

# COUPLED STATIC AND DYNAMIC AEROELASTIC SIMULATIONS IN TRANSONIC AND SUPERSONIC FLOW

M. Stettner, W. Haase, A. Eberle, J. Grashof, M. Schneider  
DaimlerChrysler Aerospace AG, Military Aircraft

**Keywords:** *Aeroelasticity, CFD, fluid-structure coupling, mesh deformation, parallel processes.*

## Abstract

*In design processes involving aeroelasticity problems, linear aerodynamic methods are most frequently used in industry. However, in the case of transonic and/or high-angle-of-attack flow, the critical and non-linear flow phenomena are not captured, hence the obtained interdisciplinary, aeroelasticity results are incorrect.*

*In the present paper the status of effort, needed to close the fidelity gap between aerodynamics and aeroelasticity by application of highly sophisticated techniques, is described, as there are parallel Euler and Navier-Stokes (CFD) solvers, mesh deformation algorithms, generic fluid-structure interfaces, and last but not least high-performance computing.*

*Static and dynamic aeroelastic simulation results are presented for the X-31A research aircraft wing.*

## 1 Introduction

Today, CFD methods of high fidelity are applied to both complex 3D geometries and complex flow situations covering all speed regimes. However, due to the neglect of structural flexibility in "pure" CFD simulations, the actual flow conditions may differ significantly from those computed. On the other hand, in industrial applications, the aeroelasticity methods do often base on the assumption of linear relations between deflection and aerodynamic pressure. Thus, critical non-linear phenomena may not be captured, particularly in transonic and high-angle-of-attack conditions where flow separation is likely to occur. However, both

flow areas concerning cruise and take-off and landing situations, are of particular interest in the context of modern civil and military aircraft design, i.e. the fidelity gap between aero- and structural dynamics needs to be closed.

Aiming at that goal, a practical approach is to combine sophisticated CFD methods (Euler or Navier-Stokes) with structural solvers in loosely coupled computational processes. Although the major drawback of such high-fidelity aeroelastic analyses might be seen in the required high amount of computing time, this technology is going to be developed and validated today, both because of their predictive accuracy and dramatically improving computing resources. Nevertheless, while linear methods deliver flutter speeds or control effectiveness within minutes at most, the more sophisticated analyses might take day(s) to yield a reliable answer on a workstation. Definitely, to become acceptable for daily use and application in aircraft design and optimisation, turnaround times have to be reduced by about two orders of magnitude. This goal can be met by either applying faster, application-tailored methods<sup>1,2</sup> or by the (current) perpetual increase of computing power. Of course, to gain reasonable turn-around times, high-performance computing is required with processes adapted to utilise opportunities for parallelisation within the analysis tools, on the level of aeroelastic processes and, additionally, within optimisation schemes.

In the following, the effort undertaken at DaimlerChrysler Aerospace, Military Aircraft (Dasa-M) to develop fast and robust algorithms and to identify and exploit parallel processes in

order to support fast, reliable and robust applications of high-fidelity methods in industrial aeroelasticity is described.

## 2 Structural Analysis

The structural analysis and optimisation package LAGRANGE<sup>3</sup> is a Dasa-M in-house development. It allows optimisation of finite element structural models with buckling static (stress and displacement), dynamic (time and frequency domain), aeroelastic (static and dynamic), and manufacturing restrictions with primarily analytical sensitivity calculation. Design variables can be of type “sizing” (element thickness) or “shape” (element geometry), and a suite of gradient-based optimisation algorithms is available. The structure is modelled by finite elements (FE) similar to those used in NASTRAN assuming geometric and material property linearity. For aeroelastic analysis, interfaces exist for reading the Aerodynamic Influence Coefficient (AIC) matrices from external linear-panel or doublet-lattice methods. In static aeroelasticity, component effectiveness is calculated, while in dynamic aeroelasticity the flutter velocity is determined at constant altitude as a function of Mach number and reduced frequency.

Initially, the structural system’s mass and stiffness matrices are assembled based on the element information contained in the input deck. In the case of static analysis, the stiffness matrix is Cholesky-decomposed, and the system deflections are calculated for each load case by forward-backward substitution. In the case of transient dynamic problems, an eigenvalue analysis is performed, and the entire system is transformed into modal co-ordinates, allowing a quick integration of the equations of motion due to the inherent reduction of degrees of freedom.

## 3 Aerodynamic Analysis

The CFD code FLOWer<sup>4</sup>, developed at the German Aerospace Centre (DLR), features parallel solution of the Euler- and Navier-Stokes equations on block-structured grids. The code

was enhanced to accommodate for the needs of transient aeroelasticity by implementation of restart capabilities which allow for second order accuracy in time, and mesh velocities in the flux formulations. FLOWer bases on the Jameson-Schmidt-Turkel approach and time dependent flow is carried out by using the dual time stepping<sup>5</sup> method.

To cope with aeroelasticity needs, a major addition to FLOWer was a new mesh deformation algorithm which is taking care of even large displacements of the body surfaces. In general, two main approaches exist: 1. The computational grid is treated as a quasi-structural system with translational and rotational springs between the grid points<sup>6</sup>. 2. A Poisson algorithm is applied to the entire grid accounting for hybrid, unstructured grids consisting of a combination of hexahedra, tetrahedra, prisms, and pyramids<sup>7</sup>.

Based on a comparison of the two major approaches, an algorithm has been developed and integrated in the FLOWer code which uses a combination of an unstructured Poisson algorithm and an algebraic mesh generation approach based on Transfinite Interpolation (TFI) in the curvilinear co-ordinate system. This algorithm was found to be advantageous with regard to robustness, mesh quality, and efficiency. The mesh deformation algorithm has been kept general enough to also efficiently handle block configurations with segmented block faces.

## 4 Fluid-Structure Interaction

In aerodynamic and structural analysis, the generation of models and meshes follows different criteria. In particular, meshes at the interface between the fluid and the structural domain differ, and since the relevant parameters like structural deflection or aerodynamic loads (as there are pressure and shear) are given at grid points, a direct exchange of information is impossible.

