Mixed Fidelity Aerodynamic and Aero-Structural Optimization for Wings

Mengmeng Zhang, Tomas Melin
Airinnoa AB
SE-182 48, Stockholm, Sweden
[mengmeng.zhang,tomas.melin]@airinnova.se

Jing Gong, Michaela Barth, Lilit Axner
PDC Center for High Performance Computing
KTH Royal Institute of Technology
SE-100 44, Stockholm, Sweden
[gongjing,caela,lilit]@kth.se

Abstract— **Automatic multidisciplinary design optimization is one of the challenges that are faced in the processes involved in designing efficient wings for aircraft. In this paper we present mixed fidelity aerodynamic and aero-structural optimization methods for designing wings. A novel shape design methodology has been developed – it is based on a mix of the automatic aerodynamic optimization for a reference aircraft model, and the aero-structural optimization for an uninhabited air vehicle (UAV) with a high aspect ratio wing. This paper is a significant step towards making it possible to perform all the core processes for aerodynamic and aero-structural optimization that require special skills in a fully automatic manner – this covers all the processes from creating the mesh for the wing simulation to executing the high-fidelity computational fluid dynamics (CFD) analysis code. Our results confirm that the simulation tools can make it possible for a far broader range of engineering researchers and developers to design aircraft in much simpler and more efficient ways. This is a vital step in the evolution of wing design processes as it means that the extremely expensive laboratory experiments that were traditionally used when designing the wings can now be replaced with more cost effective high performance computing (HPC) simulation that utilize accurate numerical methods.**

Keywords: Multidisciplinary design optimization (MDO), Computational fluid dynamics (CFD), High performance computing

I. INTRODUCTION

Designing more efficient wings is one of the cornerstones of aircraft manufacture and the details of the procedures that are used for it are the well-guarded intellectual property of the aircraft manufacturers. The ultimate goal of the wing design process is to design lighter, more fuel efficient and quieter airplanes by reducing the drag, especially at high speeds. Shape optimization is an important task within the wing design process in industrial research. Consequently, multidisciplinary design optimization (MDO) is a key factor when it comes to optimizing the shape of aircraft wings, see Fig. 1.

Usually the sub-tasks within the MDO process can be split into aerodynamic and aero-structural optimizations. Aerodynamic optimization problems can be set-up in different ways within the industrial aviation design process. For example, to set up design parameters for a wing could set up to facilitate the optimization work either using free-form deformation box control points around a wing, or using the standard shape parameters of an aircraft planform as well as the twist, camber and thickness for a wing. When searching for more efficient wing designs, it is important to investigate many different candidate shapes because we do not just want to find the solutions that are close to the baseline shape. The creation of those alternative candidate shapes requires different methods that involve computationally expensive fluid dynamics analysis. In addition, high fidelity computational fluid dynamics (CFD) simulations, such as Reynolds-averaged Navier-Stokes (RANS) solvers with turbulence models, must be parallelized to reduce the total run-time when using high performance computing (HPC) resources.

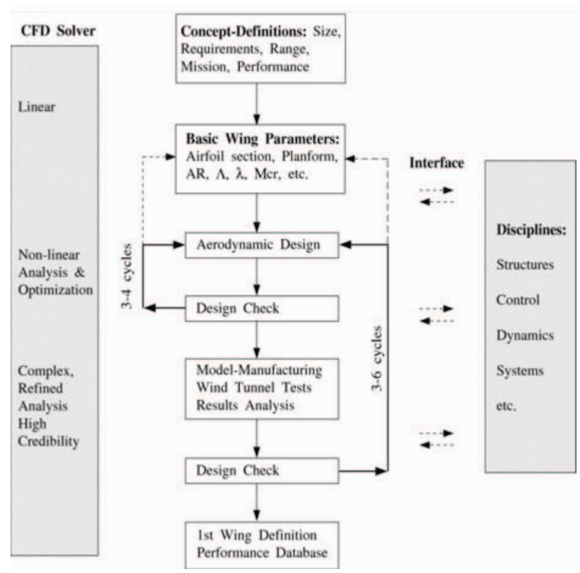

Figure 1 Wing Design Process in industry

When it comes to the other aspect of the sub-task in the MDO process, namely the aero-structural optimization, the computational solution of the relevant aeroelastic problems involves unsteady aerodynamic-structural coupled simulations, where low-fidelity methods based on experience and statistical data may be employed. We have developed advanced computational technology solutions for cutting-edge aircraft preliminary design and multi-disciplinary optimization for aerodynamics [1-2]. Within these research projects, high

978-1-5386-7879-4/18/$31.00 ©2018 IEEE
DOI 10.1109/HPCS.2018.00081

fidelity CFD analysis was employed in the automatic workflows as a major tool for establishing the aircraft geometry defined by computer-aided design (CAD) files then generating meshes, and finally performing CFD analyses with the selected turbulence models.

In this paper, we focus on blending the automated aerodynamic optimization for a reference aircraft model, the common research model (CRM) aircraft released by NASA [3], and the aero-structural optimization for the uninhabited air vehicle (UAV) with a high aspect ratio (AR) wing from the AGILE project [4] in CPACS format [5]. The research in this paper addresses the aerodynamic shape optimization for wings using mixed fidelity tools (especial the high-fidelity tool) [6], and performs the aero-structural optimization using a simple low-fidelity linearized model.

