Numerical Simulation of the Elastic and Trimmed Aircraft

Andreas Michler and Ralf Heinrich

DLR, Institute of Aerodynamics and Flow Technology,
Lilienthalplatz 7, 38108 Braunschweig, Germany
Andreas.Michler@dlr.de

Summary

A simulation environment has been developed enabling the computation of the elastic and trimmed aircraft. It consists of a trim algorithm which is coupled with a procedure to account for the interaction between fluid and structure. The trim algorithm is based on a Newton method with a discretized Jacobian. It incorporates the six degrees-of-freedom (DoF) flight-mechanics equations and thereby enables to compute different trimmed states. The fluid-structure interaction (FSI) procedure uses a partitioned approach to compute the flow around the configuration in static aeroelastic equilibrium. This simulation environment has been successfully applied to trim a rigid transport-aircraft configuration in viscous flow as well as in inviscid flow with rigid and flexible wing.

1 Introduction

Cruise constitutes the major phase of flight for a transport aircraft, in which most of the aircraft's fuel is consumed. Hence for this phase, it is of utmost importance to accurately predict the aerodynamic coefficients, which are closely related to fuel consumption. Their computation involves the disciplines fluid, structural and also flight mechanics. Firstly, during flight, especially the wing noticeably deforms and reaches an equilibrium position in cruise, the so-called *static aeroelastic equilibrium*. This is accounted for by a partitioned approach, enabling the application of specialized software tailored to each discipline and already in use in industry. Secondly, in cruise, the loads acting on the aircraft are balanced: a *trimmed state* is reached, in which the statically stable aircraft returns after perturbation. In order to simulate these interactions, a simulation environment has been developed coupling trim algorithm with an FSI procedure for the computation of the static aeroelastic equilibrium. The FSI procedure uses the hybrid RANS code TAU [8] for the flow computation and the commercial software ANSYS [10] for the solution of the structural equations. The trim algorithm is based on a Newton method, in which the derivatives of the Jacobian are discretized

A. Dillmann et al. (Eds.): Numerical & Experimental Fluid Mechanics VII, NNFM 112, pp. 109–116.
springerlink.com

by finite differences. For the deflection of control surfaces, prescribed by the trim algorithm, the Chimera technique is used. It provides a fast and reliable means of rigid-body component rotations even in the case of large rotations.

The outline of this paper is as follows. Section 2 describes the algorithm used for calculating the trimmed state, and Sect. 3 deals with the coupling between fluid and structure, as well as between FSI procedure and the trim algorithm. Section 4 explains the results for applications involving some or all of the disciplines mentioned above. This paper is concluded by Sect. 5.

2 Trim Algorithm

For stability reasons, a trimmed state is adopted in cruise [9]. It is achieved by deflecting so-called *control surfaces*, thereby cancelling moments acting on the aircraft, as noted in [9]; elevator, aileron, rudder and even the horizontal tail (HT) may act as such.

More generally, trim can be seen as the process of balancing forces and moments acting on the aircraft to achieve a certain state of flight. To this end, the aircraft is perceived as a system with input and output parameters: the output parameters are of interest to the observer and influenced by the input parameters, determining the state of the aircraft. This state is governed by the 6-DoF flight-mechanics equations [6], linking input and output parameters. These equations require the aerodynamic forces.

The trim problem can then be stated as follows. Let $\tilde{\boldsymbol{z}} \in \mathbb{R}^n$ be the vector of trim input parameters and $\tilde{\boldsymbol{y}} \in \mathbb{R}^m$ the vector of trim output parameters, composed of some or all the input or output parameters, respectively; find the values for the trim input parameters $\tilde{z}_i, i = 1, \ldots, n$ such that the trim output parameters $\tilde{y}_j, j = 1, \ldots, m$ are equal to prescribed values $\hat{y}_j$ specifying a certain trimmed state. In case the trim input vector $\tilde{\boldsymbol{z}}$ does not contain all the components of the input vector, the other components not included are kept constant during the trim process. This problem is formulated as a root-finding problem and, for the solution, a Newton method is applied. The basic equation reads as follows:

$$\mathbf{J}^{(k)} \Delta\tilde{\boldsymbol{z}}^{(k)} = \boldsymbol{q}^{(k)}, \tag{1}$$

where k denotes the current trim iteration number, $\Delta\tilde{\boldsymbol{z}}$ the vector of the step sizes of trim input parameters and $\boldsymbol{q}$ is the difference between current values of the trim output parameters and their prescribed values, $\boldsymbol{q} = \hat{\boldsymbol{y}} - \tilde{\boldsymbol{y}}$, termed the *quality vector*. $\mathbf{J}$ is the Jacobian with the trim output parameters differentiated w.r.t. the trim input parameters as entries. Since, in general, these derivatives are not given explicitly, they are discretized using finite differences. This may be computationally expensive in case CFD methods are involved; fortunately, not every variation of the trim input parameters requires a CFD computation. For the solution of (1), singular value decomposition (SVD) is used, since it is also applicable to the general case of $m \neq n$, resulting in a non-square Jacobian; furthermore, it provides a means of coping with ill-conditioned Jacobians [2]. SVD factorizes the Jacobian and yields a product of three easily invertible matrices.