A common feature is the representation of deflections by splines<sup>8</sup>, in which the deflection of a surface point on one mesh is represented in

terms of the deflections of some or all nodes of the other mesh. Under the assumption that the geometric relation between grid points on the interfacing meshes remains unchanged, the coefficients are constant, and the transformation reads

$$\vec{u}_a = \underline{T} \vec{u}_s \quad (1)$$

with the vector of displacements of the structural nodes,  $\vec{u}_s$ , the vector of aerodynamic grid point displacements,  $\vec{u}_a$ , and the spline-dependent transformation matrix,  $\underline{T}$ . This form facilitates the conservation of virtual work, an essential condition for assuring that loads,  $\vec{F}$ , and deflections,  $\vec{u}$ , are transferred without introduction of errors, which relate to artificial positive or negative damping in dynamic analyses. The condition is expressed by

$$\vec{F}_s^T \delta \vec{u}_s = \vec{F}_a^T \delta \vec{u}_a \quad (2)$$

where  $\delta \vec{u}$  indicates compatible virtual displacements. Due to the compatibility requirement, equation (1) also applies to virtual displacements, and it can be shown that a transformation of loads conserving virtual work is given by

$$\vec{F}_s = \underline{T}^T \vec{F}_a \quad (3)$$

I.e., after integration of aerodynamic pressures and shear loads and reduction into the CFD grid points, structural node loads,  $\vec{F}_s$ , can be computed by reusing the transformation matrix derived for the deflections. Similarly, it is possible to start by imposing a distribution that may base on physical considerations like equilibrium of force and moment on the level of structural elements,

$$\vec{F}_s = \underline{N} \vec{F}_a \quad (4)$$

and arrive at an expression for the deflections that is similar to (1):

$$\vec{u}_a = \underline{N}^T \vec{u}_s \quad (5)$$

This often used approach is highly efficient, as  $\underline{T}$  (or  $\underline{N}$ ) is assembled only once

for a given configuration. However, one has to take care that

- the relative location of grid points is not changing during mesh deformation, particularly when large deflections result in geometric non-linearities, i.e. the use of a constant matrix  $\underline{T}$  is expected to cause errors,
- the desired properties contained in the deflection transformation (1) may lead to undesired effects in the loads transformation (3) and vice versa.

For the sake of generality, and at the immediate expense of computational efficiency, the present approach combines separate methods for deflection interpolation and loads transformation, in particular, a combination of (1) and (4) is used. In fact, the matrices,  $\underline{T}$  and  $\underline{N}$  are not explicitly assembled, but the underlying relations are used to build appropriate transformation relations. One of these transformations is a spline to provide a smooth interpolation of deflections. The other is a set of equations ensuring certain load distribution properties per structural surface element. A correction is included in the latter step to guarantee conservation of virtual work.

The central element of the current method is a so-called “neutral interface”, a continuous representation of the three-dimensional surface of the body, described in terms of two independent surface parameters. This interface can either be supplied from CAD, or be generated from the aerodynamic or structural meshes. Grid points and associated properties are mapped into the geometrically closest point on the neutral interface using Newton-Raphson iteration, by this reducing the three-dimensional coupling problem into two dimensions. The association of groups of structural nodes and elements with aerodynamic grid points on the neutral interface is referred to as “pairing” (**Figure 1**). Two different implementations of this general approach were developed.

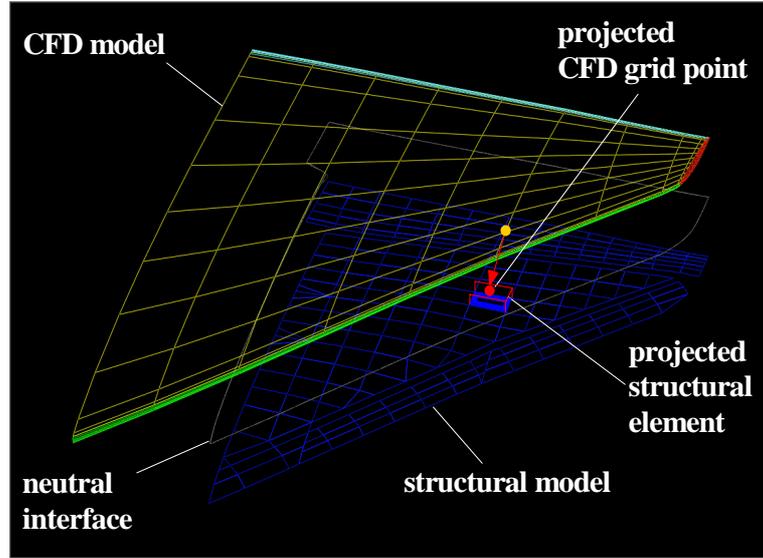


Figure 1: Neutral interface and pairing

The first, TRAEST (TRansformation Aero-Structures), includes splining routines of the structural analysis tool LAGRANGE, which are based on B-splines and enforce diminishing curvature of the spline surface at its domain boundaries. Three least-square spline fits are used for each component of the displacement vector. Loads are transmitted by projecting CFD pressure onto the neutral interface, fitting a least-square spline, and numerically integrating within the bounds of each projected structural element. The resulting load per element is distributed into its corner nodes in the 3D space by solving a linear system of equations in the unknown nodal loads. Concern of a four-node element and nodes to which no moments can be applied results in twelve unknowns. Six relations ensure that force components and related moments with respect to the element's area centre of gravity are counteracted by the nodal loads. For the remaining equations, requesting that the nodal loads be parallel to the aerodynamic load vector was found to yield the most reasonable solutions. However, the resulting system of equations is over-determined. A low weight is assigned to the directional requirement, and the system is solved by singular value decomposition (SVD).

The map of the structural model onto the neutral interface usually covers a smaller sub-domain than the aerodynamic model projection. Therefore, the load on the structure after this step is smaller than on the aerodynamic model. By SVD, residual loads are distributed on all structural nodes with no directional requirements. Because of encountered weaknesses in the deflection part due to mapping problems in highly curved and/or segmented areas of the neutral interface, this approach was not used.

