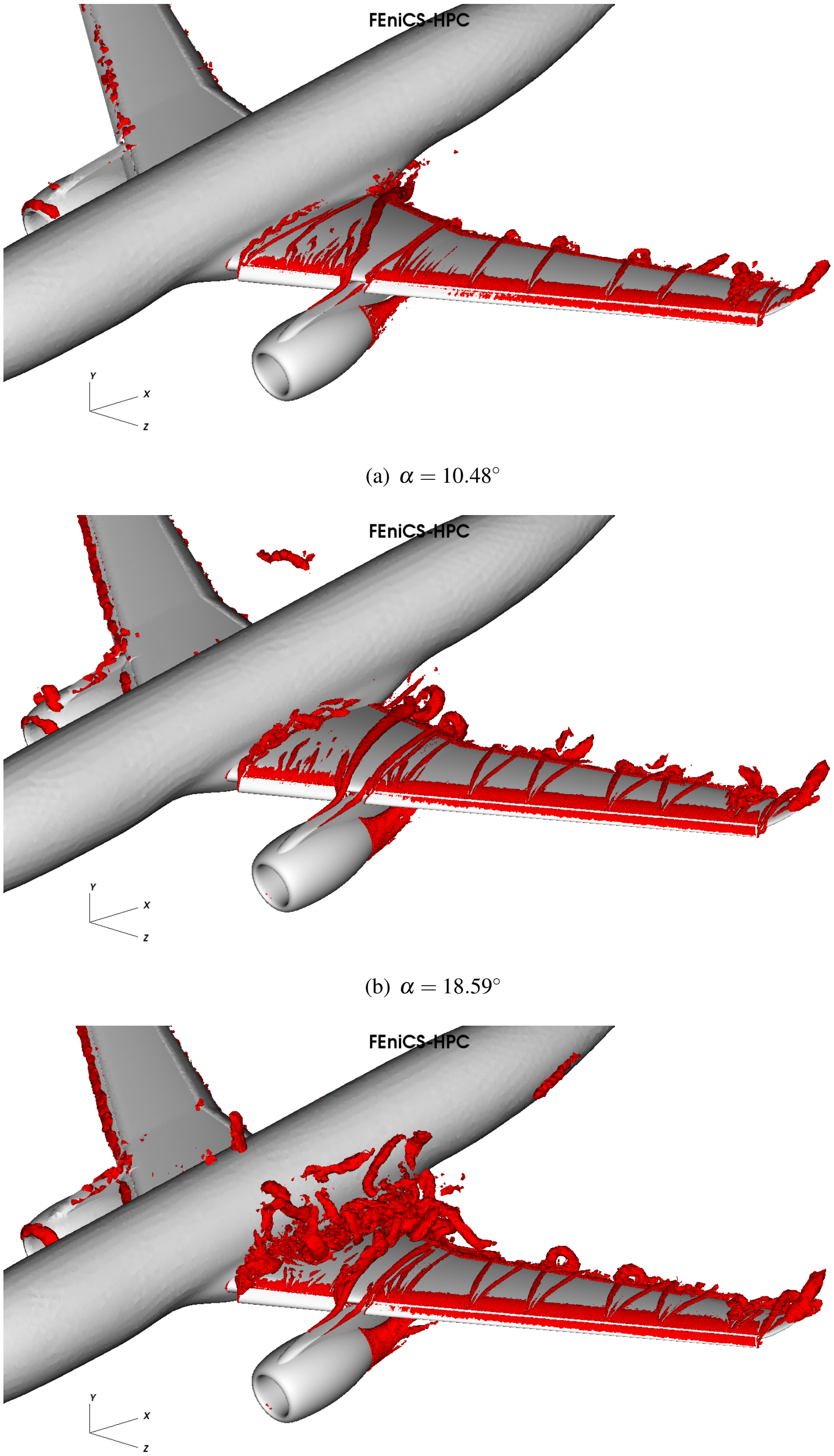
(e) $\alpha = 21.57°$

(f) $\alpha = 22.56°$

Fig. 11 Comparison between experimental oil film visualization (left) and surface rendering of the velocity magnitude (right).

features of the flow but only takes into account the residual of the equations of motion and the solution of the dual problem.

