M. Errera, A. Dugeai, Ph. Girodroux-Lavigne, J.-D. Garaud, M. Poinot, S. Cerqueira, G. Chaineray
(Onera)

E-mail: marc.errera@onera.fr

Multi-Physics Coupling Approaches for Aerospace Numerical Simulations

The purpose of this paper is to present coupling strategies for aerospace numerical calculations. In the first part, the basic approach used relies on the partitioned coupling of a finite-volume Navier-Stokes solver and a finite-element solid code. These two separate and independent simulation tools carry out exchanges via a coupling library. Two different applications illustrate the capabilities of this coupling method. The main advantage of this approach is to benefit, for each software application, from the experience developed by a large number of specialists over many years. In the second approach, mono-disciplinary software is extended to multi-physics modeling, by adding new simplified modules for other disciplines and by implementing specific coupling algorithms. The modeling of aeroelastic systems is presented as well as several applications to demonstrate the capabilities of this method. Finally, a software structure for code coupling is described in the third part. It consists of an Open System approach, based on a powerful open source assembly of public interfaces.

Introduction

Coupled problems

The adjective "coupled" appears frequently in the literature, in different contexts and sometimes with different definitions. In this paper, we have adopted the definition proposed by Zienkiewicz [1]:

"Coupled formulations are those applicable to multiple domains and dependent variables, which usually describe different physical phenomena in which:

- neither domain can be solved separately from the other
- neither set of dependent variables can be explicitly eliminated"

At this stage, it is convenient to mention that there are two distinct categories. In the first, domains may totally or partially overlap. In the second, they interact through a common interface. Both of these categories are considered in this paper.

Computational Fluid Dynamics (CFD) is the most sophisticated type of airflow model. It predicts the detailed spatial distribution of velocity, temperature and pressure, by solving the flow governing equations. CFD programs can provide detailed predictions of convective and temperature distribution, but need input information from the solid, such as deformations or temperature field. Computational Structural Dynamics (CSD) tools, on the other hand, are unable to give information on the airflow and convective characteristics (pressure, heat transfer coefficients, etc.) are usually empirical. Coupling CFD and CSD eliminates many of these assumptions, since the information provided by each model is complementary. It is therefore a very attractive solution, but robust and efficient interaction models between the fluid and the solid media are generally required. Furthermore, it may be very computationally expensive, if specific approaches are not used.

Partitioned strategy for fluid-structure coupling

Fluid-Solid coupling - Fluid-Structure Interaction (FSI) or/and Conjugate Heat Transfer (CHT) - can be achieved via two different methods. In the monolithic approach, the equations are solved simultaneously, i.e., they directly operate on the aggregated fluid and structure equations. On the contrary, in the partitioned solution approach, systems are spatially decomposed into partitions. This decomposition is based on physical considerations. The solution is separately advanced in time over each partition. The partitions interact with each other on a common physical interface. Partitioned techniques are widely popular, because they allow the direct use of specifically designed solvers for different fields and may offer significant benefits in terms of efficiency over the monolithic techniques. Moreover, smaller and better conditioned subsystems are solved, instead of one overall problem. All the cases presented in this paper use this partitioned technique.

In order to obtain coupled results on a level of detail generally provided by CFD, the solid and fluid must be solved in sequence (staggered strategy). Interaction effects are accounted for by transmission and synchronization of coupled state variables, the results on the coupled interface being compared and consequently corrected. Although the idea and principle of this code coupling concept is straightforward, this approach can be challenging in practice, due to three main discontinuities between fluid and solid models. First, there is a time-scale discontinuity due to the significant difference between the two media. Second, there is a space-scale discontinuity. The last discontinuity can be referred to as a "CPU discontinuity" and is due to the discrepancy between the memory requirement and the execution time, between a fluid and a solid solver.

Specific coupling treatments need to be developed to bridge these discontinuities: numerical algorithms, a search process, appropriate numerical approximations, coupling strategies and a computing environment. As a consequence, the radically different physics, length scales, dynamics and objectives involved in multidisciplinary aerospace problems require the use of a wide range of simulation multi-physics approaches.

Software architecture

A multi-physics problem has three layers to be overcome before an actual operating simulation is achieved on a computer. The first is the physical layer, the coupled system is decomposed into physical fields and their mathematical models are described by means of field equations. The second layer consists in the numerical treatment of this dynamical coupled system. Fields are discretized in space and time and an appropriate numerical algorithm must be determined. The third and last layer is the software, i.e., the numerical algorithm is programmed using one or more programming languages on a hardware system.

A complex simulation can involve several types of software, for example: an in-house code made by a single engineer, COTS[1] , or even Open Source software. The application writer picks up the codes he needs, makes an assembly of these codes using a Steering language and finally obtains a complex software system. The hardware platform for this simulation can be complex too, for instance a large cluster of Linux nodes connected to a remote workstation and these connections can be synchronous or asynchronous.

Multi-physics approaches

Each discipline has developed specialized tools, which must be adapted to multidisciplinary applications. The purpose of each application may be: purely static (static on both sides of the coupled interface), a mixed stationary/non-stationary solution, or it can be a complete transient solution. The most efficient numerical approach for each solution must be designed carefully and the resulting coupled interface treatment must guarantee stability, consistency and accuracy properties.

Coupling CFD and CSD is a very challenging approach and the goal of this paper is to present some particular aspects and global strategies applied to various specific aerospace problems. This paper is organized as follows:

In the first part, the basic approach relies on the partitioned coupling of a finite-volume Navier-Stokes solver (CEDRE code) and a finite-element solid code (Z-set code) is presented. This coupling is based on the use of an interface library. Two different applications illustrate the capabilities of this coupling method.

In the second part, the modeling of aeroelastic systems is presented. The aim of aeroelasticity is either to compute the behavior of the coupled system at equilibrium, or to determine the potential occurrence of instability of a fluid-structure dynamic system.

In the third part, a software architecture for code coupling is described. It consists of an Open System approach, based on a powerful open source assembly of public interfaces.

External coupling

Numerical strategy

The basic approach used in this section is the loose coupling of a finite-volume Navier-Stokes code and a finite-element solid code, a coupling library being in charge of transferring information from one code to the other. In this approach, independent models are simulated separately and the fluid-solid interaction is achieved by partitioning the problem into fluid and solid parts, solved separately with boundary conditions calculated by the other part. This leads to a sequential treatment that can be seen as a Conventional Serial Staggered procedure. During the coupling process, these solvers are called alternately. Specific algorithms are then required, specifically in transient problems [2]. Only this approach allows a direct extension to general multidisciplinary problems.

The flow solver

The computer code in the fluid, known as CEDRE, can compute turbulent and reactive flows of realistic aerospace configurations and is widely used for a great variety of scientific and engineering problems: turbojet, ramjet, solid propellant rocket and cooling circuits. This code is a three-dimensional finite-volume unstructured code. The governing equations are the time-dependent Navier-Stokes equations, which express the conservation laws of mass, momentum, total energy and any other scalar quantity, written in the conservation form. Details about the Navier-Stokes solver can be found in another papers [3], [4] of this issue.

The solid solver

The computer code in the solid, known as Z-set, is a three-dimensional finite-element code. It is an advanced object-parallel code for structural mechanics, with many non-linear solution capabilities. More details about this code are given in Box 1.

The coupling library

In order to reduce the effort involved in coupling the two codes, a loose coupling approach has been chosen and the exchange of physical data between the finite-volume fluid code and the finite-element solid code is performed through the MpCCI library [5]. It is a code coupling interface for multidisciplinary applications. Instead of just transferring data from one process to another, MpCCI takes into account the grids as well as the processes on which the data is located. Due to the different discretization techniques for the CFD and CSD simulations, the fluid-solid interface is represented by different grids. The task of MpCCI is to calculate the neighborhood relations when these grids are non-matching and to transfer the coupling values across the interface. This library has allowed us to couple pre-existing physics applications with a limited amount of modification and to continue to develop them independently.

1 Commercial Off The Shelf: software made and sold by a commercial company

Box 1 - Z-set: a Computational mechanics toolkit for material and structures

Z-set is a versatile toolkit developed by the École des Mines ParisTech, Onera and NWNumerics. Its development started in the early 80s to model the non-linear response of material specimens subjected to various loads. Initially specialised in highly non-linear material models, it has evolved over the years and is now a full-featured state-of-the-art finite element solver.

Initially written in Fortran77, it was entirely re-engineered during the 90s and is now completely developed in an object oriented framework, using C++ as the implementation language.

It addresses the entire modeling chain in the field of computational mechanics and thus contains pre and post graphical and batch processing, meshing tools, optimisation tools, sequential and parallel finite element solvers, a large material library, etc. Furthermore, its multithreaded core components take full advantage of modern SMP architectures.

The reader is invited to look at http://www.zset-software.com to get a complete overview of Z-set's features. It is possible to model more or less any structural mechanical problem, even using innovative modern methods such as parallel processing, extended finite elements, adaptive re-meshing and non-local models, etc.

The Z-set design is based on a dynamic object factory pattern, coupled with a plug-in technique, which allows the end-user to modify and enrich the package without any modification to the core code. For instance, the coupling methods described in this issue have been entirely developed within a plug-in, *without* any modification to the main code.

Two different space applications illustrate the capabilities of this coupling method. First, a fluid-solid coupling interaction in a solid propellant rocket motor is presented and then an aero-thermo-mechanical study in the Vulcain 2 rocket engine. The latter involves three independent solvers.

Fluid-structure interaction for a solid propellant rocket motor

Fluid Structure Interactions (FSI) have been shown to be extremely important in the analysis of pressure oscillations in Solid Propellant Rocket Motors (SPRM) (see Box 2). To improve the physical understanding of pressure oscillations inside SPRM, a coupling methodology involving two codes, CEDRE and Z-set, has been developed for the flow and structure subsystems respectively.