The second implementation, GridCAD<sup>9</sup>, builds on a CAD and mesh generation software tool and features a graphical user interface. The common basis of the deformation transformation are Non-Uniform Rational B-Splines, NURBS. Linear interpolation is used to provide deflection values at the required spline base grid points with the spline fitted exactly through these points. Loads on CFD grid points,  $\bar{F}_{a,j}$ , are calculated using the linear base function of four-node CFD elements,

$$b_j = 0.25(\xi_j - \xi)(\eta_j - \eta); \quad -1 \leq \xi, \eta \leq 1 \quad (6)$$

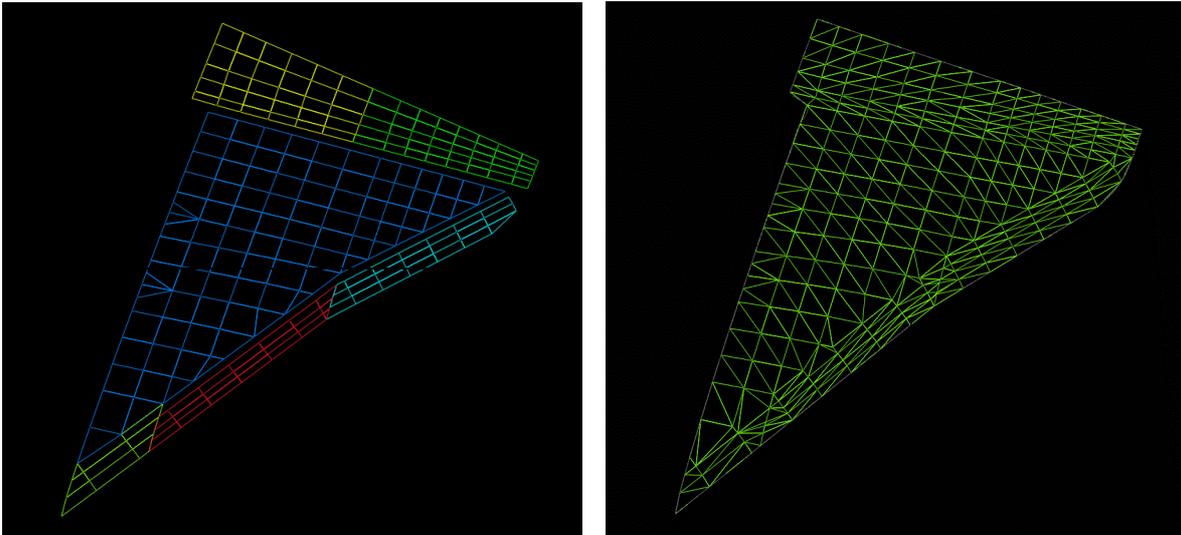
and integration of both pressure,  $p$ , and wall shear stress,  $\bar{\tau}_F$ ,

$$\bar{F}_{a,j} = \int_{\Gamma_F} (-p\hat{n} + \bar{\tau}_F) b_j ds \quad (7)$$

Loads on the CFD grid points are then transferred into the structural nodes for all grid points “paired” with a specific structural element at a time. The transformation matrix  $\underline{N}$  of equation (4) is built using the linear base functions of the respective structural finite element (refer to Farhat et. al.<sup>10</sup> for details). A distribution of residual loads is not implemented in this approach, since the problem of non-over-

lapping structural model and aerodynamic model sub-domains in the neutral interface is avoided.

The additional mesh between the projected structural nodes is generated by Delauney-triangulation closing by this all “gaps” between different parts of the structure, see right-hand side of Figure 2



**Figure 2: Original and triangulated structural model**

## 5 Aeroelastic Processes

Most commonly, the structural model yields a linear relationship between loads and deflections. Traditionally, also the relationship between aerodynamic grid deflection and generated aerodynamic loads is assumed to be linear. The two systems can then be combined and solved simultaneously. In the more general case, the coupled system is represented by a non-linear system of ordinary differential equations in time:

$$\underline{M} \ddot{\vec{u}} + \underline{D} \dot{\vec{u}} + \underline{K} \vec{u} = \vec{F}_s(M, \alpha, \beta, \vec{\delta}, q_s, \vec{u}, \dot{\vec{u}}) \quad (8)$$

with the structural mass, damping, and stiffness matrices  $\underline{M}$ ,  $\underline{D}$ , and  $\underline{K}$ , respectively, the vector of structural degrees of freedom (elastic deflections and rigid body motions, both functions of time),  $\vec{u}$ , Mach number,  $M$ , angle of

attack,  $\alpha$ , yaw angle,  $\beta$ , vector of control surface deflections,  $\vec{\delta}$ , and the stagnation pressure,  $q_s$ . If inertial loads from control surface motions are neglected,  $\vec{F}_s$  is the solution of the flow problem, which is performed with the tool described in section 3, transformed into loads on the structural nodes as outlined in section 4. Three separate computer programs are therefore used to solve equation (8). The process that governs the sequence in which they are executed depends on the specific problem. It requires additional process control software, and determines the total computational effort and wall clock time required to solve the task.

### 5.1 Static Aeroelasticity

In the static case, from a structural point of view, equation (8) reduces to a non-linear system of equations for the structural displacements,  $\vec{x}$ , which can be solved by iteration.

Starting with the flow field around, and the aerodynamic loads on the rigid body, the inherent computational process is a sequential execution of structural solution, transformation

of deflections, aerodynamic solution, transformation of loads, and the structural solution (Figure 3a).