The paper is divided into two main sections. The high-fidelity aerodynamic optimization for a CRM model is presented in Section II, which is followed by the low-fidelity aero-structural optimization for the high-AR wing model in Section III. The discussion of the results and the conclusions are contained in the final section.

II. AERODYNAMIC OPTIMIZATION FOR THE CRM MODEL

Over the past three years, the Aerodynamic Design Optimization Discussion Group (ADODG), which is part of the American Institute of Aeronautics and Astronautics (AIAA), has proposed a basic set of benchmark test cases to the scientific research community. These test cases are suitable for exercising aerodynamic optimization methods in a constrained design space in order to highlight and compare the specific contributions of different approaches to the benchmark suite [7-8]. Computed and measured data for the CRM produced with the same Mach numbers and models has been made available for benchmarking tests with CFD calculations. The CRM is representative of a contemporary transonic commercial aircraft, of a size similar to that of a Boeing 777. However, the CRM has 3.5 degrees more quarter chord wing sweep and 10.3 m^2 more wing area than the Boeing 777-200. The geometry of the CRM has been optimized in aerodynamic performances.

A. Optimization strategies taken

The optimization objective is to minimize the drag coefficient C_D with the lift coefficient constrained to $C_L = 0.5$, and the pitching-moment coefficient constrained to $C_M \geq -0.17$. The flow is fully turbulent at a transonic speed with Mach number $M_\infty = 0.85$, a chord Reynolds number $Re = 5 \cdot 10^6$, and an initial angle of attack $\alpha = 2.2^\circ$. The geometric constraints also apply for this case. The baseline geometry is a wing-only geometry with a blunt trailing edge extracted from the CRM wing-body configuration, which was the subject of the Fifth Drag Prediction Workshop. A CAD file in IGES format was provided by the ADODG Group to make sure that the baseline is identical for every participant. The optimization problem is written as follows:

$$\min: J = \sum_{i=1}^{3} \mathcal{T}_i \, C_{D_i}(\vec{p}, u(\vec{p})) \tag{1}$$

subject to:

$$C_{L_i}(\vec{p}, u(\vec{p})) = (C_{L_i})_{prescribed},$$

$$C_M(\vec{p}, u(\vec{p})) \geq -0.17,$$

$$g_j(\vec{p}) \geq 25\% \cdot g_{j,0}(\vec{p}), \quad 1 \leq j \leq m,$$

$$\mathcal{V} \geq \mathcal{V}_0$$

where $\mathcal{V}$ is the internal volume of the wing, and g is the thickness of the wing at arbitrary stations (j). The wing shape parameters are denoted by $\vec{p}$, and the state variables by $u(\vec{p})$. In Eq. (1) the flow condition is transonic at $M_\infty = 0.85$ and $Re = 5 \cdot 10^6$ based on the mean aerodynamic chord. C_D, C_L and C_M are the drag, lift and pitch moment coefficients respectively. The design variables are the twist distributions together with the profiles of each wing station along the wing. The optimization is considered to have converged when the SNOPT optimality condition reaches 10^6 while the design satisfies the lift constraint. We have the mesh generation algorithm $M(\vec{X}, \overrightarrow{X_\Gamma}) = 0$ and the surface parametrization algorithm S is constrained by $S(\overrightarrow{X_\Gamma}, \vec{p}) = 0$ where $\vec{p}$ are the design variables which determine the surface $\overrightarrow{X_\Gamma}$.

The change in J can be estimated by making a small variation $\delta\vec{p}$ to the parameter vector and recalculating the flow to obtain the change in J, thus approximating the directional derivative,

$$J(\vec{p} + \delta\vec{p}) = J(\vec{p}) + \frac{dJ}{d\vec{p}} \cdot \delta\vec{p} + O(\|\delta\vec{p}\|^2) \tag{2}$$

Numerous optimization algorithms seeking to solve the mathematical problem are available. Most optimization algorithms employ a line search along search direction d for Eq. (2) such that,

$$\vec{p}^{n+1} = \overrightarrow{p^n} - \lambda d \tag{3}$$

where λ is a step size parameter.

The search directions d in Eq. (3) are composed from the gradients dJ/dp; Quasi-Newton methods are also employed to compute the approximations to the Hessian matrix of second derivatives, $H_{ij} = \frac{d^2 J}{d\vec{p}_i d\vec{p}_j}$, by differences of gradients in an updating scheme. As a result, the computation of the gradients dominates the computational cost for high-dimensional parameter spaces. The gradient can be calculated by simply applying the finite-difference method [8],

$$\Delta J(\vec{p}) = \frac{J(\vec{p} + \delta\vec{p}) - J(\vec{p})}{\delta\vec{p}} \tag{4}$$

The finite difference method can be used in Eq. (4) to find these sensitivities but is in general significantly more expensive, requiring at least one additional flow solution per parameter. It works particularly well if the design parameter $\vec{p}$ is of order O(10) since the re-meshing technique is fast and robust, even for generating RANS meshes. The optimization approach can be sped-up by calculating the gradient using "smart parallelization", i.e., all derivatives are computed at once with each one being computed on its own cluster node. Thus the computation time can be reduced to almost that of just one additional CFD run.