The new trim input parameters are then calculated as follows:

$$\tilde{\boldsymbol{z}}^{(k+1)} = \tilde{\boldsymbol{z}}^{(k)} + \kappa \cdot \Delta\tilde{\boldsymbol{z}}^{(k)}, \tag{2}$$

where κ is not equal to 1 and scales the step size vector appropriately in the cases of violation of

(a) given maximally allowed changes of input parameters from one iteration to another (termed *relative bounds*), or
(b) given upper and lower limits on trim input parameters (termed *absolute bounds*).

The solution's quality is measured by the quality function $Q^{(k)}$, computed, e.g., as the ℓ^2-norm of the current quality vector $\boldsymbol{q}^{(k)}$.

The trim algorithm terminates if

(1) the given maximum iteration number is reached,
(2) the current value of the quality function reaches a prescribed tolerance,
(3) the ℓ^2-norm of the current step size vector $\Delta\tilde{\boldsymbol{z}}^{(k)}$ reaches a prescribed tolerance,
(4) the value of the quality function increases (more than allowed), or
(5) relative or absolute bounds are violated.

A successful trim calculation ends because of either stopping criterion (2) or (3). Although only one solution may be found at a time, starting from different values of input parameters may yield different solutions, due to the non-linearity of the underlying physics or due to a different number of trim input and output parameters. This can be avoided by imposing bounds on the trim input parameters.

Common trimmed states are cruise, i.e. steady level flight, climb or descent with constant acceleration. For cruise conditions, the translational and rotational body accelerations, $\dot{u}_b$, $\dot{w}_b$ and $\dot{q}$, respectively, as well as the flight path angle γ have to be zero, hence: $\tilde{\boldsymbol{y}} = [\dot{u}_b, \dot{w}_b, \dot{q}, \gamma]^T \overset{!}{=} [0,0,0,0]^T$. This can be achieved by a certain combination of angle of attack (AoA) α, tail angle η, pitch angle Θ and thrust T, i.e. $\tilde{\boldsymbol{z}} = [\alpha, \eta, \Theta, T]^T$. The Jacobian is hence a 4×4-matrix. The results for such a trim calculation are shown at the beginning of Sect. 4. For small AoA and $\alpha \equiv \Theta$, the thrust balancing the drag and acting in the aircraft's center of gravity, as well as the pitching rate q being zero, the 3-DoF equations can be simplified to

$$L = G, \tag{3}$$

$$M = 0, \tag{4}$$

where L is the aerodynamic lift force balancing the aircraft's weight G, M its aerodynamic moment. Thus, the trim vectors are $\tilde{\boldsymbol{z}} = [\alpha, \eta]^T$ and $\tilde{\boldsymbol{y}} = [C_L, C_M]^T \overset{!}{=} \left[\hat{C_L}, 0\right]^T$ with C_M being the moment coefficient and C_L the lift coefficient. This yields a 2×2-Jacobian and reduces the number of necessary flow / FSI computations. Trim calculation with these parameters were also performed in Sect. 4.

3 FSI Procedure and Coupling between FSI and Trim Algorithm

Fluid-structure interaction constitutes a multi-physics problem in which two disciplines are coupled on the common boundary or *interface*. A *partitioned approach* [1] is adopted here: each domain is spatially independently discretized and the equations governing each domain solved with a different solver. The solution of each set of governing equations is performed subsequently until the static aeroelastic equilibrium is reached. On the interface, coupling terms have to be exchanged: aerodynamic forces have to be transferred to the structural grid, and the nodal displacements of the structural grid have to be transferred to the CFD grid. For the first, a *nearest neighbor mapping* technique is used, for the latter a volume spline interpolation algorithm [3]. The subsequent deformation of the CFD grid is performed by an algorithm based on radial basis functions [3]. This coupling scheme reads as follows:

(1) **CFD:**
Calculate the solution of the Euler- or Navier-Stokes equations
(2) **CFD → CSM:**
transfer the aerodynamic forces to the nodes of the structure grid
(3) **CSM:**
calculate the nodal displacements for the structure grid
(4) **CSM → CFD:**
transfer the nodal displacements of the structure grid to the CFD grid
(5) **stopping criteria:**
check if a stopping criterion is met
`if YES`: end and take the aerodynamic coefficients of last CFD computation
`if NO`: **deform** the CFD grid and go back to step (1)

The coupling procedure ends if

(A) a given maximum number of coupling iterations is reached;
(B) the aerodynamic coefficients of subsequent iterations do not change significantly, or
(C) the difference between maximum nodal displacements for the structure (and hence the distance between CFD meshes) of subsequent coupling iterations is negligible,

(C) being applicable for static aeroelastic equilibrium.

The speed of the modern transport aircraft in cruise falls in the transonic flight regime, governed by the non-linear compressible stationary Navier-Stokes equations. These equations can be adequately solved by modern CFD solvers, such as the hybrid RANS solver TAU, which is employed here. The software Centaur [11] was used for the generation of unstructured grids, consisting of tetrahedra for the case of inviscid flow described by the Euler equations, and additionally prisms for the resolution of the boundary layer for visous flow, as described by the Navier-Stokes equations. For the deflection of the horizontal tail necessary for trim, the Chimera technique was used, enabling the relative rotation of overlapping component grids [5]. Geometries have been created or modified with the CAD program CATIA [12]. The DLR-F12 (cf. Sect. 4) transport-aircraft model was cleaned using this program; furthermore, in order to use the Chimera technique, a gap between horizontal tail and fuselage was generated and Chimera cylinders around the horizontal tail defined.

The structure is governed here by the elastostatic equations. They are solved using the commercial software ANSYS. For now, solely the influences of the elasticity of the wing is taken into account. The structure of the wing is modelled following the approach described in [7], where a model of the structure is generated based on the surrounding aerodynamic surface mesh. This model basically consists of skin, spars and ribs. As to the boundary conditions, no translation is allowed for the nodes at the root; this constitutes a simplification compared with a free-flying aircraft. The loads, transferred from the CFD solution, act on the nodes of the surface of the structure.

The trim algorithm acts as master program during the computation, setting the input parameters for the FSI procedure. The FSI procedure then computes the corresponding static aeroelastic equilibrium and passes the aerodynamic coefficients on to the trim algorithm.

The simulation environment was developed under Linux using the scripting language Python [4] to interface between stand-alone applications such as ANSYS or TAU and the routines for the transfer of coupling terms and deformation. It was also used for the communication with one of the institute's clusters, where the time-consuming flow calculations for the DLR-F12 were performed. The other applications were run on a local PC. The trim algorithm was written in Python, as well as the program for the generation of the model for the structure. ANSYS' scripting language APDL is used for the generation of the actual model within ANSYS, the mesh generation, the setting of the boundary conditions, the solution of the elastostatic equations and post-processing. For the generation of these scripts, templates stemming from [7] were adapted.

4 Applications

The applications are presented in the order of increasing number of disciplines involved: first, only flight mechanics, then CFD additionally and finally also structural mechanics.

The first test case involves four trim input and trim output parameters (cf. Sect. 2). It exemplifies the successful application of the trim procedure for more complex trim problems involving flight mechanics equations. The aerodynamic coefficients were provided by simple analytical models depending linearly on AoA α and tail angle η. As further simplification, the thrust was assumed to act at the center of gravity. The accelerations used as trim output parameters were computed by solving the 3-DoF flight-mechanics equations. Figure 1 shows that trim input and output parameters converge at the end of the trim calculations, and the trim output parameters reach the values initially prescribed.

The configuration used in the next test case is the rigid DLR-F12 transport-aircraft configuration, a windtunnel model scaled to real dimensions. Cruise conditions were assumed for a speed of Ma=0.85 at a height of about 10km. The grid is made up of about 4.25 million grid points, and prism layers were used to resolve the boundary layer (cf. Fig. 2). The Spalart-Allmaras turbulence model was chosen. On the right side of Fig. 2, the success of the trim calculation can be seen, as the *goal values* of $\hat{C}_L = 0.46$ and $\hat{C}_M = 0$ for cruise condition had been prescribed.