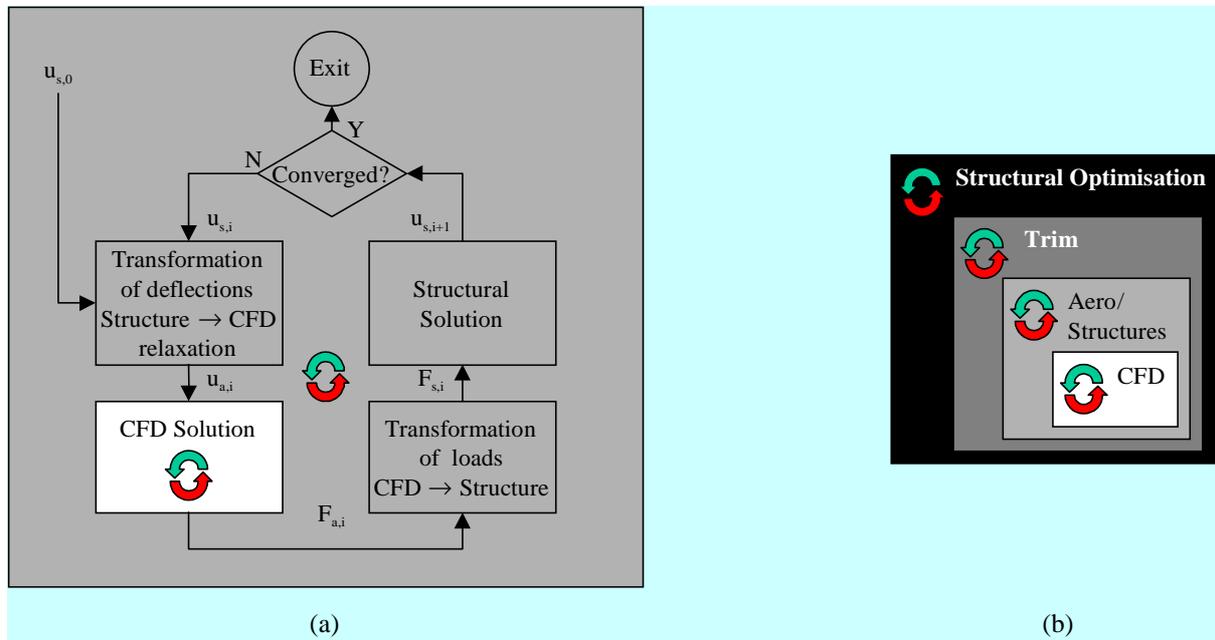


Figure 3: Iteration processes in static aeroelasticity and aeroelastic optimisation

The solution of the flow problem by CFD is an iterative process by itself, however, previous research<sup>11,12</sup> and experiences made in the course of this work indicate that CFD solutions do not need to converge “completely” to yield reasonable loads for the next aero-structures loop. Raveh and Karpel<sup>12</sup> have also shown that updating trim settings by linear predictions every few aero-/structures loops did not cause a major decrease in reduction of the CFD residual either, compared to a “rigid, untrimmed” CFD calculation of the same configuration - or when the structural dimensions were optimised every few trim updates in order to accommodate for changing design loads<sup>13</sup> (Fig. 3b).

### 5.2 Dynamic Aeroelasticity

Modal transformation is a useful tool for reducing the size of dynamic problems. In dynamic aeroelasticity, it also allows for an elegant way of solving the equations of motion (8) for the coupled system’s eigenvalues. With the matrix of  $N$  desired eigenvectors of the

structural system (without damping),  $\underline{X}$ , commonly associated with the  $I$  lowest eigenvalues,  $\omega_i$ ,  $i = 1, \dots, I$ , the modal basis of possible structural displacements,  $\vec{u}$ , in terms of the modal or generalised co-ordinates,  $\vec{q}$ , is formed by

$$\vec{u} = \underline{X} \vec{q} \tag{9}$$

When structural damping is neglected, the equations of motion are pre-multiplied by  $\underline{X}^T$ , and orthogonality properties are used.

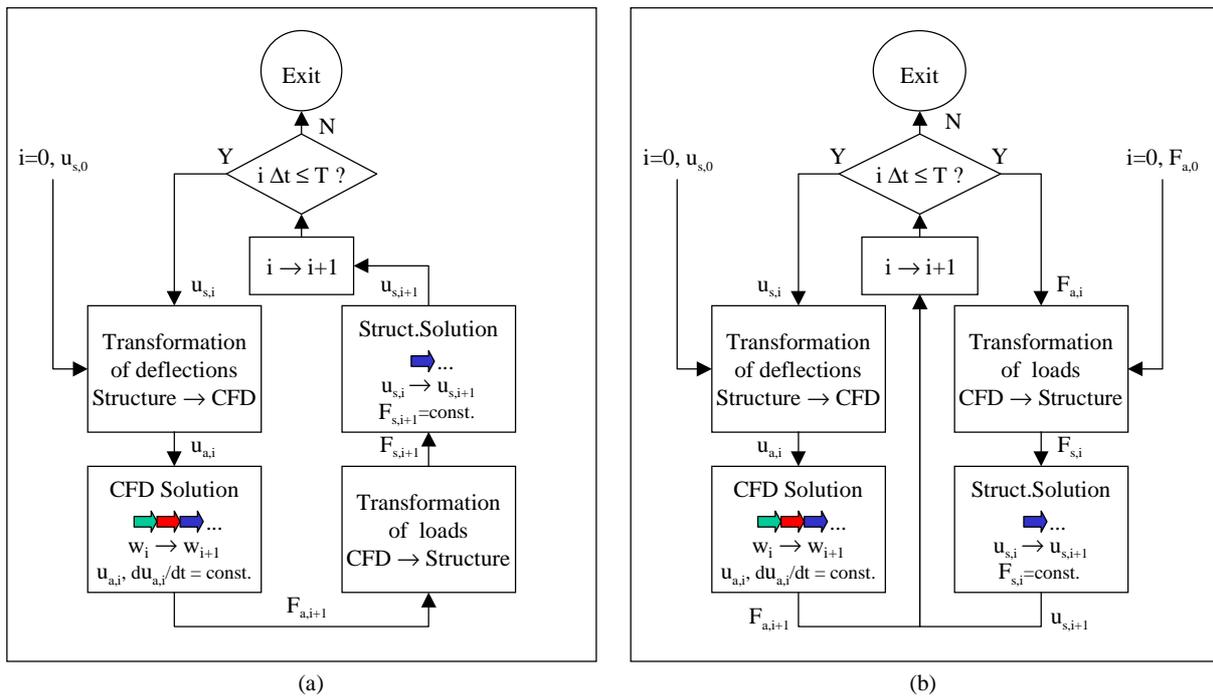
In the common case of Doublet-Lattice aerodynamics<sup>14</sup>, unsteady loads on aerodynamic grid points are functions of Mach number and the reduced frequency,  $k = (\omega b)/V_\infty$  (with the reference semi-chord,  $b$ , and freestream velocity,  $V_\infty$ ), and linear in the displacements,  $\vec{u}_a = \underline{TX} \vec{q}$ .

The  $\underline{AIC}$ , matrix of Aerodynamic Influence Coefficients, needs to be supplied for a range of Mach numbers and reduced frequencies. Solvers for calculating the flutter

speed using this approach are included in NASTRAN<sup>15</sup>, ASTROS<sup>16</sup>, and of course in LAGRANGE<sup>3</sup>.