B. The mesh generators

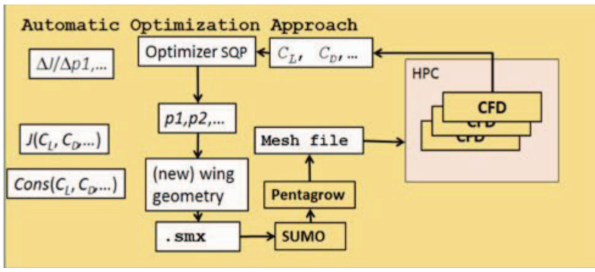

Figure 2 Automatic optimization approach using SUMO

Only unstructured meshes are considered in this paper in order to provide some comparisons. The mesh generators are evaluated since the quality of the meshes is one of most important issues for shape optimization designs. Two different mesh generators (the commercial software Pointwise [9], as well as the in-house software SUMO developed by KTH [10]) are used to create meshes on the same surfaces of geometry. SUMO can be used for automatically generating Euler (tetrahedron) and RANS (pentahedron) meshes. To generate the RANS mesh, Pentagrow [11] needs to be run before TetGen in batch mode by executing a configuration file with a list of user-defined parameters to set up the prismatic layers such as the first cell height, the total number of layers, the growth rate and so forth.

The automatic optimization framework is built based on loosely coupling different computational modules. The CFD solvers and geometric handling packages are complex. Moreover, the commercial packages have data structures and application program interfaces, which are proprietary with documentation not available to users. However, every package has input and output files which are well documented and thereby make them machine assessable. The computational modules can be coupled by monitor-type programs which understand the file formats and know how to read and write the files and how to translate between different parameterizations.

Fig. 2 shows the process of a set of optimization scripts creating a SUMO configuration file with the geometry computed from the parameters. The grid is based on a surface grid generated by SUMO and then sent on to Pentagrow which makes a prismatic-layer based volume grid, with the output being a mesh-file for CFD solvers. The CFD solver(s) calculates the flow solution and return quantities of interest (such as pressure fields, forces, and moments) as files. These are read by the optimization scripts, which then evaluate the objective and constraint functions, compute the updated values of the parameters, and repeat the sequence. The gradients that are needed by the optimization algorithm are computed by finite differences. The loose coupling makes it possible to easily distribute the requisite tasks on cluster computers.

The baseline CRM wing-body configuration is investigated by starting from the same surface mesh and gridding generated

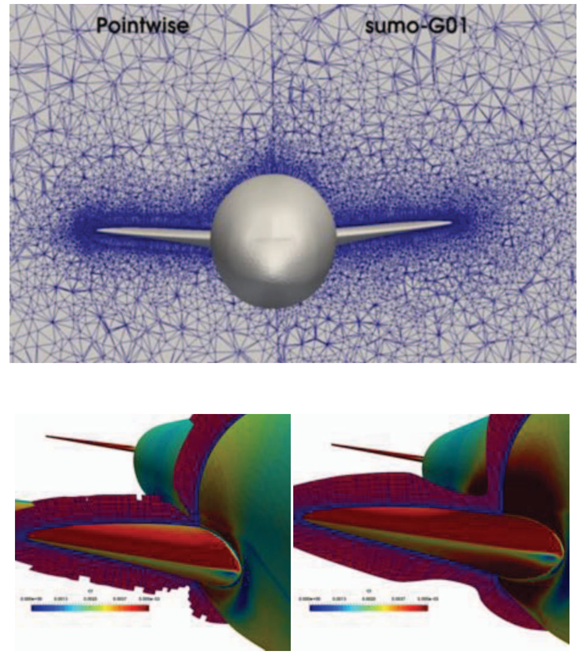

Figure 3 Grid comparison of tetrahedral resolution for CRM wing-body configuration, Pointwise-grid vs. SUMO-gird. Top: near-field; Bottom-left: near-field (zoom-in) using Pointwise; Bottom right: near-field (zoom-in) using SUMO.

TABLE I. THE COMPARSION FOR THE MESH GENERATE BY POINTISE AND SUMO

Mesh generator	#Points	#Tetrahedron	#Pyramid	#Prism
PointWise	4,601,010	4,185,417	49,413	7,618,103
SUMO-G00	10,296,466	5,507,393	-	18,635,680
SUMO-G01	10,548,294	7,108,594	-	18,635,680

TABLE II. VALIDITY FOR THE RESULTS OF LIFT, DRAG AND PITCHING COEFFICIENTS USING SU2

Mesh generator	C_L	C_D	C_M
PointWise	0.500426	0.0299926	0.0417215
SUMO-G00	0.488478	0.0307847	0.0485054

by both SUMO and Pointwise. The meshes for the two generators shown in Table 1 are generated to make sure that the SUMO mesh with refinement of level 1 (SUMO-G01) and the Pointwise-mesh share the same settings (so that the meshes that are obtained are as similar as possible). Note that the mesh generated by Pointwise is a half configuration with a symmetric plane while SUMO can only generate a full configuration within the domain. The grid comparison for the tetrahedral resolution for the CRM wing-body is shown in Fig. 3. The mesh generator SUMO can increase the resolution in the region where the separated flow appears around the surface of wing, see the bottom sub-figure in Fig. 3.