Partitioned approach

In this coupled study, the FSI is achieved by partitioning the problem into fluid and solid parts, solved separately with interacting boundary conditions [6]. Figure 1 shows the sequence of iteration steps. It starts with the calculation of the aerodynamic field (Path 1).

The resulting pressure distribution is transferred to the finite element nodes (path 2). Using this new interface conditions the structural code and computes the deformation of the structure (path 3). The resulting displacements modify the fluid surface grid and consequently change not only the boundary conditions (path 4), but also the entire grid in the fluid domain in the next step (path 5).

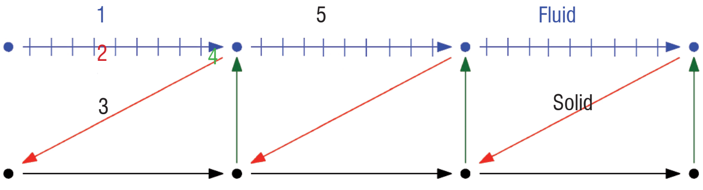


Figure 1 - Subcycled Conventional Serial Staggered procedure for FSI.

Two-Dimensional problem in the field of solid propulsion

The potential of the coupling strategy in handling two-dimensional transient FSI problems is assessed by computing an experimental case, performed by the IUSTI laboratory [7]. Such an experimental device describes a flexible panel protruding into a shock tube and submitted to a shock wave. A close-up view of the experimental set-up is given in figure 2.

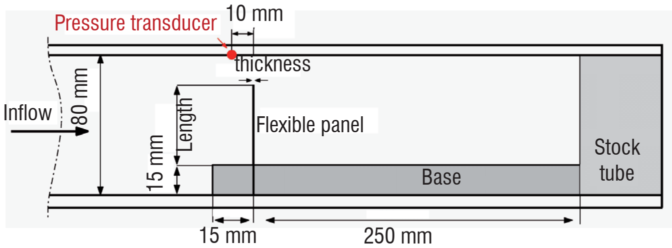


Figure 2 - IUSTI experimental setup.

The panel is fixed on a base, assumed to be infinitely rigid. As a shock travels down the tube, pressure gradients result in a panel motion. In this experiment, fluid pressure evolution is measured by a pressure transducer and top panel displacement is monitored. Furthermore, ombroscopic pictures provide visual information on the behavior of the transmitted and reflected shocks.

Numerical model

The coupled simulation is performed considering an isotropic steel panel (linear elastic), with a Young's Modulus E=220 GPa and a density ρ =7600 kg.m^{-3}, with a length equal to 40 mm and 1mm thickness. The shock wave moves from the inlet boundary condition, where air is injected at standard atmosphere conditions, at a Mach number of 1.2. Considering the short duration of the experimental run, turbulence is neglected. For such a compressible flow, Rankine-Hugoniot relations allow the determination of pressure and temperature values related to the shock wave. The coupling time step is taken to be equal to 10^{-6} s, which is about 100 times smaller than the structure characteristic time. Mesh resolution and time step have been chosen in order to ensure a sufficient numerical accuracy of the coupled dynamics.

Box 2 - Pressure oscillations in solid rocket engines

Early to be discovered in the late 1930's, unsteady motion in Solid Propellant Rockets were the first examples of combustion instabilities in propulsion systems. Since that time, extensive work has been done on pressure oscillations arising in a number of large Solid Propellant Rocket Motors (SPRM), including the Space Shuttle RSRM, Titan and Ariane 5 boosters. Reason for such intensive concern can be found in significant vibrations transmitted to the payload.

Several mechanisms had been identified as the chief origin of instabilities arising in solid-fuel motor configuration. Those mechanisms involve the sensitivity of burning surfaces to pressure and velocity fluctuations and purely fluid mechanical processes leading to the interaction of large scale vortices with acoustic waves B2-[1].

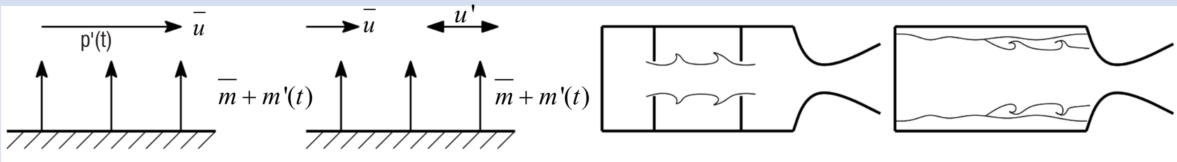


Figure B2-01 - Possible mechanisms for pressure oscillations in SRM

Among all the potential causes of flow unsteadiness, shear layers caused by protruding inhibitors (as a result of the propellant consumption) or by the geometry of the grain and the unstable behaviour of the flow induced by wall injection are of major importance B2-[2]. As a consequence, computational efforts made to better understand pressure oscillations inside SRM require, for instance, the accurate prediction and modeling of inhibitors and/or propellant mechanical response under fluid stress. Coupling physical models through independent physics solvers can be now achieved thanks to multi-physics numerical approaches.

References

B2-[1] G.A. FLANDRO and H.R. JACOBS – *Vortex Generated Sound in Cavities*. AIAA Paper 73-1014, 1973

B2-[2] F.E.C. CULICK - *Unsteady Motions in Combustion Chambers for Propulsion Systems.* NATO Research and Technology Organisation, December 2006.

Numerical results and comparison with experimental data

The interaction between the shockwave and panel gives rise to a transmitted and a reflected shockwave. A detailed description of the physical phenomena involved is discussed in detail by Giordano [7].

As shown in figure 3, numerical schlierens present a good agreement with experimental ombroscopic pictures and thus indicate that the flow field dynamic is being quite well captured.

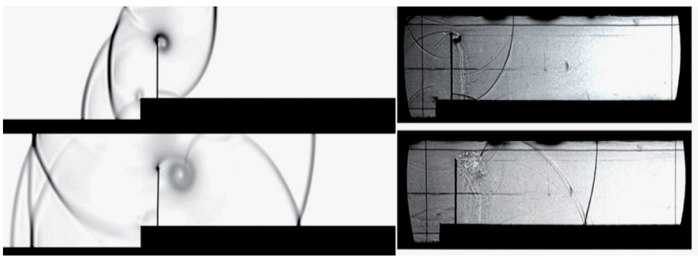


Figure 3 - Numerical schlierens (left) and experimental ombroscopic pictures (right)

A plot of the pressure at the position of the experimental transducer is reported in figure 4, paying particular attention to the pressure time evolution. Although there is a slight difference for t > 0.0025s, probably due to some boundary condition reflection error for the reflected shockwave, the two plots show a similar transient evolution. As a result, from fluid dynamic loads, the panel movement (figure 4) is also well captured and falls within experimental uncertainties. All these comparisons between numerical and experimental data emphasize that the coupling method succeeds when computing highly transient phenomena and thus should be used to cope with more elaborate computations involving stronger coupling phenomena.

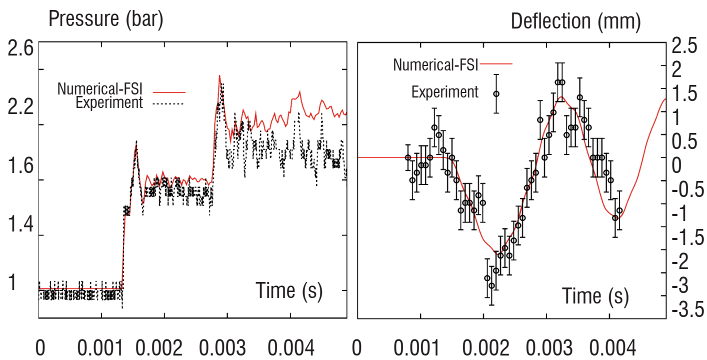


Figure 4 - Pressure evolution at probe location (left) and corresponding panel deflection (right).

Simulation of a rocket nozzle

Vulcain 2

The nozzle of a Vulcain 2 engine is subjected to many types of solicitations. It is assembled from *Inconel 600* tubes, a material resistant to high temperatures. Since it has to withstand the heat generated by the exhaust gases (~3000 K), it is cooled by Hydrogen circulation. The nozzle must withstand severe thermo-mechanical loads during pre-flight testing and during the actual flight. In non-reinforced nozzles, this cyclic loading causes progressive bulging of the tubes, which in turn modifies the circulation of both the exhaust and coolant fluids. The simulation of the complete system thus requires a fully coupled approach.

Model

The complete system is extremely complex, so this study will be restricted to a single tube portion, located near the critical zone of the nozzle (Figure 5).

Decoupled studies by Roos and Chaboche [8] have shown that a nonlinear elasto-visco-plastic material behavior is needed to properly model the bulging of the tubes; the authors have identified a Norton flow law:

$$\| \underset{\sim}{\dot{\varepsilon}}^p \| = \left\langle \frac{\| \underset{\sim}{\sigma} - \underset{\sim}{X} \| - R - k}{K} \right\rangle^n$$

where $\| \underset{\sim}{\dot{\varepsilon}}^p \|$ is the plastic strain rate, $\underset{\sim}{\sigma}$ is the stress tensor, $\underset{\sim}{X}$ is the kinematic hardening term, R is the isotropic hardening, k is the initial flow stress, and K and n are both Norton viscosity coefficients [9]. All parameters appearing in this behavior are strongly dependent on the temperature.

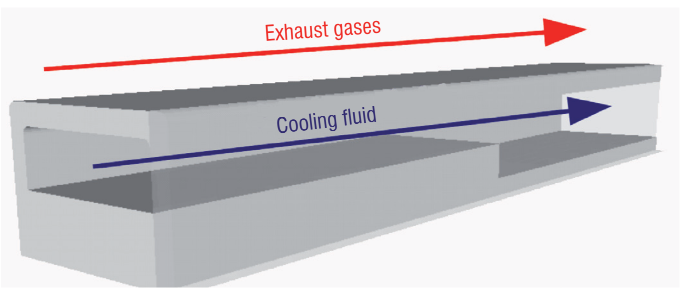


Figure 5 - Tube section model.