Solutions of equation (8) require direct interaction of structural and flow solvers. Simultaneous solution of the sub-problems is impossible due to the separation into three software tools (see section 5.1). Hence, partitioned solution algorithms are needed to control the order of execution. **Figure 4** shows

two candidates, adapted from Farhat and Lesoinne<sup>17</sup>. In both flow charts, CFD and structural solvers advance the solution by identical physical time steps,  $\Delta t$ , from given initial conditions of structural deflections or aerodynamic loads, through the total time interval under observation,  $T$ . The states of the flow problem are denoted by  $w$ ; multiple arrows within the CFD solution boxes indicate sub-cycling for each physical time step.



**Figure 4: Conventional serial and parallel staggered algorithms, CSS and CPS**

Following Reference 17, **Figure 4a** is the conventional, serial staggered approach (CSS), where the CFD solution at time step  $i+1$  is fed into the structural solver to yield its solution for time step  $i$ . Because CSS was shown to be only first order time-accurate, either very small time steps or full sub-cycling in the CFD solution are required for appropriate accuracy. Since both alternatives increase the computational effort, an improved serial staggered (ISS) algorithm was derived<sup>17</sup>.

Figure 4b is a simple parallel version for saving wall clock time. This straightforward approach also shows accuracy limitations and an improved version still requires time steps being three times smaller than those of the ISS to

exhibit comparable convergence<sup>17</sup>. Hence, advantages from parallelism are largely offset by limitations in accuracy, and the serial ISS currently looks more promising.

### 5.3 Process Control Architecture

In order to control aeroelastic computations, to allow for the use of serial and parallel algorithms, and to permit extension to additional analysis types like trim or structural optimisation, the architecture of the process-control software must not impose any restrictions, neither on the process structure nor on the location of the analysis tools on a heterogeneous network of workstations. A prototype with the operational components

Module, Feeder, and Executive was developed to suit these requirements. A central data repository serves for storing versions of files exchanged between Modules. Module data repositories in each host's file system are used for communication between Module and process control.

The complete technology was validated for a generic three-Module problem with nested loops. The aeroelastic process consists only of two Modules, "struc" and "aero". They contain the respective solvers, fluid-structure interaction tools TRAEST and GridCAD, respectively, and smaller utilities. Figure 5 illustrates the

execution of a static aeroelastic loop (Figure 3a) or the CSS algorithm (Figure 4a). Due to autonomous operations of Modules and Feeders the process can be triggered by providing at least one input data set, here the initial displacements (a). Completion of "aero" results in storing aerodynamic loads in the central process data repository, P, which represents input for "struc" (b). If the new displacements are available, they are placed in P (c), which again leads to execution of "aero" (a), and so forth.

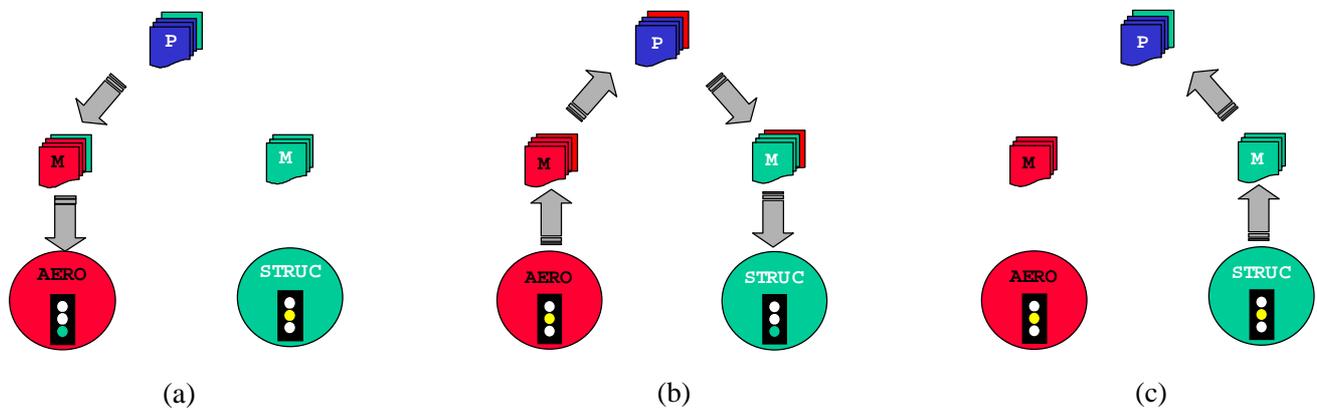


Figure 5: Serial aeroelastic process

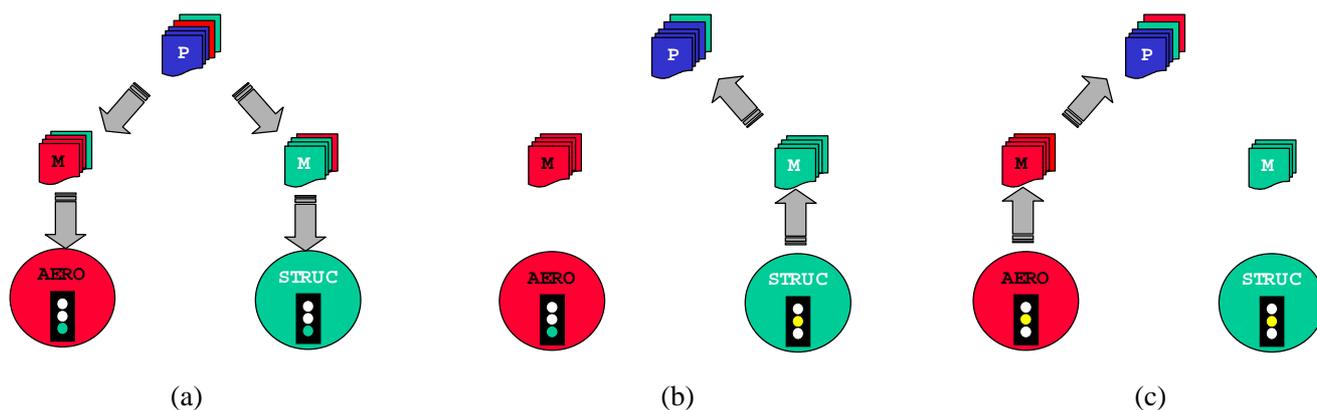


Figure 6: Parallel aeroelastic process

A parallel process, like the CPS in Figure 4b, is started by placing complete input data sets for both Modules into P (Figure 6a). Usually "struc" will terminate first and return to idle (b). This status lasts until "aero" completes its task

(c). Now, both Modules have new data to process (a). Hence, Module and Feeder functionalities result in a synchronisation of the two parallel branches.