As the quality of the meshes directly affects the computational results, we compared the results of the lift coefficient C_L, the drag coefficient C_D, and the pitching moment C_m with different meshes using SU2 [12]. The results are shown in Table 2. These coefficients are very close to each other, which illustrates the validity of the meshes.

C. The CFD Solvers

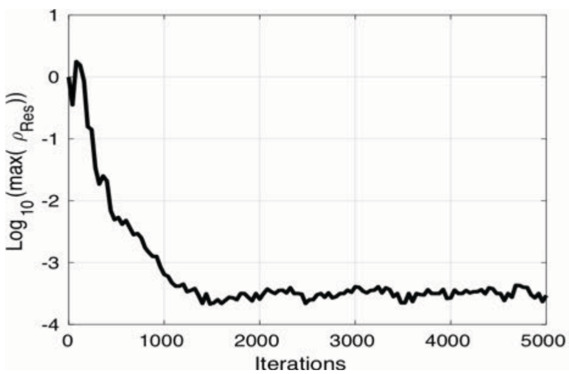

Figure 4 The residual of the different efficiencies with time iterations

SU2, an open-source code developed by the Aerospace Design Lab of the Department of Aeronautics and Astronautics at Stanford University, is used for the high-fidelity aerodynamic shape optimization [13-15]. It is a computational analysis and design package that has been developed to solve multi-physics analysis and optimization tasks using unstructured mesh topologies. The optimization package within SU2 is used by typing just one command line, provided that the cost function, constraints, the control points of the Free Form Deformation (FFD) Bézier box for wing parametrization and the other numerical parameters are all correctly set up in the configuration file. In the SU2 code, the CFL number can be set to be larger compared with some other CFD solvers since the implicit Euler method is used for the numerical integration in time. The RANS simulations are performed with the improved SA-DDES (Spalart-Allmaras–Delayed Detached Eddy Simulation) turbulence model implemented in SU2 [16]. The historical maximum residual of the density with time iterations is shown in Fig. 4. After around 60 iterations, the efficiencies are convergent to the objectives.

D. The performance results

As every CFD simulation is very time-consuming, the scalability tests were carried out on the high-performance computing clusters Beskow and MareNostrum before we performed large simulations. Beskow [17] is based on Intel Haswell processors and Cray Aries interconnect technology. It consists of 2,060 compute nodes, each of which consists of 32 Intel Xeon E5-2698v3 or 36 Intel Xeon E5-2695v4 cores. MareNostrum [18] is a supercomputer based on Intel SandyBridge processors. It has 36 racks dedicated to calculations. These racks have a total of 48,448 Intel Sandy Bridge cores with a frequency of 2.6 GHz and 94.625 TB of total memory. In total, each rack has a total of 1,344 cores and 2,688 GB of memory. The top figure in Fig. 5 shows the performance results for the mesh SUMO-G01 consisting of 10.5 million points on Beskow. We ran the benchmarking tests for 10,000 iterations. The strong scaling performance measured in total execution time in seconds is presented in this figure. The parallel efficiency $\eta(\%)$ at the maximum number of used cores is defined as:

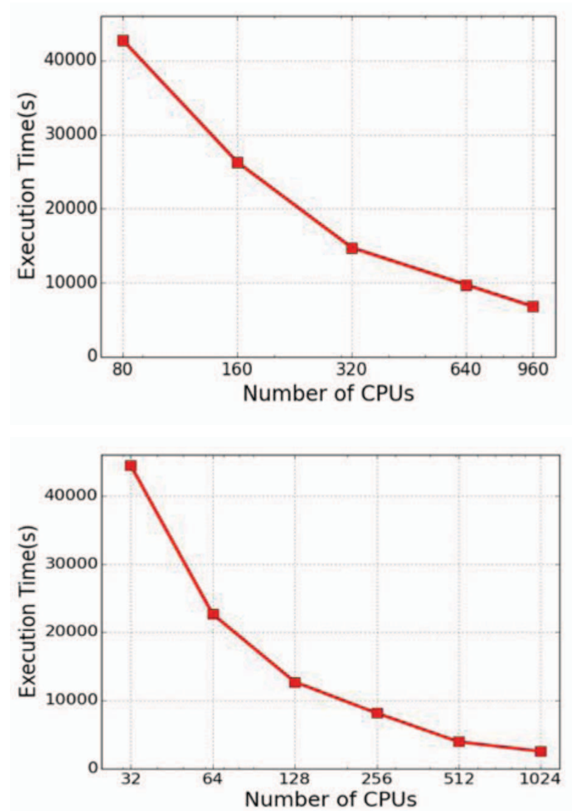

Figure 5 Performance results on Beskow (top) and MareNostrum (bottom)

$$\eta(\%) = \frac{T_{min} \cdot N_{min}}{T_{max} \cdot N_{max}} \cdot 100$$

where $T_{\min}$ is the total execution time for the minimum number of cores $N_{\min}$, while $T_{\max}$ is the total execution time using the maximum number of cores N_{max}. A parallel efficiency $\eta = 57.1\%$ can be achieved for up to 960 cores compared with 80 cores. The bottom graph in Fig. 5 shows the performance results for a mesh with a baseline of 10 million mesh points on MareNostrum. A parallel efficiency $\eta = 54.8\%$ can be achieved for 1024 cores compared with 32 cores. As a consequence of this, we ran all production simulations on 256 - 640 cores to obtain reasonable balances between the elapsed times and cost performances.