For this computation, the coolant fluid is considered a perfect gas; the partitioned approach will allow us to switch to more complex fluid models later on, without needing a full re-design of the coupling software. The exhaust gases are not simulated: a simple Robin thermal boundary condition is applied on the structure to represent them.

On the cold fluid side, both mechanical and thermal interactions are taken into account. Preliminary decoupled computations have shown that the deformation of the structure does not significantly influence the fluid flow, at least in this particular configuration. This part of the coupling was thus neglected and we kept only those shown in figure 6.

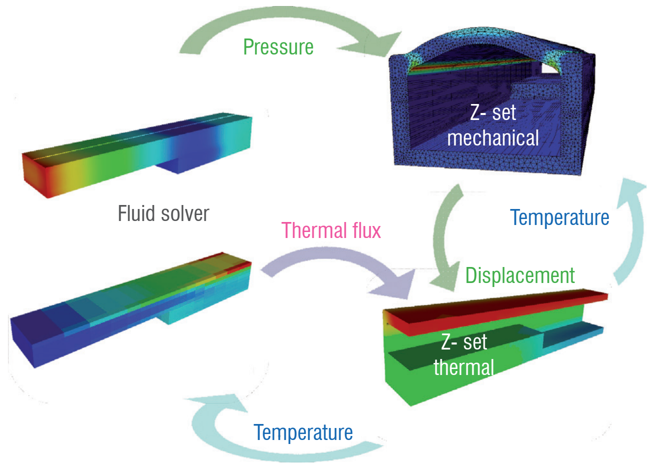


Figure 6 - Aero-thermo-mechanical coupling sequences.

Note that in this model, the mechanical and thermal responses of the structure are computed by two separate solvers, running on different meshes. It turns out that the same software (Z-set, see Box 1) is able to do both computations, but they are considered separate in the algorithm below.

Algorithm design for an aero-thermo-mechanical problem

The standard coupling algorithms [6] are adapted to this 2½ code coupling. Our experience has indeed shown that each application requires a specific tailoring of the algorithm, in order to benefit from the specificities of the problem. In this coupled problem, the main interest is in the response of the structure to a typical load and the coolant fluid is only modeled in order to obtain reasonable boundary conditions. The algorithm is thus designed around the structure, rather than around the fluid.

For the sake of simplicity, let us first consider the thermal coupling. The characteristic time of the structure is extremely large compared to that of the fluid. The algorithm considers the fluid as stationary around each of the configurations of the structure and standard coupling algorithms, such as those presented by [10], are adapted as follows:

```
While t < t_max :
    repeat:
        predict flux (or alpha) at interface
        solve solid thermal problem
        send temperature at interface from S to F
        solve fluid
        send flux (or alpha) at interface from F to S
    until prediction is good enough
    t ← t + Δt
```

Figure 7 - Coupling algorithm for a thermal problem: S stands for structure solver and F for fluid solver, α is the convection coefficient at the interface.

The algorithm presented in figure 7 features three specificities:

• it includes correction steps, because a fully explicit algorithm (e.g., one without correction steps) would require extremely small time steps to ensure stability;

• the coupling frequency is not fixed: each solver is allowed to adapt it during computation, depending on the evolution of the coupled variables;

• in order to further improve the convergence speed of this coupling algorithm, Robin boundary conditions are used at the interface; we refer to [10] for further details.

Let us now complete this algorithm to include mechanical coupling. There are two additional terms to take into account:
• the volumic thermo-mechanical coupling within the structure;
• the mechanical coupling at the surface between the fluid and the structure, as can be seen in figure 6.

It has been decided that the mechanical code should impose only its displacement variables on the thermal code and nothing on the fluid. The mechanical problem can thus run alongside the other two codes, without needing a full synchronization with them. The algorithm is shown in figure 8.

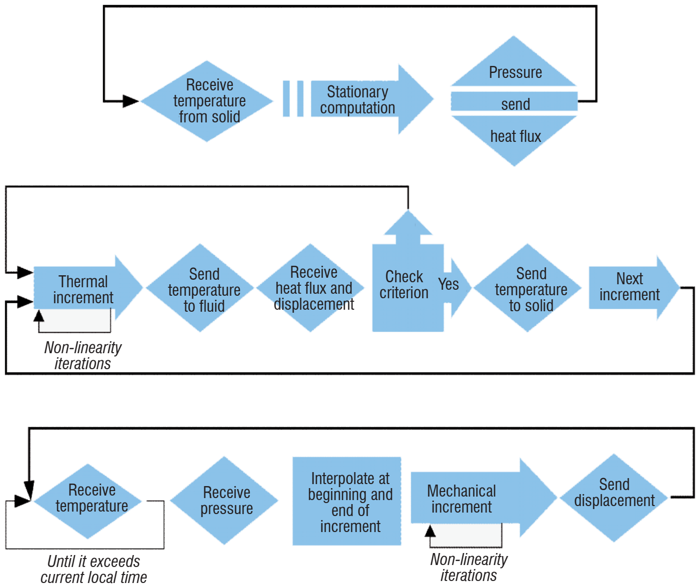


Figure 8 - Algorithm implementation in the fluid code (top), thermal code (middle) and mechanical code (bottom)

Results and performance

Our designed algorithm is applied to a 20 second simulation, corresponding to the first part of a test sequence. At this point, the thermal problem is mostly stabilized, but the mechanical problem will still continue to evolve due to creep for the rest of the test (600 s). Figure 9 presents the solution at the end of this first sequence. The asynchronous 3-code algorithm performs well on this application. Concerning the thermal part, the adaptive coupling time steps require a minimal number of coupling time steps: they are sufficiently small at the beginning of the computation when the thermal fluxes evolve rapidly and become progressively larger when it starts to stabilize. The use of Robin interface conditions between the structure thermal problem and the fluid also allows for larger time steps, without destabilizing the coupled process [10]. The use of asynchronism permits a reduced number of time steps for the mechanical solver: at the beginning of the computation, when the temperature is still low, it behaves elastically and accepts large time steps; however, when the thermal problem reaches higher temperatures, the mechanical problem enters a more complex, nonlinear stage, where it is principally governed by the visco-plasticity terms and requires small time steps. This type of model problem demonstrates the feasibility of a coupled resolution within the context of nonlinear mechanics, 3-code interactions and strongly coupled physics. Future work will include the simulation of more realistic models that can be confronted with experimental results and the extension to physics with less contrasted characteristic times.

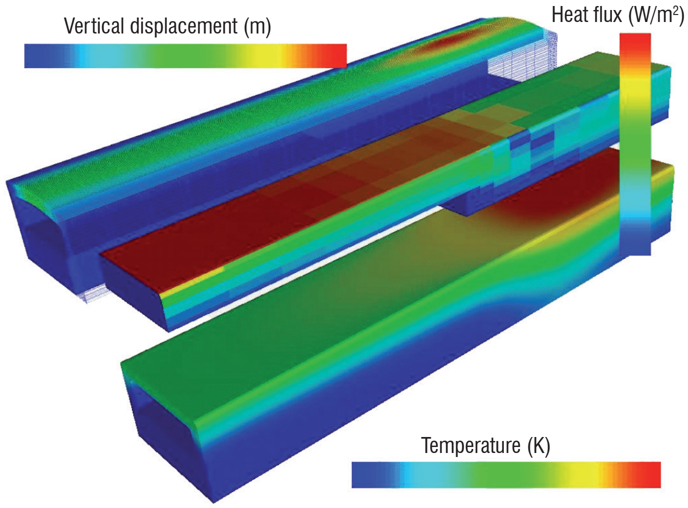


Figure 9 - Coupled aero-thermo-mechanical results: vertical displacement (top), heat flux (middle) and temperature (bottom) after 20s of simulation.

Aeroelasticity: introduction

Aeroelasticity is the scientific discipline that studies the coupled behavior of a mechanical system basically submitted to 3 kinds of loads: elastic forces, aerodynamic forces and inertial forces.

Figure 10 depicts the well known Collar triangle that details the various scientific domains dealing with the interactions between these 3 sources:

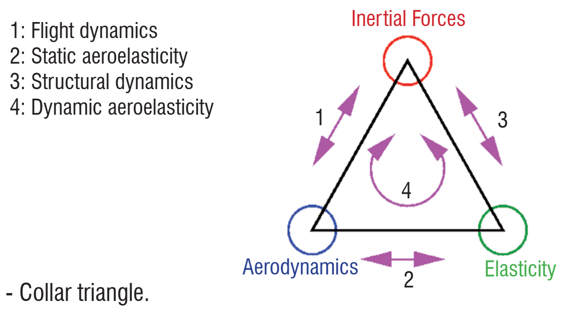


Figure 10 - Collar triangle.

The aim of aeroelasticity is, either in the static or in the dynamic case, to be able on the one hand to predict the behavior of a coupled system at equilibrium and, on the other hand, to investigate the potential occurrence of instability of the fluid-structure coupled system. At equilibrium, the objective is to predict the coupled structural deformations and stresses, and the aerodynamic loads. A proper evaluation of the structure deformations under operating conditions is necessary to precisely predict, and possibly optimize, the performance of the aeronautical system, aircraft lift and drag, for example. Aeroelastic instability may appear in the static case, leading to static divergence and in the dynamic case, with the so-called flutter phenomenon. Of course, the study of the instability conditions is of prominent importance, in order to ensure the flight-safety of aeronautical structures, due to the catastrophic explosive character of flutter. The coupled fluid-structure mechanical system may also be non-isolated and be subjected to additional external forces, such as gravity, forced excitation, or unsteady aerodynamic excitation (wake or gust response) and may be coupled with the flight control system, thus introducing additional complexity into the study of the coupled system behavior.