In our numerical experimentation we found that this is exactly what happens, as we are about to show. Consider Figure 14, showing a crinkled slice of the mesh for the initial and the finest meshes for a given angle of attack. It is clear that the mesh refinement procedure is concentrating both on the area around the surface where the aerodynamic forces are computed and in the upstream region. Some cells are refined downstream due to the large residual.



(a) $\alpha = 10.48°$

(b) $\alpha = 18.59°$

(c) $\alpha = 22.56°$

Fig. 12 Instantaneous isosurface rendering at the final time of the Q-criterion with value $Q = 100$.

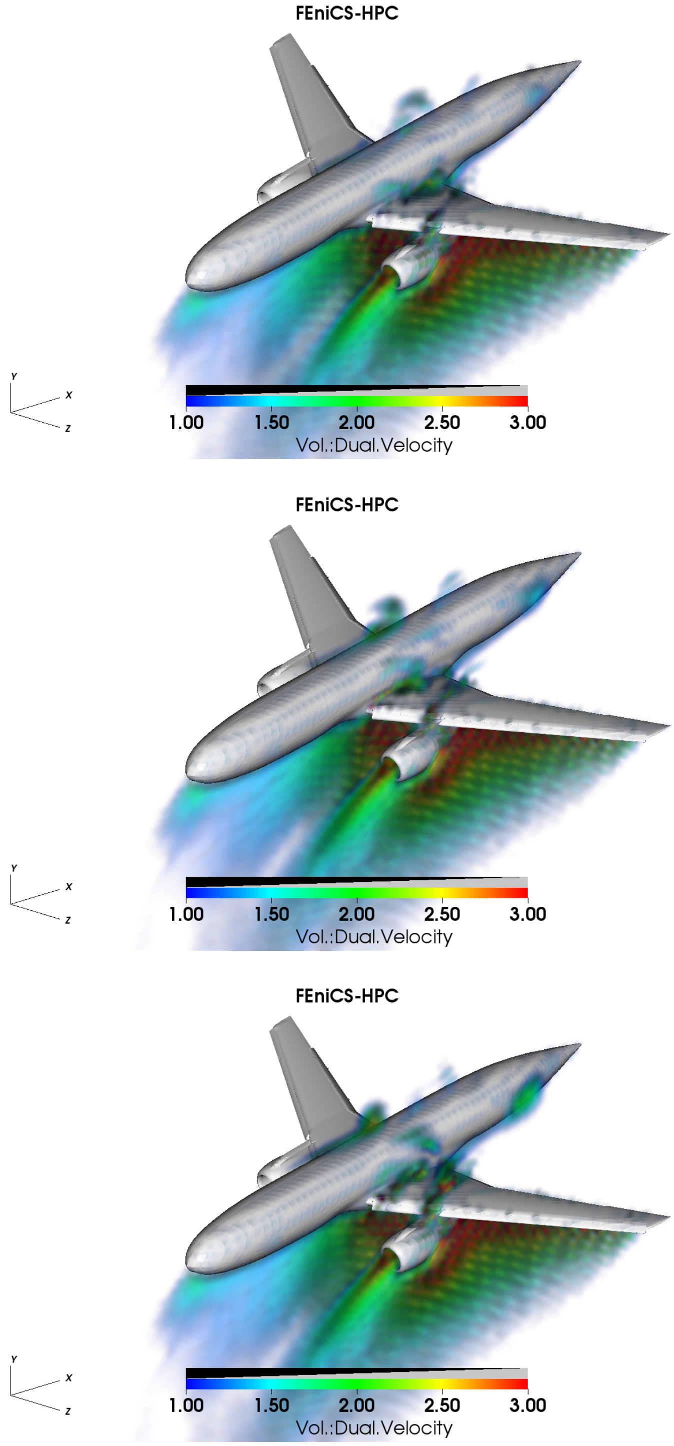
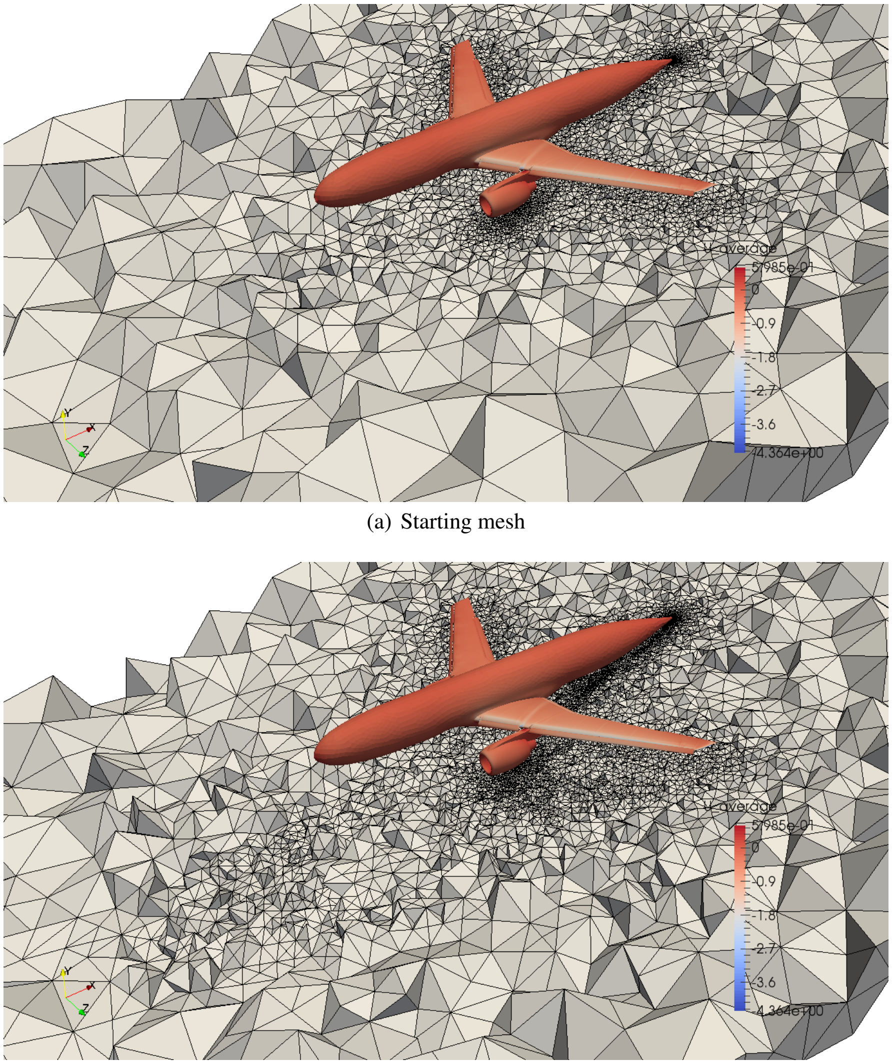