E. The optimization results

The optimization result for the CRM using the SU2 adjoint solver (with mesh deformation by FFD) with 242 DV are shown in Fig. 6. The boundary nodes of the computational grid need

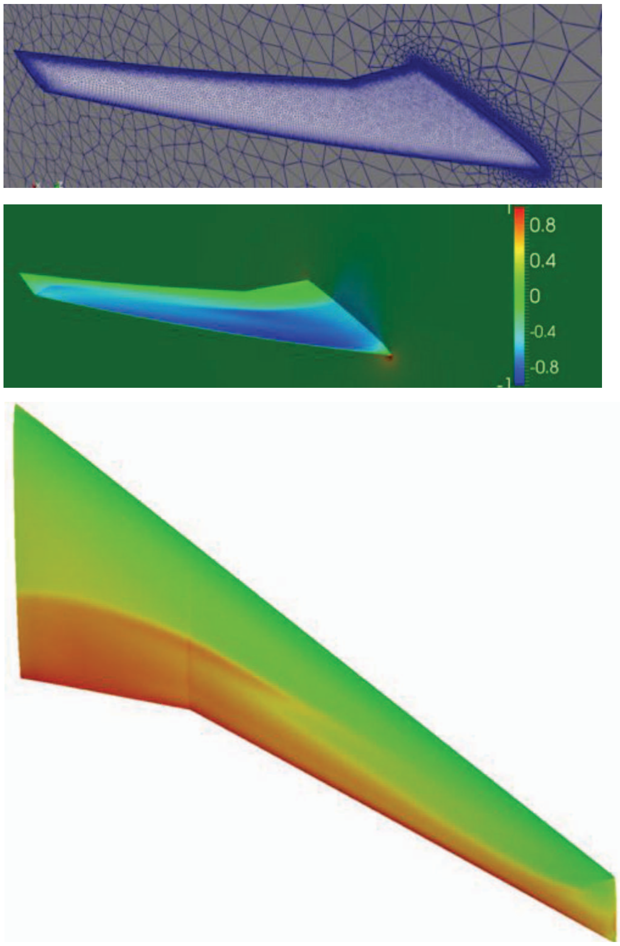

Figure 6 The optimization result for the CRM wing

to be deformed outwards (red colour) in order to reduce the drag, see Fig. 6.

III The aero-structural optimization for UAV with an High-AR Wing

The UAV configuration is provided in WP4 within the AGILE project, with cruise Mach around $0.2-0.3$ and wing planform parameters as design variables (e.g. span, chord, winglet cant angles etc.), see Fig. 7. In this paper, we only focus on preliminary designs with the static aero- elastic analysis, for example, aileron reversal analysis using the low-fidelity analysis tool based on the linearized model.

The flow chart of the MDO process for the aero-structural optimization is presented in Fig. 8. The VLM (Vortex-Lattice Method) solver for CFD takes the aerodynamic mesh generated from the geometries defined by a CAD or the CPACS format. The FEM (Finite Element Method) solve for the structural optimization is connected to the VLM solver via the mapped forces and then generates the deformed meshes.

F. Tornado, a VLM with structural modelling for the conceptual design

The work on developing tools for collaborative conceptual aircraft design at Airinnova is progressing within the AGILE EU-Project. Within this framework, new methods and software