The modeling of aeroelastic systems

Structural modeling

For many years, flutter has been considered such a dangerous and badly understood phenomenon, that designers have tried to build stiff and rather compact aeronautical structures to get rid of it. Flight deformations were supposed to remain very limited. Therefore, for aeroelastic studies, the structural behavior is classically supposed to be properly represented by a linear, small deformation model. The dynamic behavior of such an aeroelastic structure is modeled by a differential equation in time, involving mass, damping and stiffness matrices, and external aerodynamic loads:

$$M\ddot{x} + D\dot{x} + Kx - F_A(t, x, \dot{x}) = 0$$

In the static case, the equilibrium is given by the following steady equation:

$$Kx_{stat} - F_A(x_{stat}) = 0$$

Several numerical models are possible. A finite element approach is more general, but modal reduction is frequently used. In this case, the solution may be projected on the basis of the modal eigenvectors Φ of the undamped system. The structural dynamics equation becomes reduced to:

$$\mu\,\ddot{q} + \beta\,\dot{q} + \gamma\,q - \Phi^t F_A(t) = 0$$

where q stands for the generalized coordinate vector,
$\mu = \Phi^t M \Phi$, $\gamma = \Phi^t K \Phi$, $\beta \approx \Phi^t D \Phi$ and $\Phi^t F_A(t)$ is the unsteady generalized aerodynamic force vector (GAF).

Today, aeronautical designers strive to benefit from the flexibility of the structure, to improve aircraft and engine performances. Aircraft lifting surfaces and engine bladings become thinner, wider and more flexible. Structural deformations tend to increase and the need for better modeling arises. It may be necessary to take into account structural non-linearities, such as friction, mechanical clearances, or even geometrical non-linearities due to large displacements. The tendency is to benefit from more sophisticated fluid and structural modeling via FSI with FE non-linear solvers.

Aerodynamic modeling

For many years, steady and unsteady linear methods have been the only tools available in the subsonic and supersonic regimes to analyze the aeroelastic behavior of aeronautical structures. Here again, improving the performance of aircraft requires a better understanding of the steady and unsteady aerodynamic operating conditions of the aeronautical structures. More sophisticated numerical aerodynamic models have been developed, especially in the transonic regime, where strong non-linear aerodynamic phenomena occur, such as shocks, boundary layers and flow separations. RANS and URANS solvers have been developed, first for steady applications, then in the unsteady domain, and these are now of common use, even for aeroelastic applications.

In the unsteady case, specific numerical features of the aerodynamic solvers are necessary for aeroelastic applications. The main issue is to be able to take into account the structural deformations. This requires:
- the availability of mesh deformation techniques;
- the taking into account of grid velocity;
- efficient and time-consistent unsteady resolution schemes;
- convergence acceleration techniques.

The solvers must handle static and dynamic RANS and URANS equations, with a variety of turbulence models. Due to the large CPU costs of aeroelastic computations, parallelization of the solvers has become compulsory today. *elsA*, which is currently Onera's multi-block structured grid aerodynamic solver, is the development tool chosen for aeroelasticity studies. More information about elsA is available in this issue.

Transfer techniques

One of the key issues for multidisciplinary simulation is to ensure the proper transfer of interacting variables between the various physical models involved. As far as aeroelasticity is concerned, the simulation must ensure:
- the proper load transfer, from the fluid domain to the structure;
- the proper displacement (and velocity) transfer, from the structure to the fluid.

Therefore, it is necessary to define a fluid-structure interface that allows data exchanges between both domains. However, it must be pointed out that, in the general case, the structural and fluid discretization grids at the interface are distinct. Specific transfer techniques must be implemented that totally depend on the physical models used for both domains.

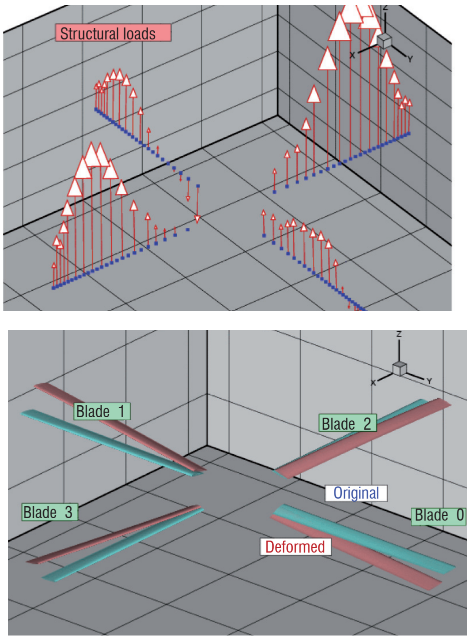
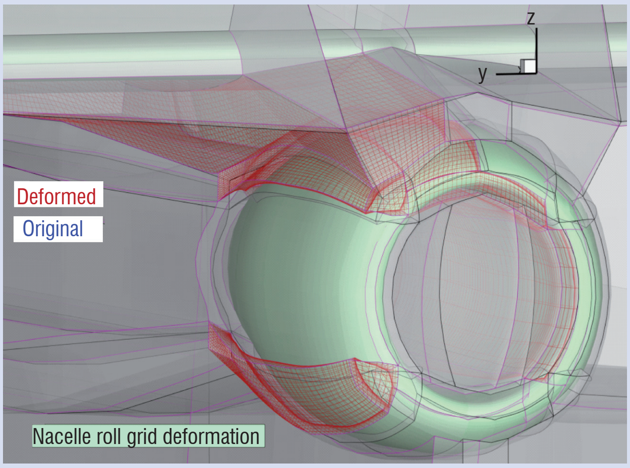


Figure 11 - Example of a beam structure model compared to aerodynamic surface grid (helicopter case).

Box 3 - Mesh deformation

Taking into account the motion of the structure is a general issue for numerical aeroelasticity purposes. A first approach is to keep the aerodynamic grid unchanged, using a specific aeroelastic boundary condition that takes into account wall velocity. This approach may be valid for inviscid two-dimensional computations, but appears to be unsuccessful in the 3D case and for viscous flows. Therefore, developing efficient and robust aerodynamic grid deformation techniques has been a research topic since the eighties. At that time, Batina first proposed to take advantage of a structural analogy in order to deform aerodynamic grids. The idea was to build a discrete spring network, located on the edges of the 3-dimensional aerodynamic grid, and to solve the static elastic equilibrium of the system, boundary displacements being prescribed.

Other structural analogy techniques are also used. The *elsA* code has several such techniques [12]. In particular, it may solve the grid deformation problem using a linear elastic material analogy, combined with finite element modeling. Due to the basic properties of the structural stiffness matrices, efficient linear system resolution techniques may be implemented, such as the pre-conditioned conjugated gradient. The so-called volume-spline (or infinite-plate) technique is also a structural deformation inspired technique that solves the problem of an infinite two-dimensional elastic plate, loaded at discrete nodes. This model leads to the resolution of a bi-Laplacian PDE, independent from any grid topology information. Related methods are based on the resolution of similarly built linear systems, using radial basis functions (RBF).



Integral methods are also popular, some of them corresponding to a linear aerodynamic flow problem analogy. Purely analytical techniques are also available, such as the transfinite interpolation technique, or fitting techniques, however, based on expensive distance computations. In order to reduce the computational cost, multilevel techniques combining analytical and structural analogy methods may also be implemented.

When the modal approach is used for the structural modeling, generalized quantities are naturally used for the transfer of loads (generalized forces $\Phi^t F_A(t)$) and displacements (generalized coordinates q : $x = \Phi q$). In this case, the conservation of virtual work is strictly ensured, but equivalent mode shapes Φ must be defined also on the aerodynamic surface grid, which implies the use, in a pre-processing step, of interpolation or fitting techniques. When using finite element structural modeling, things may be more complicated. Displacement transfers need to be performed via interpolation or fitting techniques at each coupling step. Load transfer must satisfy the conservation of mechanical work at the fluid/structure interface. Several approaches are available to do this, such as the so-called "nearest neighbor" strategy and methods based on the virtual work conservation principle.

Aeroelastic dynamic stability prediction

Several approaches may be used in order to analyze the dynamic stability of a fluid-structure coupled system. The first one corresponds to a weak coupling approach, in the frequency domain, where the fluid solver is implemented to compute the aerodynamic response to an unsteady structural forced motion. Classically, harmonic vibration motions following modal deformations of the structure $(q_n(t) = q_n^* e^{pt})$ are prescribed and give access to the values of the generalized harmonic forces for a single value of the excitation frequency. So-called "pulse" excitation techniques may alternatively be used to obtain harmonic GAFs within a frequency range, with a single aerodynamic simulation. In the weak coupling approach, aerodynamic forces may be considered as additional aerodynamic stiffness and additional positive or negative damping terms of the mechanical system:

$$(p^2\mu + p(\beta - B) + (\gamma - A))q^* = 0$$

$$(A + j\omega B) = \frac{\Im(\Phi^t F_A)}{\Im(q^*)}$$

Complex eigenvalues of the coupled system expressed in the frequency domain are extracted and give access to the frequencies and dampings. A negative damping value leads to an unstable behavior of the system.

In a second approach, the resolution is carried out in the time domain, following a strong coupling method. In this case, a staggered scheme is used to solve the fluid and structure equations, in order to benefit from the large discrepancy between both domain typical time steps. An additional mechanical convergence loop is necessary to simultaneously achieve the convergence of deformation, fluid and structural coupled equations. In this process, load and displacement transfers are performed up to convergence at each coupling step. The time histories of coordinates or loads are analyzed in the frequency domain, to obtain the eigen-frequencies and damping values of the coupled fluid-structure system.