### 5.4 Simulation Results

First validation of the process elements and their integration was carried out for static and dynamic aeroelastic simulations on the AGARD 445.6 weakened model No. 3<sup>18</sup>.

In the following, results are provided for the wing of the X-31A research aircraft wing with two leading and two trailing edge control surfaces, gaps between these and the wing box, and a large extrapolation area at the root which is not modelled structurally. In order to trigger control surface flutter and to investigate the behaviour around the flutter speed, the trailing edge surface actuators were set to 1% of their original stiffness (weak model - "weich" in Fig. 7). The processes were in all cases serial (CSS in the dynamic case), fluid-structure interaction was performed by GridCAD, and the flow was taken to be inviscid (utilizing an Euler approach). The time step in all dynamic simulations was set to  $\Delta t = 10^{-3}$  seconds.

#### 5.4.1 Subsonic Speed ( $M=0.8$ , $\alpha=10^\circ$ )

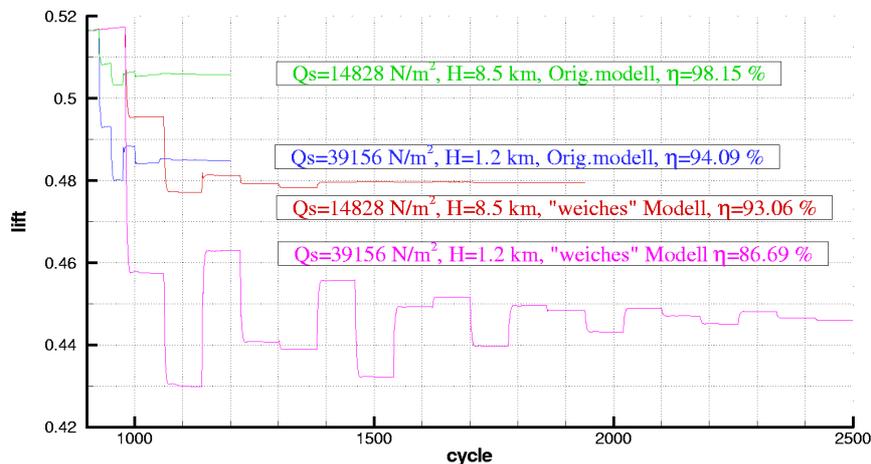
Two stagnation pressures were applied at subsonic speed,  $q_s = 14828 \text{ N/m}^2$ , related to an altitude of  $H = 8500\text{m}$ , and  $q_s = 39156 \text{ N/m}^2$

which is equivalent to  $H=1200\text{m}$ . Convergence of the lift coefficients for both the original and weak model and the two stagnation pressures is shown in on the basis of 900 multi-grid cycles of the CFD solver on the rigid wing and the effectiveness,  $\eta$ , describes the quotient of the lift coefficient for elastic and rigid body.

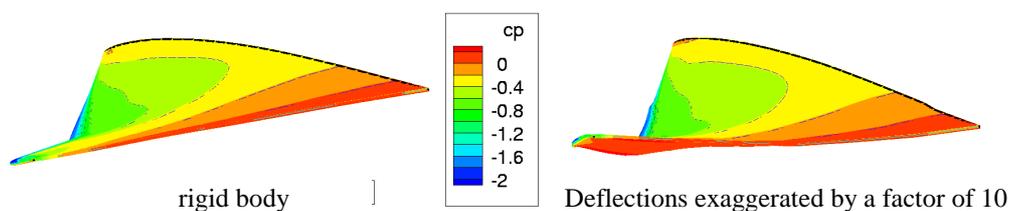
For the lower stagnation pressure, convergence of the residual, indicating a stable flow situation, is shown in Figure 7. The corresponding deflections are provided in Fig. 8, together with pressure distribution on the rigid and the deflected wing.

Fig. 9 depicts the transient response for the original and the weak model. It can be easily recognised that the weak model becomes unstable at  $q_s=39156 \text{ N/m}^2$ , and appears to be very close to flutter speed at  $q_s = 30000 \text{ N/m}^2$ .

Dynamic simulations are based on initial flow results obtained for the non-deflected wing with a converged flow field about the rigid configuration.



**Figure 7: Lift coefficient convergence,  $M=0.8$ ,  $\alpha=10^\circ$**



**Figure 8: Deflections for the original wing at  $M=0.8$ ,  $\alpha=10^\circ$ ,  $q_s = 14828 \text{ N/m}^2$**

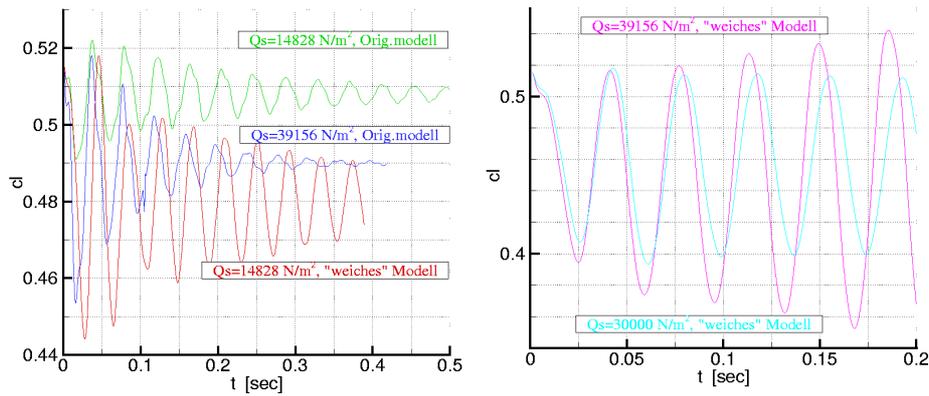


Figure 9: Transient response,  $M=0.8$ ,  $\alpha=10^\circ$

The original model is stable for both investigated stagnation pressures as it can be detected from Figure 9.