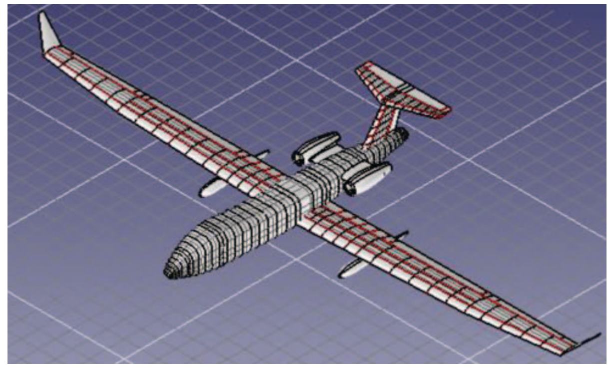

Figure 7 The concept of the UAV with high AR wing

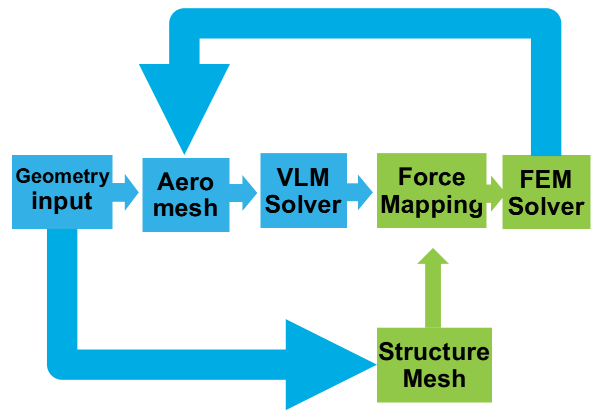

Figure 8 The flow charter between VLM and FEM

for aircraft design are being developed. One of these tools is the Vortex lattice method implementation Tornado. Vortex lattice software has been around for a couple of decades. The initial work derived from Prandtl's lifting line theory was one of the first computational fluid dynamic methods that was implemented for use on computers. The relative simplicity of the code makes it easy to implement and the robustness of the geometry input makes it very useful in conceptual aircraft design. Today, there is a range of VLM software available for download on the internet. One of these is Tornado, which is implemented in MATLAB. The Tornado software package is distributed under a GNU - General public license and the package is designed so it can be integrated easily into other software, see Fig. 9 which shows Tornado integrated into a structural Finite Element tool.

One of the current modules being developed is a structural model for evaluating wing structure stiffness and deformation modes. Together with the aerodynamic modelling provided by the VLM, the structure modelling provides a tool that can be used to assess many aeroelastic effects, such as static divergence, aileron control power reduction/reversal, gust loads, different load case shapes and clearances and initial structural weight estimations.

G. PyTrondo: a VLM solver for low-fidelity aero-structural

PyTornado is an enhanced and optimized code from Tornado, which ports the Tornado software from the MATLAB

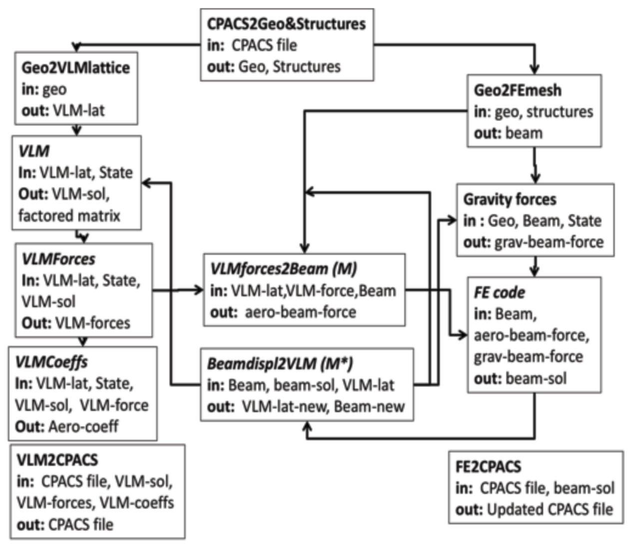

Figure 9 The instruction code of Tornado

proprietary platform to a truly open access platform in Python. In this implementation, the user interface and graphics will be coded in Python, while core functions will be compiled in C. It was therefore anticipated that there would be a significant speedup of the code execution in optimization settings. Fig. 10 shows an initial comparison between the computational time of the Tornado MATLAB implementation and the C implementation when increasing the number of panels. The indicated speedup is roughly one order of magnitude. The Python/C implementation preserves all the functionalities that MATLAB Tornado has, and also includes the following new features:

- the input format is CPACS compatible;
- it is possible to use a simple beam model for structural analysis, and
- new geometrical features have been added such as slats, all moving wings (canards, tailerons), and control surface taper amongst others.

The architecture makes it possible for the C code to be run stand-alone, without a user interface using input files or it can be run by being called from a Python User interface, or from MATLAB. Fig. 11 shows an architectural outline, with the three tracks (Python, C or MATLAB). The Python track can call all of the internal functions in C, for example to access the downwash matrix (aerodynamic influence matrix) directly. The C- and MATLAB-tracks operate on a black box executable using input files without access to the internal memory structures.

H. The methods

The aerodynamic code is a straight forward vortex lattice implementation with the usual assumptions and limitations, including thin wings, incompressible flow, and small angles of attack. The structural code models the wing as a cantilever stick beam clamped at the fuselage centre line. The second moment of inertia of the beam is computed from a box beam assumption, where the spar depth is determined from the available space in the selected wing profile. Several hard points for the attachment of engines are incorporated to transfer the weight and thrust of the engines. An allowance for fuel and structural weight is made. All forces are collected and applied to the structure.

The aerodynamic and structural meshes are separate. Aerodynamic forces and moments from the vortex lattice are mapped on the structural finite element mesh using a linear interpolation. The internal forces of the structure are resolved and the structural response in deformations and rotations is computed. These deformations are then mapped back to the process is reiterated with the deformed aero meshes until the changes in structure between the iterations are small, the aerodynamic mesh which is transformed accordingly. The aerodynamic forces are computed for the new mesh and structure diverges, or the limit for the maximum number of iterations is reached.