An FE model for the wing of the DLR-F12 was developed, which consists of spars, ribs and skin and is meshed with shell elements. The total number of nodes on the surface is about 10,300. Trim calculations were performed with and without considering the elastic deflection of the wing with this structural model. In both cases, viscosity was

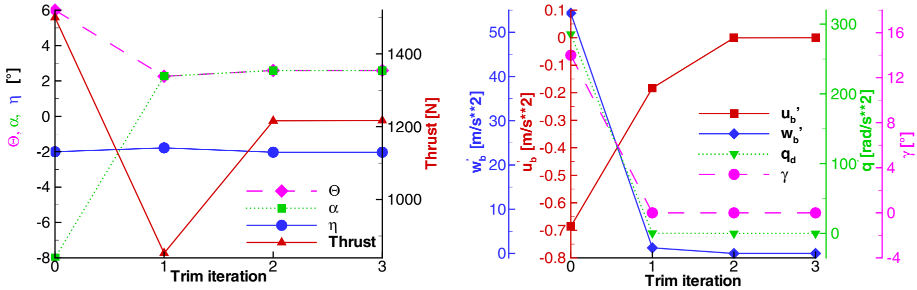

Fig. 1. Convergence of trim input (left) and trim output parameters (right)

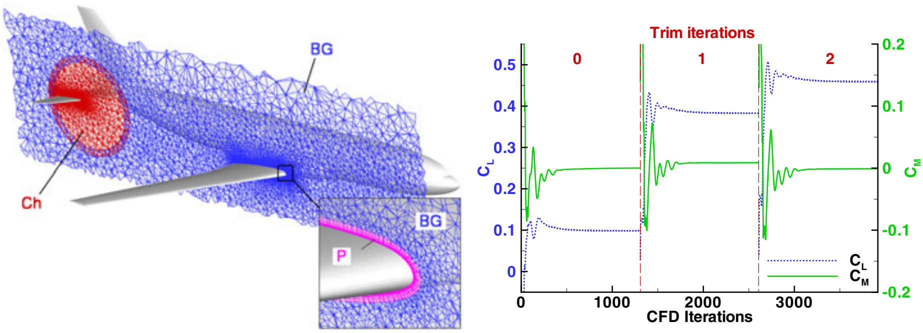

Fig. 2. A cut through the hybrid grid of the DLR-F12 is shown on the left side. The grid section, labeled "Ch", belongs to the Chimera block of the horizontal tail, overlapping the background grid ("BG"). The close-up shows prism layers ("P") near the wing root. On the right side, the convergence of the aerodynamic coefficients is plotted for CFD calculations of subsequent trim iterations.

neglected, but the same ambient conditions and goal values were prescribed as before. The CFD mesh was comprised of tetrahedra and had about 1.2 million points. For each trim iteration, FSI calculations had to be performed to obtain the values for C_L and C_M in static aeroelastic equilibrium. For the last trim iteration, Fig. 3 exemplarily shows the CFD meshes and the convergence of the aerodynamic coefficients for subsequent coupling iterations. Figure 4 (left) shows the convergence of trim input and output parameters during the trim process for both the rigid and the elastic case. Starting from different values for the aerodynamic coefficients C_L and C_M for the same angles α and η, the goal values of $\hat{C}_L = 0.46$ and $\hat{C}_M = 0$ were reached in both cases after just a few trim iterations. At the trimmed state, both the values for angle of attack and tail angle were higher in the case of the partially elastic configuration. Furthermore, as can be seen in Figure 4 (right), the vertical forces over the wing are shifted towards the wing root. An increase in drag (of about seven drag counts) was also found.

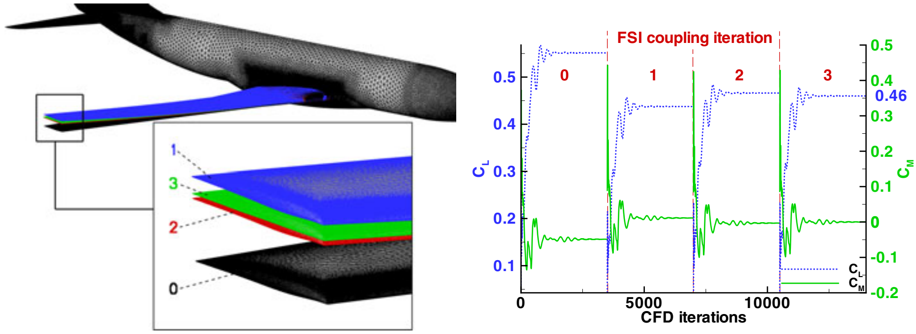

Fig. 3. On the left side, the surface meshes for subsequent FSI coupling iterations for the last trim iteration are presented: the undeformed mesh, labeled "0", and the deformed meshes after the first ("1"), the second ("2") and the third ("3") iteration. On the right side, the corresponding convergence histories of lift and moment coefficients are displayed.