Box 4 - Turbomachinery aeroelasticity

Turbomachinery aeroelasticity involves additional complexity at several levels [16]. First of all, rotation must be taken into account, which leads to extra centrifugal and gyroscopic force terms for the numerical formulation. The second point is the cyclic periodic design of such machines. Concerning the structural dynamics, these effects induce an increase in the eigen frequencies of the system with rotation speed (Campbell diagram) and the occurrence of complex mode shapes, combined with inter-blade dephasing. On the other hand, the aerodynamic flow exhibits complicated features, such as shocks, boundary layers and separation, as well as tip and hub corner vortices. Instability may be produced by unsteady aerodynamic features, inducing classical flutter or stall, or choke flutter. Unsteady phenomena, such as a rotating stall may propagate over the complete wheel.

Moreover, turbomachines are made of a set of several wheels, rotating at different speeds and all interacting together. Even in the rigid blade case, the global aerodynamics must be considered as unsteady, due to the periodic effect of the blade passing of adjacent wheels. As a consequence, the numerical cost of unsteady aeroelastic simulations may be even more important than for external flows around aircraft. This is the reason why reduction techniques are necessary, in order to keep the CPU cost acceptable for every-day industrial use. As an alternative to URANS modelling, linearized formulations have been proposed to reduce computational costs. Harmonic Balance formulations are also under development and already show promising results for single frequency problems, and they may be of great interest in the case of multi-stage unsteady applications. Geometric reduction is possible for the weak coupling approach, in the framework of the dynamic aeroelastic stability analyses. In this case, harmonic forced motion simulations can be conducted on a single sector of the machine, using a specific space-time periodic boundary condition at upper and lower boundaries of the channel. Due to the vicinity of the blades, the inter-blade phase angle of the modal vibration is a main parameter in the study of the dynamic aeroelastic stability of blade rows. In-phase or out-of-phase blade vibrations lead to different aerodynamic damping values. Moreover, the aeroelastic behaviour is highly dependent on the operating conditions of the machine, rotation speed and pressure ratio.

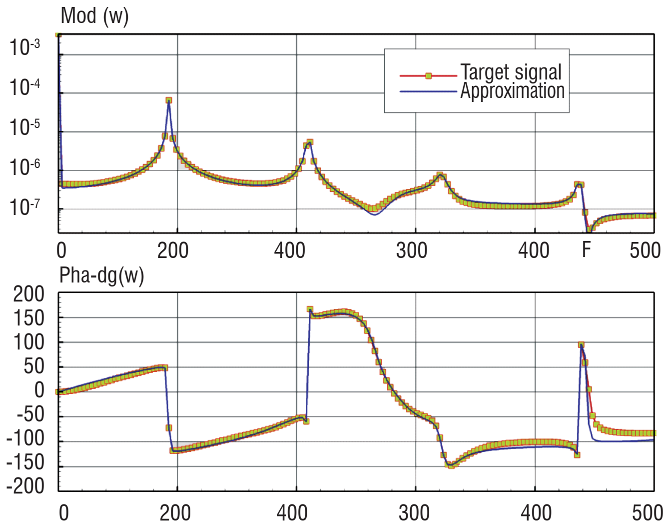


Figure 12 - Identification technique applied to extract frequencies and damping information from unsteady signals.

elsA/Ael simulations

Ael subsystem overview

The fluid-structure problem can be formulated as a coupled field problem, where the solutions are coupled only at the boundary interfaces between the fluid and the structure. It is then possible to run separate solvers for the flow computation and the structure computation, and to reach a coupled solution by exchanging information at the common fluid-structure boundaries. However, in most aeroelastic problems, the structure may be assumed to be linear. In this case, it is much easier and more efficient to extract the structural information from a full finite element model in a pre-processing step and to solve the mechanical system directly in the aerodynamic code. This is the strategy that has been adopted to extend the aerodynamic *elsA* solver for fluid-structure coupling.

A general framework has been developed in the optional "Ael" subsystem of *elsA* over the last few years, in order to extend *elsA* to different types of static or unsteady aeroelastic simulations (figure 13).

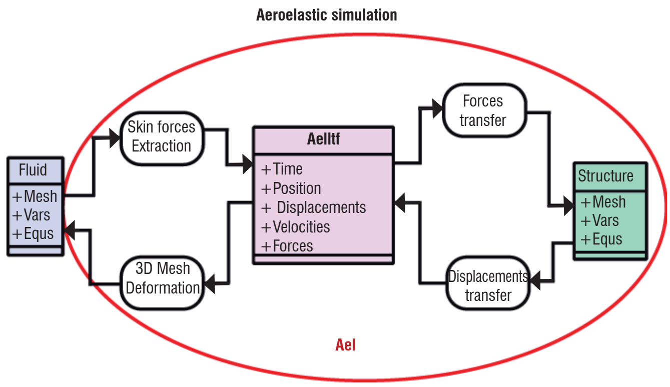


Figure 13 - *elsA*/Ael aeroelastic subsystem.

The purpose of these simulations is the prediction of the in-flight static or dynamic behavior of flexible aerodynamic structures and their aeroelastic stability. This "Ael" subsystem gives access in a unified formulation to different types of aeroelastic simulations, while minimizing the modifications of the flow solver. The available simulation types range from non-linear and linearized harmonic forced motion computation, to static coupling and consistent dynamic coupling simulations in the time-domain, with different levels of structural modeling ("reduced flexibility matrix" for static coupling, modal approach, or full finite element structural model) .

Aerodynamic modeling available within Ael

Euler and RANS equations are available for static aeroelasticity, using the *elsA*/Ael module. In the dynamic case, several specific formulations are available for aeroelastic modeling. The most general is the

URANS model, but alternate models may be used for periodic unsteady simulations:

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} = 0$$

The first one is the linearized URANS formulation. In this case, the flow field is considered to be the sum of a steady part and a small time harmonic perturbation field:

$$W = W_S + \delta W^* e^{j\omega t}$$

These assumptions lead to a complex formulation, taking into account structural motion via mesh deformation, specific boundary conditions and ALE. A linear system is solved using the LU-SSOR classical approach, giving access to the first harmonic of the perturbation field and thus to unsteady loads due to the vibration motion of the structure.

Another alternate approach is the harmonic balance method [11]. The Time Spectral Method (TSM), available in elsA as from version 3.4, is dedicated to the simulation of time-periodic flows. Basically, the flow is assumed to be given by a Fourier series expansion of 2N+1 terms:

$$W = \sum_{n=-N}^{n=N} \omega_n e^{jn\omega t} \quad F = \sum_{n=-N}^{n=N} f_n e^{jn\omega t}$$

The TSM casts a time-periodic Unsteady RANS problem to the simultaneous resolution of 2N+1 coupled steady RANS problems, corresponding to a uniform sampling of the time-period. This odd number stands for stability and allows the capturing, at the most, of the Nth harmonic of the fundamental frequency {Shannon 1949}. The coupling is handled by a pseudo-spectral time derivative only valid for periodic flows. This operator appears as a source term of the steady Navier-Stokes equations. The convergence of steady problems is better mastered than the unsteady transient of U-RANS simulations, which enables a better efficiency for the resolution of time-periodic problems.

Applications

Prediction of control surface efficiency

An accurate prediction of the effectiveness of control surfaces is needed, both for aircraft manoeuvre studies and for load alleviation control. The prediction of aileron efficiency, taking into account aeroelastic effects, is investigated here for the case of the HiReTT wing model, experimentally investigated in ETW. Rigid wing and flexible wing simulations have been achieved for transonic flow conditions, in the case of a 3° down deflection of the aileron [13].

The rigid wing computation clearly overestimates the global coefficient variations (see Table 1), due to a deficient shock location (Figure 14c). A good prediction of the lift and drag fluctuations is provided, on the other hand, by the static coupling simulation, which takes into account the small change of wing deformation induced by the aileron deflection, see figure 14d.

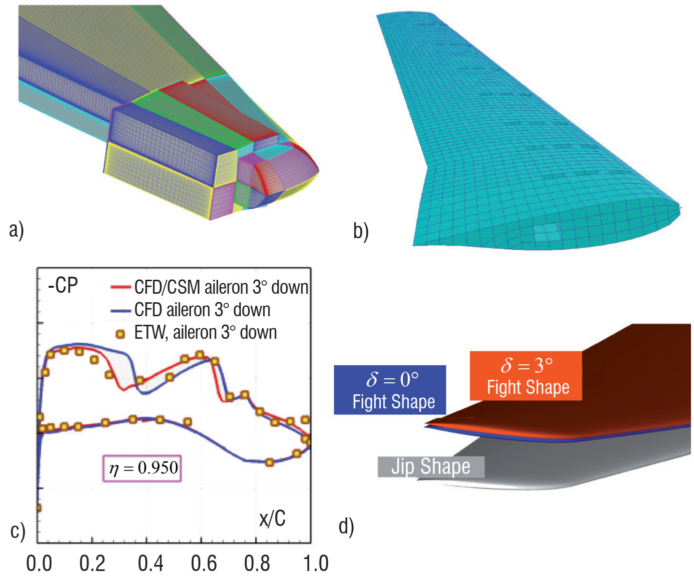


Figure 14 - a) Aerodynamic Chimera grid, b) structural finite element model, c) pressure distributions, d) wing deformations.

	C_L δ=0°	C_L δ=3°	$dC_L/d\delta$	C_D δ=0°	C_D δ=3°	$dC_D/d\delta$
Rigid wing	0.4750	0.4908	5.26 E-03	0.0235	0.0245	3.3 E-04
Flexible wing		0.4835	2.83 E-03		0.0243	2.7 E-04
Experiment	*0.4869*	*0.4956*	*2.90 E-03*	*0.0250*	*0.0257*	*2.3 E-04*

Table 1 - Lift and drag variations induced by the aileron deflection.