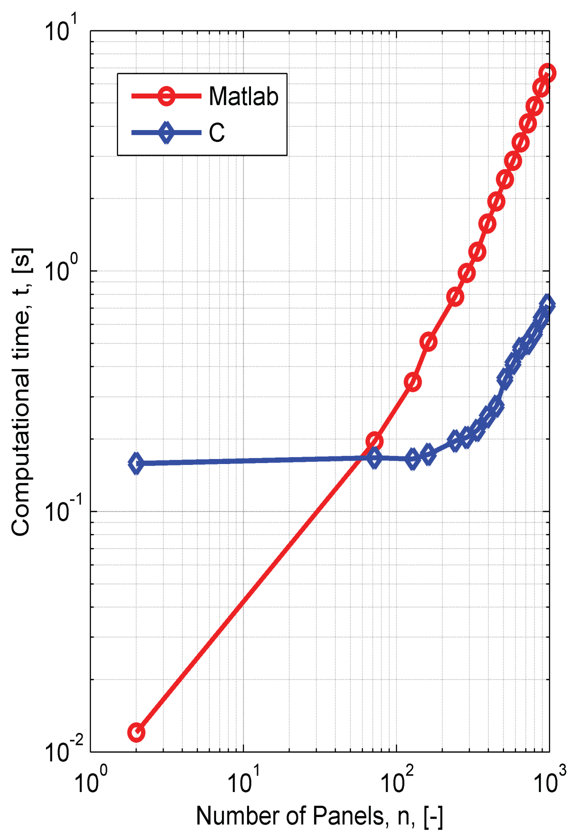

Figure 10 Computational speed comparision of Matlab Tornado and C Tornado

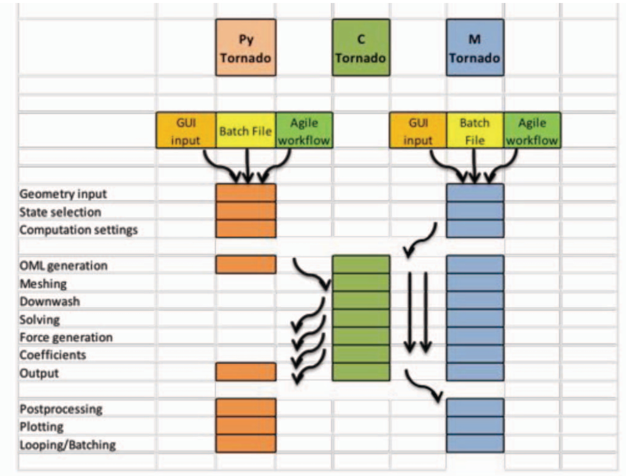

Figure 11 The PyTornado Architecture

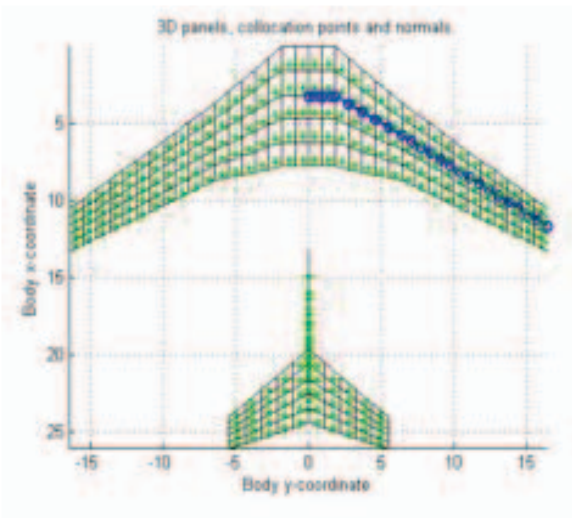

Figure 12 Aerodynamic and structural mesh of the Boeing 727 used as a test case

A test case aircraft, the Boeing 727-100 was used, for which the aerodynamic mesh is shown in Fig. 12. All six degrees of freedom in deformations of the cruise load case (1 g) are shown in Figure 13. Examining the steady solutions, several interesting key design points may be studied. Maximum manoeuvre loads and deformations may be evaluated, as well as gust loading. Static divergence can be detected and roll control authority for the detection of aileron reversal can be evaluated. From the stiffness of the wing, the wing box structural weight and the stress along the span can be computed structures that diverge structurally usually do so within three iterations, while converging solution generally converge within five iterations. The Boeing 727 case span-wise distribution is shown in Figure 14, with 4 iterations. Numerical limit cycle oscillations, triggering the maximum number of iteration cut-off have been observed but are rare.