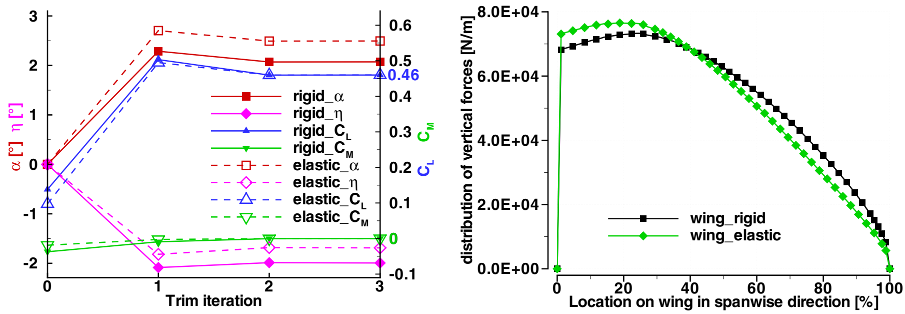

Fig. 4. On the left side, the convergence of trim input and output parameters are shown for the rigid and partially elastic DLR-F12 configuration in inviscid flow. On the right side, the distribution of vertical force in spanwise direction over the wing is displayed for the trimmed state. A location of 0% corresponds to the wing root, 100% to the wing tip.

5 Conclusions

A simulation environment has been presented enabling the computation of trimmed elastic configurations in static aeroelastic equilibrium. The trim algorithm uses a Newton method and can handle different trim conditions involving multiple trim input and output parameters. The simulation environment was applied to the DLR-F12 transport-aircraft configuration. First, the rigid configuration immersed in viscous flow was successfully trimmed for cruise conditions. Then, for inviscid flow, the same configuration was trimmed with a rigid wing and also for an elastic wing. For both cases, the trim algorithm succeeded in achieving the values for the aerodynamic coefficients prescribed for cruise conditions. For the same cruise conditions, the elastic case was found to require a higher angle of attack and tail angle, as well as to yield an increase in drag. Furthermore, a shift of wing loading towards the wing root was observed. This clearly indicates that elasticity cannot be neglected when considering cruise conditions. In the near future, the partially elastic DLR-F12 in visous flow will be trimmed and the results compared to those of the rigid configuration, such that the effects of elasticity, especially on the drag, can be more accurately assessed.

Acknowledgements

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG) within the framework of the Graduiertenkolleg 432, "Wechselwirkung von Struktur und Fluid".

References

[1] Felippa, C., Park, K., Farhat, C.: Partitioned Analysis of Coupled Mechanical Systems. Computer Methods in Applied Mechanics and Engineering 190, 3247–3270 (2001)

[2] Heath, M.T.: Scientific Computing: An Introductory Survey, 2nd edn. McGraw-Hill, New York (2002)

[3] Heinrich, R., Wild, J., Streit, T., Nagel, B.: Steady Fluid-Structure Coupling for Transport Aircraft. Deutscher Luft- und Raumfahrtkongress, Braunschweig (2006)

[4] Langtangen, H.P.: Python Scripting for Computational Science, 2nd edn. Texts in Computational Science and Engineering. Springer, Heidelberg (2006)

[5] Madrane, A., Raichle, A., Stürmer, A.: Parallel Implementation of a Dynamic Overset Unstructured Grid Approach. In: Neittaanmäki, P., Rossi, T., Majava, K., Pironneau, O. (eds.) Proceedings of the ECCOMAS 2004 Conference

[6] Murman, S.M., Aftosmis, M.J., Berger, M.J.: Simulations of 6-DOF Motion with a Cartesian Method. In: 41^{st} AIAA Aerospace Sciences Meeting, Reno, NV, January 6-9, AIAA-2003-1246

[7] Nagel, B.: Berechnung der strukturmechanischen Geometrie von Tragflügeln auf Basis eines aerodynamischen Netzes. Diplomarbeit, DLR-Institut für Faserverbundleichtbau und Adaptronik

[8] Schwamborn, D., Gerhold, T., Heinrich, R.: The DLR TAU-Code: Recent Applications in Research and Industry. In: ECCOMAS CFD 2006, Egmond aan Zee, The Netherlands (2006)

[9] Tewari, A.: Atmospheric and Space Flight Dynamics. Birkhäuser, Boston (2007)

[10] ANSYS homepage, `http://www.ansys.com/`

[11] Centaur homepage, `http://www.centaursoft.com/`

[12] CATIA homepage in German language, `http://www.3ds.com/de/`