Fig. 13 Volume rendering of the time evolution of the magnitude of the adjoint velocity φ magnitude, snapshots at t = (16, 18, 20).



(a) Starting mesh

(b) Finest adaptive mesh

Fig. 14 Crinkled slice aligned with the angle of attack, $\alpha = 10.48°$

4 Conclusions

This chapter presents an adaptive finite element method without turbulence model parameters for time-dependent aerodynamics, and we validate the method by simulation results of a full aircraft model originating from the 3$^{\text{rd}}$ AIAA CFD High-Lift Prediction Workshop (HiLiftPW-3) which was held in Denver, Colorado, on June 3$^{\text{rd}}$-4$^{\text{th}}$ 2017. The mesh is automatically constructed by the method as part of the computation through duality-based a posterori error control and no explicit turbulence model is used. Dissipation of turbulent kinetic energy in under-resolved parts of the flow is provided by the numerical stabilization in the form of a weighted least squares method based on the residual of the NSE. Thus, the method is purely based on the NSE mathematical model, and no other modeling assumptions are made.

The DFS method and these simulations are thus *parameter-free*, where no a priori knowledge of the flow is needed during the problem formulation stage, nor during the mesh generation process. Additionally, the computational cost is drastically reduced by modeling turbulent boundary layers in the form of a slip boundary condition, and thus no boundary layer mesh is needed.

The computed aerodynamic coefficients, are very close to the experimental values for all the angles of attack that we studied. In particular, C_l is within circa 5 % of the experiments, C_d has a small overprediction of circa 10 %, which is consistent with the majority of the participants in HiLiftPW-3 across a range of methods Rumsey (2017), suggesting a systematic error in the problem statement or the experimental data.

The fact that the error is automatically estimated by the method is itself a critical feature missing in most (if not all) other computational frameworks for CFD.