5.4.2 Supersonic Speed ( $M=1.3$ ,  $\alpha=7.5^\circ$ )

For the supersonic static aeroelastic test cases, convergence to steady state required much less cycles than in subsonic flow, although the residual is more than an order of magnitude larger. Within only six coupling cycles, each including five CFD multi-grid sub-cycles, a reasonably stationary solution was reached. Results for deflections and surface pressure are presented in Figure 10.

In contrast to the clearly identifiable stability properties in subsonic flow, the transient response of the lift coefficient at  $M = 1.3$  shows very irregular behaviour with slowly decreasing or slightly increasing amplitudes for the original and weak model, respectively. Results for the lift coefficient are given in Figures 11 and 12 together with the residual for the CFD computations. The latter indicates that a reduction of the error norm per time step of more than one decade has been achieved and, at the same time, that the overall residual is constantly decreasing when oscillations are damped out.

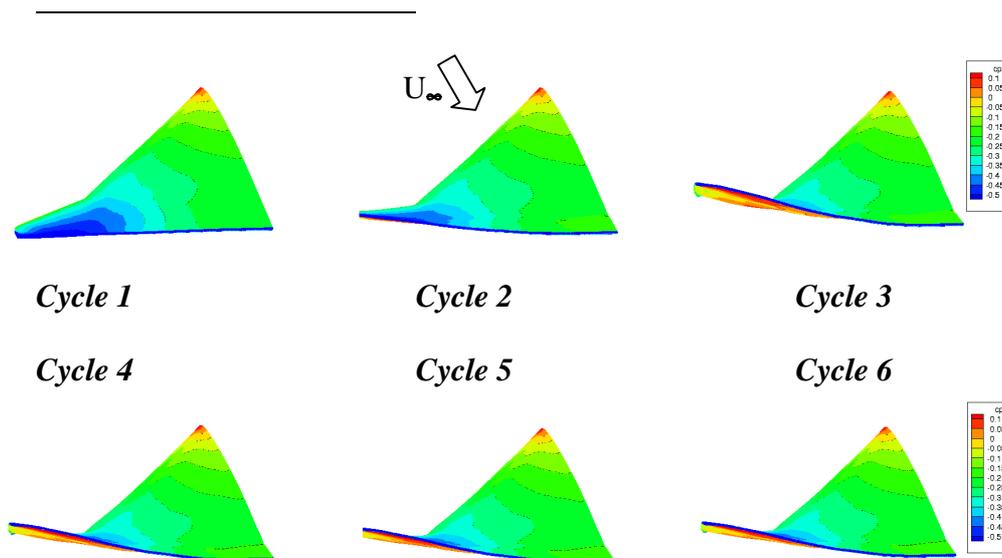
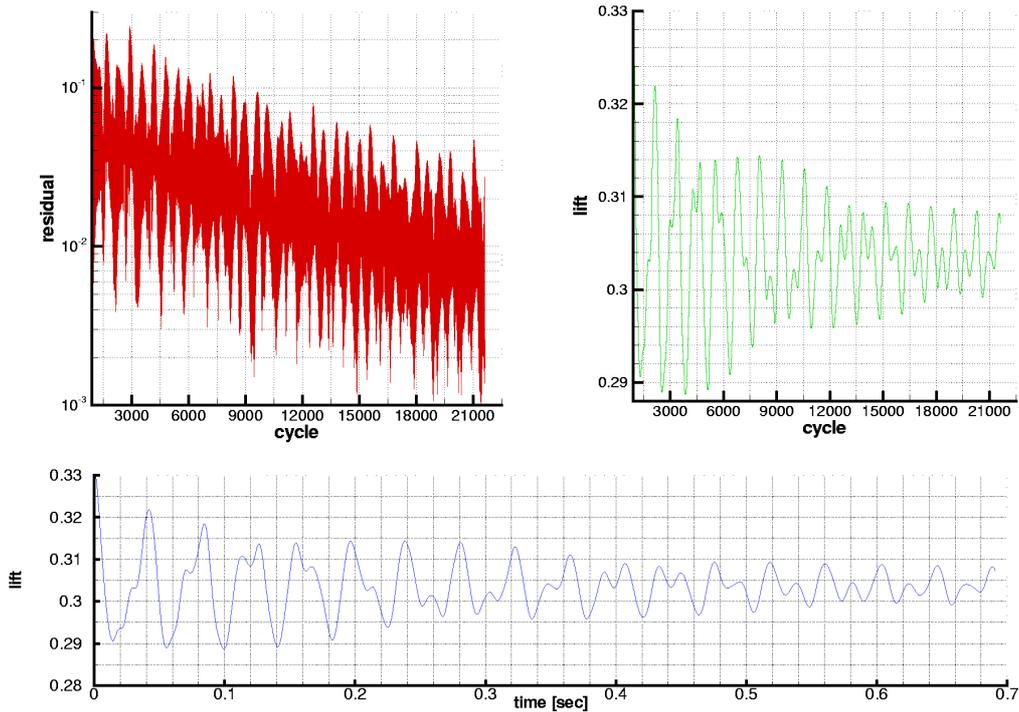
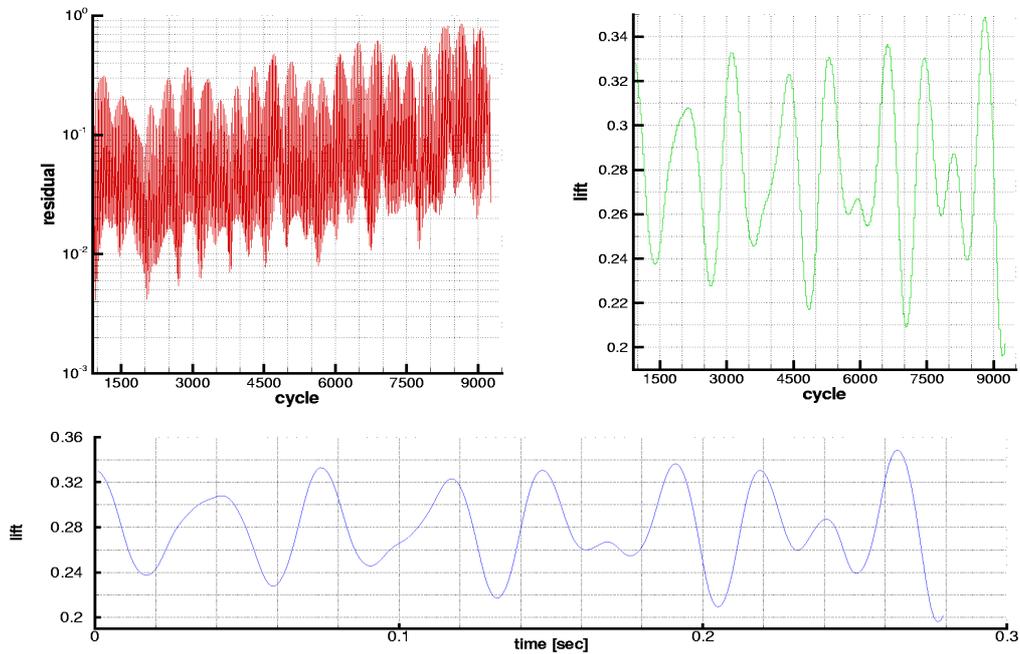