Aeroelastic stability of a centrifugal compressor

The Aeroelasticity and Structural Dynamics Department of Onera (DADS) has been involved in the MACAO project of the French "pole de compétitivité" Aerospace Valley, in cooperation with Airbus, the SAFRAN group and local aerospace companies. The aim of the actions carried out was the study of the dynamic aeroelastic stability of a Turbomeca centrifugal compressor model, using non linear aerodynamic modeling. The wheel is about 30 cm in diameter and consists of 11 sectors, each of these including a main and a secondary blade. The rotation speed of the selected operating point is roughly 50,000 rpm.

The *elsA* solver has been implemented in RANS and URANS mode, using a Smith kl turbulence model and an upwind Roe space discretization scheme. Weak coupling aeroelastic simulations have been performed to study the dynamic stability of blade-only vibration modes and blade-and-hub vibration modes.

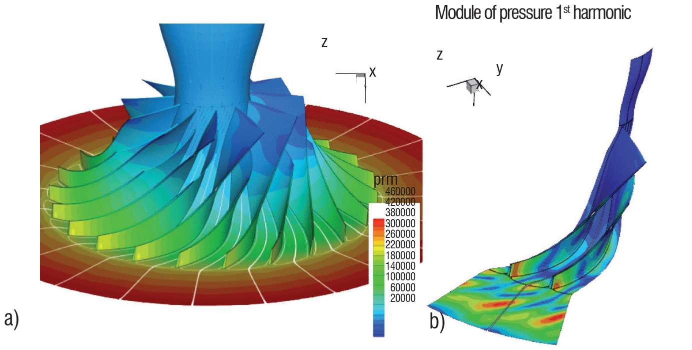


Figure 15 - (a) Centrifugal compressor mean pressure - (b) aerodynamic response to zero-dephasing vibration (pressure module 1st harmonic).

This second action leads to additional issues concerning the aerodynamic grid deformation. The structural analogy mesh deformation tool has been successfully implemented in this case and has helped to run a set of aeroelastic simulations at the targeted operating point, giving access to the aeroelastic dampings of the selected vibration modes, for various inter-blade phase angles. The intention of further study is to perform similar analyses in the case of the modeling of the blade tip clearance at the shroud. In order to reduce the CPU cost of these computations, TSM, as well as wall law approach simulations, are under investigation for this model.

Counter-rotating open rotor (CROR) aeroelasticity

The aeroelastic stability of a CROR has been studied within the framework of the European DREAM project. The model consists of a front 12 blade propeller and a rear 10 blade propeller. Two operating conditions have been selected: a take-off case at M=0.25 and a cruise case at M= 0.78. These models have been scaled to a 1/5.1 ratio, for the purpose of conducting wind tunnel experiments. The analysis of the aeroelastic stability of the front wheel has been performed, using the weak coupling approach in the frequency domain. Therefore, generalized aerodynamic forces must be computed for time harmonic motions following vibration mode shapes at natural frequencies for each operating point. The effect of the second wheel has been taken into account in this study, using a mixing plane boundary condition at the interface between front and rear wheels. This is of course a strong simplification: unsteady perturbations generated by the passing of the rear blades are neglected. However, this assumption allows the reduction of the numerical model to a single passage for each propeller, combined with the use of the space-time periodic boundary condition. To do this, the space-time periodic boundary condition available in *elsA* has been updated, in order to take into account the use of an absolute velocity formulation in a rotating frame, which is of common usage for propeller simulations.

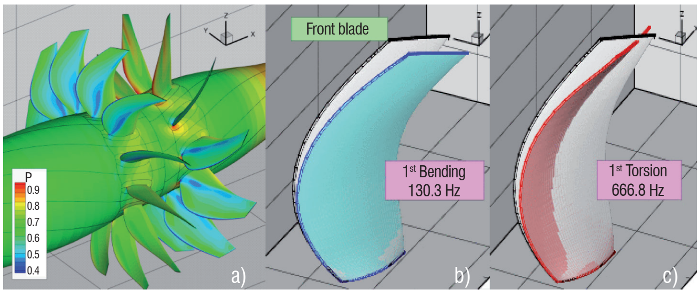


Figure 16 - (a) CROR static pressure at TO (b) 1st bending mode - (c) 1st torsion mode.

Aerodynamic damping values have been extracted for both operating points and for the three first modes of the front blade, showing a stable behavior of the propeller. Additional studies have also been conducted, to analyze the forced response of the rear propeller due to wake passing of the first propeller blades, and the static deformation of the propeller due to aerodynamic loads.

Dynamic coupling in the time-domain

The possibility of predicting flutter at transonic speeds, through direct Navier-Stokes coupled simulations in the time domain, has been validated in the case of a wing/body model experimentally studied in the Onera Modane S2 wind tunnel [14]. Dynamic coupled simulations have been run for different stagnation pressures, using the 6 first structural modes. As shown in figure 17, a good correlation with the experimental results is observed in the frequencies and the damping coefficients, which are computed from the dynamic responses.

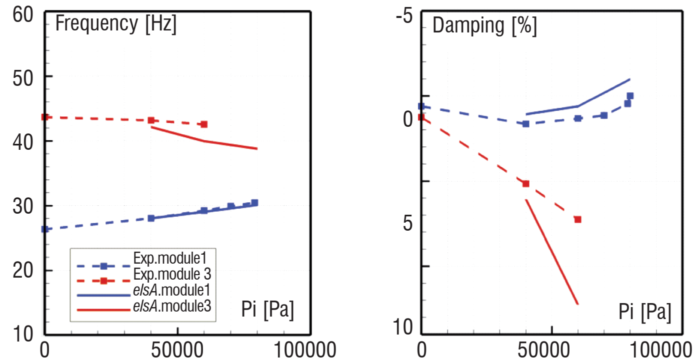


Figure 17 - Flutter diagram for $M_\infty=0.84$, $\alpha=0^\circ$.

The appearance of limit-cycle oscillations (LCO) is another important aeroelastic phenomenon, which may only occur when non-linear effects are present in the fluid-structure system. Even if the LCO behavior does not directly lead to a catastrophic failure, such as for example the flutter phenomenon, it may considerably reduce the fatigue life of the structure.

The capability to predict the appearance of aerodynamically induced LCO has been investigated in the case of the NLR7301 airfoil equipped with a 2-dof structural system allowing heave and pitch motions [15]. The predicted time evolution of the pitch angle (figure 18) exhibits the characteristic limited amplitude phenomenon experimentally observed by DLR in the case of this airfoil.

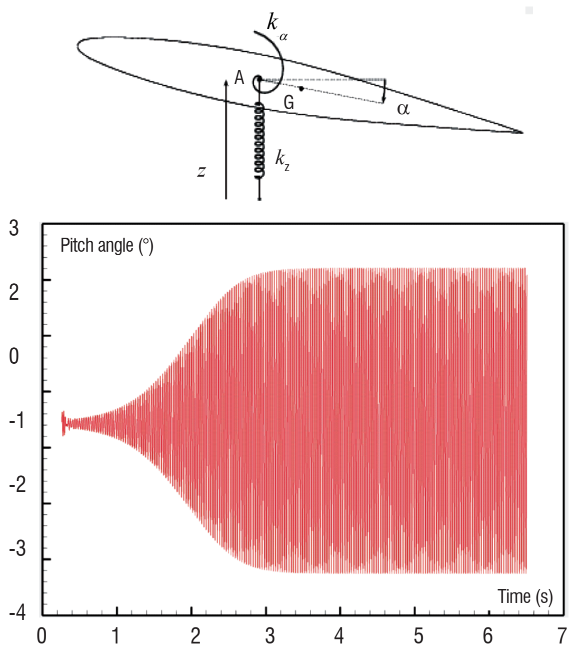


Figure 18 - Experimental set-up studied at DLR, and numerical simulation of the LCO phenomenon.

Gust response in time-domain

The design of aircraft requires the nonlinear response to a gust field to be taken into account. The dynamic loads that are induced by a gust may moreover be significantly increased by the highly flexible structures of modern aircraft. The prediction of the gust loads on the structure of an aircraft thus becomes increasingly important during the development and certification phases of a project. *elsA* has been recently extended, in order to take into account an analytical perturbation velocity field, which may be used to model a gust. The developments allow direct aerodynamic and aeroelastic gust response simulations to be performed in the time-domain.

Figure 19 compares the dynamic gust response predicted at transonic flight conditions, without/with taking into account the wing flexibility. In this case, the frequency of the "1-cosine" gust is very close to the frequency of the first torsion mode of the wing. This explains the much higher unsteady loads observed in the dynamic coupled simulation.

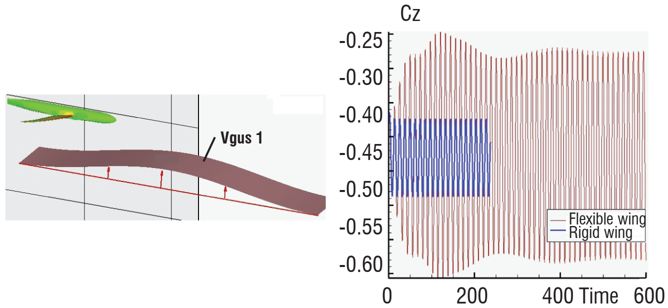


Figure 19 - Comparison between the aerodynamic and aeroelastic gust response, in the case of a wing/body configuration.

Underlying software architecture

The multi-physics simulations presented in the previous sections are performed on complex software systems. The following section focuses on this software and specifically on the code interfaces (that is, the set of functions used to interact with other software).

Nowadays, a complex simulation is performed through the use of several codes running on distributed heterogeneous computers, rather than running a single monolithic code. This situation leads to a large set of interfaces. In this section, it is shown that a common interface definition helps to reduce the entire software life cycle (development, integration, use and maintenance effort).