Moreover, the adaptive procedure in DFS is seen to converge to a mean value with oscillations of the order of 1 % to 2 %. This contributes to increase the confidence in the numerical method.

The point of adaptive computations is all about saving on the computational cost. During the workshop we had the chance to compare our performance with that of the other participating groups. In terms of number of degrees of freedom, DFS is about ten times cheaper than the leading RANS and Lattice Boltzmann Methods.

To capture stall, we applied a tripping noise term that turned out to have the effect of triggering the physically correct stall separation pattern. A similar idea with a noise term is employed in the DNS community as well, and the addition of this term seems to have no effect on non-stalling configurations, which is an important validation.

We observed that DFS was able to capture the stall mechanism of the proposed configuration, namely the large scale separation pattern that occurs at the wing-body juncture. The same mechanism is observed in the experiments. The stall angle is also captured within ca. 1 °.

Acknowledgments.

This research has been supported by the European Research Council, the H2020 MSO4SC grant, the Swedish Research Council, the Swedish Foundation for Strategic Research, the Swedish Energy Agency, the Basque Excellence Research Center (BERC 2014-2017) program and ELKARTEK GENTALVE project by the Basque Government, the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa accreditation SEV-2013-0323 and the Project of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824 and La

Caixa. We acknowledge the Swedish National Infrastructure for Computing (SNIC) at PDC – Center for High-Performance Computing for awarding us access to the supercomputer resources Beskow.

References

FEniCS (2003), 'Fenics project', `http://www.fenicsproject.org`.

Hoffman, J. (2005), 'Computation of mean drag for bluff body problems using adaptive dns/les', *SIAM J. Sci. Comput.* **27(1)**, 184–207.

Hoffman, J. (2006), 'Adaptive simulation of the turbulent flow past a sphere', *J. Fluid Mech.* **568**, 77–88.

Hoffman, J. (2009), 'Efficient computation of mean drag for the subcritical flow past a circular cylinder using general galerkin g2', *Int. J. Numer. Meth. Fluids* **59(11)**, 1241–1258.

Hoffman, J., Jansson, J., de Abreu, R. V., Degirmenci, N. C., Jansson, N., Müller, K., Nazarov, M. & Spühler, J. H. (2013), 'Unicorn: Parallel adaptive finite element simulation of turbulent flow and fluid-structure interaction for deforming domains and complex geometry', *Comput. Fluids* **80**(0), 310 – 319.

Hoffman, J., Jansson, J., Degirmenci, C., Jansson, N. & Nazarov, M. (2012), *Unicorn: a Unified Continuum Mechanics Solver*, Springer, chapter 18.

Hoffman, J., Jansson, J. & Jansson, N. (2015), 'Fenics-hpc: Automated predictive high-performance finite element computing with applications in aerodynamics', *Proceedings of the 11th International Conference on Parallel Processing and Applied Mathematics, PPAM 2015. Lecture Notes in Computer Science* .

Hoffman, J., Jansson, J., Jansson, N. & Abreu, R. V. D. (2015), 'Towards a parameter-free method for high reynolds number turbulent flow simulation based on adaptive finite element approximation', *Computer Methods in Applied Mechanics and Engineering* **288**(0), 60 – 74.
URL: *http://www.sciencedirect.com/science/article/pii/S0045782514004836*

Hoffman, J., Jansson, J., Jansson, N., Johnson, C. & de Abreu, R. V. (2011), Turbulent flow and fluid-structure interaction, *in* 'Automated Solutions of Differential Equations by the Finite Element Method', Springer.
URL: *http://www.fenicsproject.org/pub/documents/book/*

Hoffman, J., Jansson, J., Jansson, N. & Nazarov, M. (2011), Unicorn: A unified continuum mechanics solver, *in* 'Automated Solutions of Differential Equations by the Finite Element Method', Springer.
URL: *http://www.fenicsproject.org/pub/documents/book/*

Hoffman, J., Jansson, J., Jansson, N., Vilela De Abreu, R. & Johnson, C. (2016), 'Computability and adaptivity in cfd. encyclopedia of computational mechanics, stein, e., de horz, r. and hughes, tjr eds'.

Hoffman, J., Jansson, J. & Stöckli, M. (2011), 'Unified continuum modeling of fluid-structure interaction', *Mathematical Models and Methods in Applied Sciences* .