Figure 10: Convergence and deflection sequence, original wing,  $M=1.3$ ,  $\alpha=7.5^\circ$ ,  $q_s = 39156 \text{ N/m}^2$



**Figure 11: Transient response, original model,  $M=1.3$ ,  $\alpha=7.5^\circ$**



**Figure 12: Transient response, weak model,  $M=1.3$ ,  $\alpha=7.5^\circ$**

## 6 Conclusions

The present work has briefly described methods in use at Dasa-M for solving aeroelasticity problems and applications. The developed

technology efficiently couples complex and non-linear aerodynamics and structural dynamics, together with an already operational for assessing partitioned solution procedures. Next steps in this process will focus on improving

user-friendliness and scope, particularly to full configurations with stores and large deflections of control surfaces and of the elastic structure. Research along these lines has been initiated already.

Turnaround times are seen to be still a pitfall. CFD computations still take most of the time, and the focus must certainly be to make improvements here in particular simulation of trimmed and controlled aircraft performing manoeuvres of multiple seconds of duration are concerned.

The most challenging task of all is still to bring CFD into aeroelastic structural optimisation. For investigations where optimisation plays a major role, flutter constraints are more efficiently modelled using small disturbance CFD codes as non-linear transient loads, e.g. gust response, require too much computing time to be included in traditional optimisation approaches. Although in recent years, promising techniques have emerged in the area of multidisciplinary optimisation, such investigations are another focus of future work

## References

- [1] C. Weishäupl and B. Laschka, "Nonlinear and Small Disturbance Euler (SDEu) Calculations for Delta Wing Flap Efficiency," 38<sup>th</sup> Aerospace Sciences Meeting & Exhibit, AIAA-2000-0124.
- [2] M. Lesoinne and C. Farhat, "Re-engineering of an Aeroelastic Code for Solving Eigenproblems in all Flight Regimes," *ECCOMAS 98*, 1052-1061, Wiley (1998).
- [3] J. Schweiger, J. Krammer, and H.R.E.M. Hörnlein, "Development and Application of the Integrated Structural Design Tool LAGRANGE," 6<sup>th</sup> AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA-96-4169.
- [4] N. Kroll (ed.), "FLOWer Installation and User Handbook," DLR, Institut für Entwurfsaerodynamik, July 1998. MEGAFLOW-1001.
- [5] K. Becker, N. Kroll, C.C. Rossow, and F. Thiele, "MEGAFLOW – A numerical Flow Simulation System," ICAS Congress 1998, ICAS 98-2.7.4.
- [6] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne, "Torsional springs for two-dimensional dynamic unstructured fluid meshes," *Computer Methods in Applied Mechanics and Engineering*, **163**, 231-245 (1998).
- [7] A. Eberle, "A Finite Element method for re-distribution of grid points in moving, unstructured volume grids", Dasa-AMANDA-TN-003, 1998.
- [8] M.J. Smith, D.H. Hodges, and C.E.S. Cesnik, "Evaluation of Computational Algorithms Suitable for Fluid-Structures Interaction," *Journal of Aircraft*, **37**, 282-294 (2000).
- [9] H.-J. Schneider and M. Schneider, "GridCAD User Guide Load and Motion Algorithm," Beaver-Discovery, December 1999.
- [10] C. Farhat, M. Lesoinne, P. LeTallec, "Load and Motion Transfer Algorithms for Fluid/Structure Interaction Problems with non-matching discrete Interfaces: Momentum and Energy Conservation, optimal Discretization and Application to Aeroelasticity," *Computer Methods in Appl. Mech. and Eng.*, **157**, 95-114 (1998).
- [11] C. Farhat, and M. Lesoinne, "On the Accuracy, Stability, and Performance of the Solution of Three-Dimensional Nonlinear Transient Aeroelastic Problems by Partitioned Procedures", 37<sup>th</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, AIAA-96-1388.
- [12] D.E. Raveh, and M. Karpel, "Nonlinear Design Loads for Maneuvering Elastic Aircraft," *Journal of Aircraft*, **37**, 313-318 (2000).
- [13] D.E. Raveh and M. Karpel, "Structural Optimisation of Flight Vehicles with Non-linear Aerodynamic Loads," 7<sup>th</sup> AIAA/USAF/ NASA/ISSMO Multidisciplinary Analysis and Optimization Symposium, AIAA-98-4836.
- [14] E. Albano and W.P. Rodden, "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows," *AIAA Journal*, **7**, 279-285 (1969).
- [15] W.P. Rodden and E.H. Johnson, "MSC/NASTRAN Version 68 Aeroelastic Analysis User's Guide," MacNeal-Schwendler Corp., California, USA, 1994.
- [16] D.J. Neill, E.H. Johnson, and R.A. Canfield, "ASTROS: A Multidisciplinary Automated Structural Design Tool," *Journal of Aircraft*, **27**, 1021-1027 (1990).
- [17] C. Farhat and M. Lesoinne, "Higher-Order Staggered and Subiteration Free Algorithms for Coupled Dynamic Aeroelasticity Problems," 36<sup>th</sup> Aerospace Sciences Meeting & Exhibit, AIAA 98-0516.
- [18] E.C. Yates, jr., "AGARD Standard Aeroelastic Configurations for Dynamic Response: I- Wing 445.6," North Atlantic Treaty Organization Advisory Board for Aerospace Research and Development, AGARD-R-765. ISBN 92-835-0463-1 (1988).