The need for a common interface

The codes involved in a multi-physics simulation can be one of four types: legacy code (i.e., code no longer developed but still used and maintained), COTS (Commercial Off the Shelf), Open Source and in-house codes. All these codes have communication capabilities to a greater or lesser extent, which is a key issue for multi-physics simulations.

Each code provides its own interface, which is a means to interact with the user or with another application and is usually based on the use of files and/or functions. An interface formally defines how to control, send and receive data.

On the one hand, some codes have their own opaque and proprietary file format. In this case, the data exchange is only possible using specific proprietary tools. There are even monolithic codes with no interaction capabilities, which is indeed an issue for multi-physics simulation purposes. On the other hand, some codes supply a public and documented format, so that any other tool can be used to read/write the file or to connect to the network. Such programs generally provide a large set of functions, available through an Application Programming Interface (API).

Since each program defines its own interface, a straightforward approach leads to a connection graph, such as that in figure 20. Each connected code requires a dedicated interface to each other peer-code. In the worst case, such a graph is complete and should be avoided. Figure 21 shows a connection graph with a common interface, the so-called software bus, with a minimal connection graph.

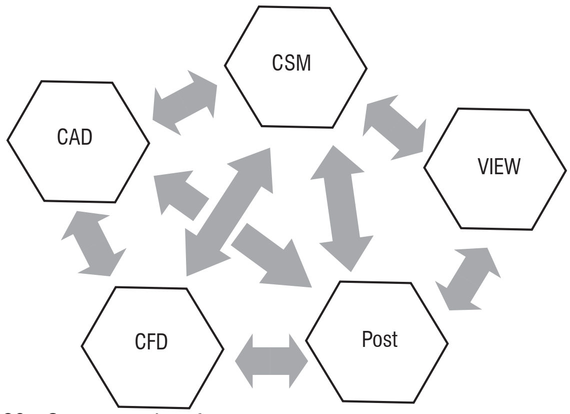


Figure 20 - One-to-one interfaces.

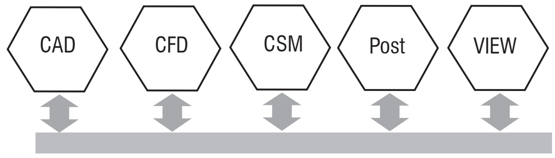


Figure 21 – Software bus.

NSCOPE (Numerical Simulation Components in an Open Python Environment), adopted by some Onera software (e.g., *elsA*, see previous section) provides a public and non-proprietary interface. The *elsA* solver, like most Onera tools, is not Open Source, but can easily be connected to Open Source software: an interface can be open while its implementation is not.

NSCOPE specifies the three elements of an interface: the control and communication are performed by Python, the data is handled by both CGNS [19] for the logical representation and HDF5 (see the hdf group website) for the physical representation. These CGNS/Python and CGNS/HDF5 technologies are Open Source, public and allow any application to connect to NSCOPE-compliant tools. NSCOPE is neither a library, nor a proprietary specification, it is a set of technologies recommended to increase interoperability.

Integration strategies

The integration is the assembly of several codes in order to achieve a multi-physics simulation. This could be performed in a specific separate controller, or directly in the application. As mentioned in the Vulcain 2 case described above, each application is a specific case and requires a specific algorithm, which impacts control, data exchange and even data representation.

Three examples are now given that show how the Open Python approach helps to achieve an easy, low cost and maintainable integration of a multi-physics application. Each example focuses on a single element of the interface, namely the data model, communication and control.

Example: data transformation

The use of CGNS/Python as a common data model and data structure was demonstrated within the framework of the CHANCE project with Eurocopter [17]: fluid/structure code-coupling was developed for a helicopter rotor application.

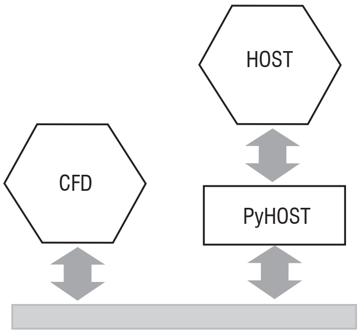


Figure 22 - The pyHOST adapter.

The coupling involved flight mechanics, with the structure via the HOST software (in-house Fortran Eurocopter code, with few interoperability features). First, data had to be transformed between HOST and *elsA*. An adapter was developed to provide HOST with an NSCOPE interface (CGNS/Python). The main difficulty was to couple *elsA*, which computes the full rotor in parallel whereas HOST only manages a single blade model. Moreover these two codes operate on different time scales.

The pyHOST interface (see figure 22) was restricted to the needs of the simulation, but the definition and implementation follow the NSCOPE recommendations. No change was necessary in *elsA*, because it was already NSCOPE compliant.

Example: communication

The communication layer sends and receives data from/to the codes. Some simulation frameworks provide their own communication layer and the coupling application must use it.

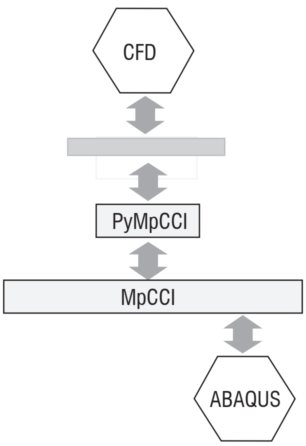


Figure 23 - Communication vs open communication layer.

In a specific project in cooperation with SNECMA, the use of the MpCCI library was mandatory. Thus, an NSCOPE compliant interface was developed on top of MpCCI API. This interface provides the user with a full CGNS/Python interface and a communication layer built using a Python network library.

Example: process control

The Python programming language was used to develop the steering and control application in a PhD thesis [18] conducted in connection with the Eurocopter SHANEL project. This project included a fluid/structure code-coupling application for a helicopter rotor. One of the goals was to add an external CSM/CSD commercial software (MSC.MARC) into a workflow already using *elsA* and HOST.

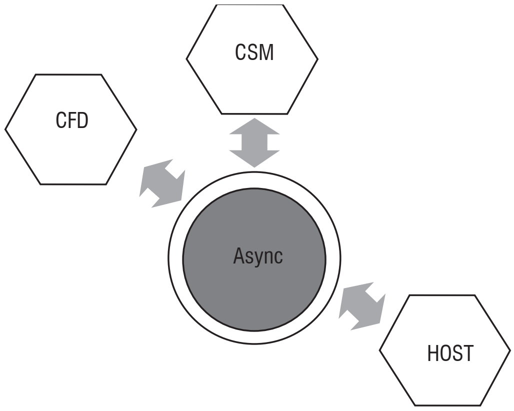


Figure 24 - Asynchronous Python server.

The simulation had to run on a heterogeneous network, due to user MSC.MARC license issues. A classical workstation was involved, as well as a supercomputer handling batch queues. The application had to synchronize all codes across the network, using an asynchronous server, which was in charge of storing data as it came from the different codes. With the use of the XMLRPC library of Python, this server had no more than 10 lines. The actual PhD could then focus on the global algorithm of the simulation, not the software middleware. SHANEL also lead to the specification of a CGNS fluid/structure code-coupling interface.

Box 6 - Python

Python is a programming language, born in the Netherlands in 1994. Its father, Guido van Rossum, has designed a nice and extensible language for network testing purposes. The language rapidly became very used in two communities having good programming skills: the Internet and numerical simulation. The language is object oriented, interpreted and can be extended by providing a Python interface to existing C or Fortran libraries. The syntax is clear and easy to learn, the language types are usual scalar types such as integers, real numbers and strings and also container types such as lists or dictionaries.

```
A=4
B=2.34*4
C=[ A, B, ['This', 'is', 'an heterogeneous list'], [0.03, 0.2e-12]]
```

Its object oriented features makes it possible (1) to create one's own application types and (2) to support good software structure for one's codes.

Python has a large number of embedded libraries, the standard installation provides the users with all of the services related to the operating system, the network, the graphical user interfaces, etc. These libraries are called modules, and the modules can be packaged, in order to be shipped to an application or another user, and then imported before use.

```
import math
def circ(R):
       C=2*math.PI*R
       return R
print R(5.28)
```

In the domain of numerical simulation, the best library comes from the so-called 'SciPy' package. This includes in particular the numpy library. This very efficient library offers many classes, functions and extensions to manipulate arrays. An array can be mapped to a C or Fortran array, so that Python/numpy can be used as a nice interface for numerical simulation functions in C, C++ or Fortran. This library is clever enough to allow almost all kinds of mapping to numerical functions and the cells in your actual Fortran array can be read/written even.

```
import numpy
a=numpy.array( [[1,2,3],[4,5,6]], dtype=numpy.float64, order='Fortran')
a[0][1]=7
```

A module creation can be a simple file in the Python language declaring and implementing constants and functions. It can be more complex, if it is extended with a Fortran function. In this case, it is necessary to draw a box around the Fortran or C code. This box is a C code and uses the Python C/API functions. It translates the Fortran data to and from Python. Extending Python in such a way is something common: there are many code generators, such as SWIG or F2PY, that take a Fortran or C code and automatically generate the Python interface.

Python is said to be a 'steering language'. It is used to encapsulate existing codes and gather them into a single large program, and it ensures network communication between existing codes, for code-coupling for example, or allows the user to quickly write a nice graphical interface for his application. Moreover, the very good portability of Python makes it run on all known platforms and, as a matter of fact, there is now almost no standard platform without Python installed as default.

Python is a very convenient interface and may be chosen as the top level language for many application programs.

Conclusions

Many aeronautical problems are basically multiphysics problems, in which the behavior of a system depends on the interaction between several distinct disciplines. In the past, these coupling effects were often ignored, or crudely taken into account, due to lack of computational capabilities. Today, with the increase in computer power, multiphysics numerical simulations are being used increasingly more in order to obtain accurate solutions, needed for the optimization and performance improvement of aeronautical components.