Hoffman, J. & Jansson, N. (2010), *A computational study of turbulent flow separation for a circular cylinder using skin friction boundary conditions*, Ercoftac, series Vol.16, Springer.

Hoffman, J. & Johnson, C. (2006*a*), *Computational Turbulent Incompressible Flow: Applied Mathematics Body and Soul Vol 4*, Springer-Verlag Publishing.

Hoffman, J. & Johnson, C. (2006*b*), 'A new approach to computational turbulence modeling', *Comput. Methods Appl. Mech. Engrg.* **195**, 2865–2880.

Hoffman, J. & Johnson, C. (2007), *Computational Turbulent Incompressible Flow*, Vol. 4 of *Applied Mathematics: Body and Soul*, Springer.

Hoffman, J. & Johnson, C. (Published Online First at www.springerlink.com: 10 December 2008), 'Resolution of d'alembert's paradox', *J. Math. Fluid Mech.* .

Houzeaux, G., Vázquez, M., Aubry, R. & Cela, J. (2009), 'A massively parallel fractional step solver for incompressible flows', *Journal of Computational Physics* **228**(17), 6316–6332.

Huang, L., Huang, P. G. & LeBeau, R. P. (2004), 'Numerical study of blowing and suction control mechanism on naca 0012 airfoil', *AIAA Journal of aircraft* **41**(1).

Hunt, J. C., Wray, A. A. & Moin, P. (1988), 'Eddies, streams, and convergence zones in turbulent flows'.

Jansson, N., Hoffman, J. & Jansson, J. (2012), 'Framework for Massively Parallel Adaptive Finite Element Computational Fluid Dynamics on Tetrahedral Meshes', *SIAM J. Sci. Comput.* **34**(1), C24–C41.

Kirby, R. C. (2012), *FIAT: Numerical Construction of Finite Element Basis Functions,*, Springer, chapter 13.

Logg, A., Mardal, K.-A., Wells, G. N. et al. (2012), *Automated Solution of Differential Equations by the Finite Element Method*, Springer.

Logg, A., Ølgaard, K. B., Rognes, M. E. & Wells, G. N. (2012), *FFC: the FEniCS Form Compiler*, Springer, chapter 11.

Mellen, C. P., Frölich, J. & Rodi, W. (2003), 'Lessons from lesfoil project on large-eddy simulation of flow around an airfoil', *AIAA journal* **41**, 573–581.

Moin, P. & You, D. (2008), 'Active control of flow separation over an airfoil using synthetic jets', *Journal of Fluids and Structures* **24**(8), 1349–1357.

Piomelli, U. & Balaras, E. (2002), 'Wall-layer models for large-eddy simulation', *Annu. Rev. Fluid Mech.* **34**, 349–374.

Rumsey, C. (2017), '3^{rd} AIAA CFD High Lift Prediction Workshop (HiLiftPW-2) (http://hiliftpw.larc.nasa.gov/)'.
URL: *http://hiliftpw.larc.nasa.gov/*

Sagaut, P. (2005), *Large Eddy Simulation for Incompressible Flows (3rd Ed.)*, Springer-Verlag, Berlin, Heidelberg, New York.

Schlatter, P. & Orlu, R. (2012), 'Turbulent boundary layers at moderate reynolds numbers: inflow length and tripping effects', *Journal of Fluid Mechanics* **710**, 534.

Shan, H., Jiang, L. & Liu, C. (2005), 'Direct numerical simulation of flow separation around a naca 0012 airfoil', *Computers and Fluids* **34**, 10961114.

Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E. & Mavriplis, D. (2014), 'Cfd vision 2030 study: a path to revolutionary computational aerosciences'.

Spalart, P. R. (2009), 'Detached-eddy simulation', *Annu Rev. Fluid Mech.* **41**, 181–202.

Vilela de Abreu, R., Jansson, N. & Hoffman, J. (2014), 'Adaptive computation of aeroacoustic sources for a rudimentary landing gear', *Int. J. Numer. Meth. Fluids* **74**(6), 406–421.
URL: *http://dx.doi.org/10.1002/fld.3856*

Witherden, F. D. & Jameson, A. (2017), Future directions of computational fluid dynamics, *in* '23rd AIAA Computational Fluid Dynamics Conference', p. 3791.