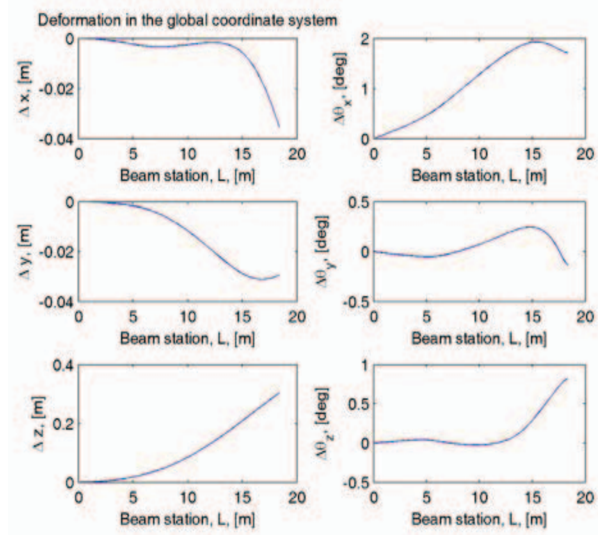

Figure 13 Computed structural deformation in the cruise load case

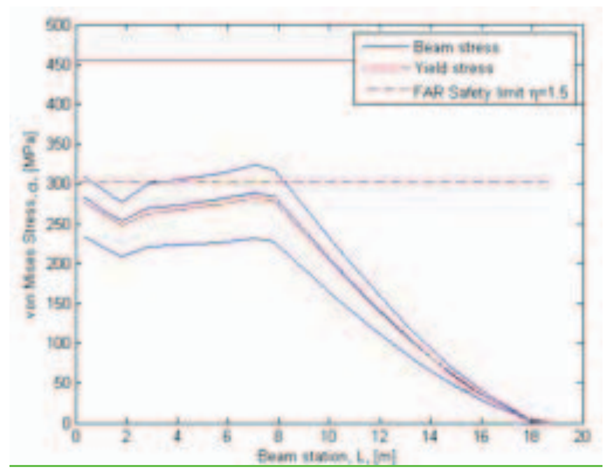

Figure 14 Spanwise distribution of maximum von Mieses stress in the Boeing 727 wingspar

III. DISCUSSIONS AND CONCLUSIONS

The results from the structural code have been verified by comparing them with analytical Euler beam deflections, however, the predicted structural weight differs significantly when compared with the "as-built" wing [8]. This is due to the significant simplification of the geometry needed to provide a manageable computational workload and matching the lack of geometrical definition in the beginning of an aircraft design project.

The mixed automatic aerodynamic and aero-structural optimizations for wing designs have been presented in this paper. The CRM model and the UAV with a high AR wing were employed in the tests using the mixed fidelity tools. The case studies illustrate that the automatic fidelity tools have potential to increase the optimized efficiency that can be achieved when using computational simulations to design aircraft wings.

ACKNOWLEDGMENT

This work is financially supported by the AGILE Project funded under H2020-EU.3.4, project reference 636202 and the Swedish e-Science Research Centre (SeRC). The use of the computational resource was supported by the PRACE project, which is funded in part by the EUs 7th Framework Programme (FP7/2007-2013) under grant agreement no. RI-312763.

REFERENCES

[1] M. Zhang, Contributions to variable fidelity MDO framework for collaborative and integrated aircraft design, Doctoral Thesis, KTH, Stockholm, Sweden, 2015

[2] M. Zhang, and A. Rizzi, "Aerodynamic wing shape optimization based on the computational design Framework CEASIOM", Aircraft Engineering and Aerospace Technology, Vol. 89, NO. 2, pp. 1-12, 2017

[3] CRM, the common research model, https://commonresearchmodel.larc.nasa.gov

[4] AGILE EU-project, www.agile-project.eu

[5] CPACS, A Common Language for Aircraft Design, https://software.dlr.de/p/cpacs/home/

[6] M. Zhang, J. Gong, M. Barth, and L. Axner, "Automation of high fidelity CFD analysis in aerodynamic design", PRACE white paper

[7] L. Osusky, H. Buckley, and D. W. Zingg, "Aerodynamic design optimization discussion group Case 4: single- and multi-point optimization problems based on the CRM wing", available at: https://info.aiaa.org/tac/ASG/APATC/AeroDesignOptDG/Test%20Cases/ADODG%20Case%204%20CRM%20Wing.pdf

[8] L. Zhoujie, K. Gaetan, W. Kenway, and R. R. Joaquim, "Aerodynamic shape optimization investigations of the common research model wing benchmark", AIAA Journal, Vol. 53, No. 4, pp. 968-985, 2014

[9] I. Griva, S. G. Nash, and A. Sofe, Linear and Nonlinear Optimization, Second Edition, SIAM, 2009

[10] Pointwise, Mesh generation software for CFD, http://www.pointwise.com/

[11] M. Tomac, Towards Automated CFD for engineering methods in aircraft design, Doctoral Thesis, KTH, Stockholm, Sweden, 2014

[12] SU2, the open-source CFD code, http://su2.stanford.edu

[13] F. Palacios, T. D. Economon, A. D. Wendorff, and J. J. Alonso, "Large-scale aircraft design using SU2", AIAA SciTech 5-9 January 2015, Kissimmee, Florida, 53rd AIAA Aerospace Sciences Meeting, 2015.

[14] T. D. Economon, D. Mudigereb, G. Bansalc, A. Heinecked, F. Palacios, J. Parkd, M. Smelyanskiyd, J. J. Alonsoa, and P. Dubeyd, "Performance optimizations for scalable implicit RANS calculations with SU2", Computers & Fluids, Vol. 129, No. 28, pp. 146-158 , 2016

[15] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonsoa, "SU2: An open-source suite for multiphysics simulation and design", AIAA Journal, Vol. 54, No. 3, pp. 828-846, 2016.

[16] M. Righi, "Turbulence modelling techniques for aeroelastic roblems: results and comments from the second AIAA aeroelastic rediction workshop", Aircraft Engineering and Aerospace Technology, Vol. 89, No. 5, pp. 683-691, 2017

[17] Beskow, www.pdc.kth.se/hpc-services/computing-systems

[18] MareNostrum, www.bsc.es/marenostrum