This paper has presented two different approaches developed at Onera to solve such multi-physics problems via a partitioned strategy. In the first one, separate and independent simulation tools are coupled via a coupling library. The main advantage of this approach is to benefit, for each software, from the experience developed by a large number of specialists over many years. In the second approach, mono-disciplinary software is extended to multi-physics modeling by adding new simplified modules for other disciplines and by the implementation of specific coupling algorithms. The advantage of this approach is to provide a simple and efficient computational environment for the simulation of steady and unsteady coupled processes.

However, in both cases, coupling may be challenging, due to great disparities between the fluid and the solid physical models. In particular for dynamic problems, the main numerical difficulty arises from the significant discrepancy between time scales of both media. The studies presented in this paper suggest that partitioned strategies increase the accuracy and details of numerical results, due to the complementary information from the fluid and the solid domains only if specific well designed mathematical models are used.

In the last part of this paper, an Open System approach has been presented. It is based on a powerful open source assembly of public interfaces developed at Onera for software coupling strategies.

It must be noted that work is currently underway to develop new coupling libraries able to exchange the information between solvers and to automate the search, interpolation and communication processes over a large number of processors in a distributed environment. The existence of such a tool will allow us to concentrate on numerical and mathematical models in fluid-structure interactions. In particular, this approach will also allow aeroelastic modeling to be improved, by taking into account both fluid and structural non-linearities ■

Acknowledgements

The authors would like to thank the French governmental DPAC and DGA agencies and the European Commission for funding the research studies presented. Several researchers must be acknowledged for their work and their contributions to this paper: C. Liauzun for the Gust response simulations and F. Sicot for the Time Spectral Method topic. Finally, we wish to thank the Z-set software development community in general, and more particularly A. Roos and F. Feyel, for their help in setting up these simulations.

References

[1] O.C. ZIENKIEWICZ - *Coupled Problems and their Numerical Solution*. Numerical Methods in Coupled Systems. Edited by R.W. Lewis, P. Bettess, and E. Hinton, John Wiley and Sons ltd, 1984.
[2] M. ERRERA, B. BAQUE and M. REBAY - *A Numerical and Experimental Study of Transient Conjugate Heat Transfer in a Flat Plate*. International Symposium on Convective Heat & Mass Transfer in Sustainable Energy, April 26- May 1, 2009, Hammamet, Tunisia.
[3] A. REFLOCH, B. COURBET, A. MURRONE, P. VILLEDIEU, C. LAURENT, P. GILBANK, J. TROYES, L. TESSÉ, G. CHAINERAY, J.B. DARGAUD, E. QUÉMERAIS and F. VUILLOT - *CEDRE Software.* Aerospace Lab Issue 2 March 2011.
[4] D. SCHERRER, F. CHEDEVERGNE, P. GRENARD, J. TROYES, A. MURRONE, E. MONTREUIL, F. VUILLOT, N. LUPOGLAZOFF, M. HUET, B. SAINTE-ROSE, P. THORIGNY, N. BERTIER, J.M. LAMET, T. LE PICHON, E. RADENAC, A. NICOLE, L. MATUSZEWSKI and M. ERRERA - *Recent CEDRE Applications*. Aerospace Lab Issue 2 March 2011.
[5] K. WOLF - *A General Coupling Library for Multidisciplinary Simulation*. Workshop on Scalable Software Multiscale Coupling and Computational Earth Science. SSS 2001 December 3-5, The University of Tokyo, Japan.
[6] S. PIPERNO, C. FARHAT, and B. LARROUTUROU - *Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems Part i: Model Problem, Theory and Two-Dimensional Application.* Computer Methods in Applied Mechanics and Engineering, 124:79-112, 1995.
[7] J. GIORDANO & al. *Shock Wave Impacts on Deforming Panel, an Application Of Fluid-Structure Interaction*. Shock Waves, 14:103-110, 2005.
[8] A. ROOS and J.L. CHABOCHE - *Vulcain II Cooling Tubes. Final report*. Tech. rep. RT 2/08500, Onera 2005.
[9] J. BESSON, G. CAILLETAUD, J.L. CHABOCHE, S. FOREST and M. BLÉTRY – Non-linear mechanics of materials, Springer 2010.
[10] F.X. ROUX and J.D. GARAUD – *Domain Decomposition methodology with Robin interface matching conditions for solving strongly coupled fluid-structure problems.* International Journal for Multiscale Computational Engineering, 2009.
[11] G.DUFOUR, F. SICOT, G. PUIGT, and C. LIAUZUN - *Contrasting the Harmonic Balance and Linearized Methods for Oscillating-Flap Simulations*. AIAA Journal No 4 Volume 48, April 2010.
[12] A. DUGEAI – *Aeroelastic Developments in the elsA Code and Unsteady RANS Applications*. International Forum on Aeroelasticity and Structural Dynamics, 28 June-1 July 2005, Munich, Germany.
[13] P. GIRODROUX-LAVIGNE – *Fluid-Structure Coupling Using Chimera Grids*. International Forum on Aeroelasticity and Structural Dynamics, 21-25 June 2009, Seattle, USA.
[14] P. GIRODROUX-LAVIGNE - *Recent Navier-Stokes Aeroelastic Simulations Using the elsA Code for Aircraft Applications*. International Forum on Aeroelasticity and Structural Dynamics, 18-20 June 2007, Stockholm, Sweden.
[15] P. GIRODROUX-LAVIGNE - *Prediction of Limit Cycle Oscillations Induced by Aerodynamic non-Linearities*. NATO RTO Symposium AVT-152 on Limit-Cycle Oscillations and other Amplitude-Limited. Self-Excited Vibrations, 5-8 May 2008, Loen, Norway.
[16] A. DUGEAI - *Turbomachinary Aeroelastic Developments and Validations Using Onera elsA Solver*. International Forum on Aeroelasticity and Structural Dynamics, 18-20 June 2007, Stockholm, Sweden.
[17] M. COSTES, M. POINOT, and B. CANTALOUBE - *CHANCE Program: Strong Coupling Between elsA and HOST*. RT 1/07782 DAAP SPAe, Onera, 2005.
[18] B. ORTUN - *CSM/CFD Coupling for the Dynamic Analysis of Helicopter Rotors.* PhD Thesis dec.2008, Onera.
[19] M. POINOT - *SHANEL Project Phase 1 - Synthesis Report RTS1 - Task A4b : Pre/Post Processing and Simulation Environment CSM Extensions of the CGNS Standard*. RT 8/11826 DAAP DGA/DET/CEP, Onera, 2008.

Acronyms

CFD (Computational Fluid Dynamics)
CSD (Computational Structural Dynamics)
RANS (Reynolds Averaged Navier-Stokes)
URANS (Unsteady Reynolds Averaged Navier-Stokes)
HOST (Helicopter Overall Simulation Tool)
RBF (Radial Basis Functions)
LUSSOR (Lower Upper Symmetric Successive Over Relation method)
ALE (Arbitrary Lagrangian Eulerian)
FSI (Fluid-Structure Interaction)
CHT (Conjugate Heat Transfer)
COTS (Commercial Off The Shelf)
MpCCI (Mesh-based parallel Code Coupling Interface)
SPRM (Solid Propellant Rockets Motors)
IUSTI (Institut Universitaire des Systèmes Thermiques Industriels)
TSM (Time Spectral Method)
ETW (European Transonic Windtunnel)
CROR (Counter-Rotating Open Rotor)
LCO (Limit-Cycle Oscillations)
NSCOPE (Numerical Simulation Components in
an Open Python Environment)
CGNS (CFD General Notation System)
CPU (Central Processing Unit)

Web sites

http://www.python.org
http://www.hdfgroup.org
http://www.cgns.org
http://numpy.scipy.org
http://www.zset-software.com
http://elsa.Onera.fr
http://cedre.Onera.fr

AUTHORS

Marc Errera, graduated from the "Ecole Centrale de Paris" in 1980 and obtained a Research Habilitation Thesis (Thèse d'Etat) from the University Pierre & Marie Curie (Paris 6) in 1990, in the field of in-cylinder flow in reciprocating engines. He is currently in charge of multidisciplinary projects.

Alain Dugeai, an Engineer of "l'Ecole Centrale de Paris", has been working at Onera since 1988, in the "Aeroelastics and Structural Dynamics" department. He is currently in charge of numerical aerolasticity developments for turbomachines.

Philippe Girodroux-Lavigne, an Engineer of "l'Ecole Centrale de Lyon", has been working at Onera since 1979, first in the "CFD and Aeroacoustics" department, then in the "Aeroelastics and Structural Dynamics" department, where he is involved in aircraft numerical aeroelasticity studies.

Jean-Didier Garaud received his PhD in 2008 from the University Pierre & Marie Curie (Paris 6) and Onera, on the topic of numerical analysis and computational mechanics. He is now an engineer in the Department of Metallic Materials and Structure, in charge of Z-set's multiphysics capabilities.

Marc Poinot is a software engineer at Onera He obtained a Masters Degree in Computer Science from Paris XI Orsay, France. He is working in the CFD team, where he is in charge of interoperability topics for code coupling and numerical simulation platforms. He is member of the CGNS steering committee and he initiated and actively participated in the migration of CGNS to HDF5 and Python.

Stéphane Cerqueira followed a double degree - French Engineer (Supmeca - Paris)/Master in Aerospace (Laval University/Ecole Polytechnique de Montréal/Mc Gill University - Canada). His PhD is focused on Fluid-Structure Interaction inside Ariane V's Solid Propellant Rocket Motors and was supported by CNES/Onera.

Gilles Chaineray received his PhD in energetics and heat transfer, in 1995. The PhD research activities were focused on study of rotating flow and heat transfer, and the development of Anisotropic Stress Model. He is now in charge of the development of thermal and mechanical coupling of the CEDRE code with other codes such as Z-set, ABAQUS, MARC, Code_Saturne.