SoNICS: a New Generation CFD Software for Satisfying Industrial Users Needs

Denis Gueyffier[1], Sylvie Plot[2], Matthieu Soismier[1]
[1]DAAA/CLEF, ONERA, The French Aerospace Lab,
[2]DPA, ONERA, The French Aerospace Lab,
FRANCE

denis.gueyffier@onera.fr, sylvie.plot@onera.fr, matthieu.soismier@onera.fr

ABSTRACT

We briefly present a novel software called SoNICS (So Naturally Innovative CFD Solver) for high performance CFD simulations. SoNICS aims at replacing in the medium term, both at ONERA and at several industrial and academic partners, the elsA aerodynamics simulation software. This software has a functional scope close to elsA with some major additional features. SoNICS also leverages innovations which rely on recent parallel computing and software engineering paradigms and on automatic code generation and differentiation. Several drivers have emerged which have led ONERA to designing an entirely new architecture for the SoNICS software. While our partner Safran prepares for the launch of SoNICS at its design offices in the next few years, Safran also makes use of elsA for its current motor programs which generate new needs for the elsA software. Development and validation of both SoNICS and elsA are steered though a close cooperation, as well as a common governance between ONERA and Safran. Both solvers are mainly developed by the same teams within a project supported by DGAC, the French Civil Aviation Authority. For the development of both solvers, we have set up an agile organization targeted at satisfying industrial users' needs. This organization relies on three pillars: i- handling user' needs instead of delivering features, ii- having the end-user involved at all steps of the development process, iii- testing in the end users' environment. We finally briefly illustrate a few recent applications enabled with elsA thanks to this new organization and the first milestones achieved with the new SoNICS software.

1 INTRODUCTION

During the past three decades, ONERA has developed aerodynamic simulation software for the industry and it has put a lot of effort in aligning its CFD offer with industrial needs. During this time period the elsA software [1] (property of ONERA and Safran) for example has grown from an academic scale to an industrial scale compressible internal and external flow solver which is now currently in use at several ASD industries partners like Safran, Airbus and MBDA where hundreds of users at design offices make use of it on a daily basis. Hundreds of simulations with elsA run each day on a dozen of supercomputers and, more recently, on individual computers for smaller design calculations. elsA is nowadays the main aerodynamics simulation software in use at Safran for RANS simulations for turbomachinery design. Industrialization has been at the heart of ONERA CFD teams' work. These teams have gained experience over the years and learned how to ship software matching end-users needs in industrial software platforms. Solid production and industrialization teams with high quality standards have been put in place in order to integrate new features, deploy the software at industrial and academic partners and bring support to the users. These teams contain product managers who gather and steer industrial needs and application teams who test software and run uses cases with excellent level of technical expertise of the industrial field. elsA has also captured over time models originating from various research fields: numerical analysis [2] [3] [4] [5], turbulence and transition [6][7][8][9][10][11], CFD-CSM [12][13], ...

However, in the past few years, because of drivers ranging from competition with other CFD software providers addressing the same market to breakthrough innovations in the global CFD offer (e.g. LBM software) or in the hardware technology offer (e.g. GPU clusters), ONERA has questioned itself and has challenged at the same time both its technologies and its means to industrialize its CFD software. This has led ONERA on the one end to propose new ways to improve communication with industrial users to better respond to their needs and on the other end to challenge its technical offer to become faster at capturing breakthrough innovations in CFD software products. The effort goes into two directions: pulling needs from industrial users in order to capture in software products the features responding to these needs, and pushing in the same products scientific and technical innovations from its most innovative technical and scientific experts or academic partners. In the following, we describe why and how we started developing a new software with breakthrough innovations and building a new organization to respond to users' needs.

Safran has decided to partner with ONERA along this strategy and to make use of elsA for its current and

near future aircraft motor designs, and then to switch its design offices in a few years to the next generation software SoNICS. Both elsA and SoNICS are property of ONERA and Safran. On the one hand, for its near ongoing motor programs, Safran needs additional features in elsA and on the other hand Safran needs to anticipate the changes, challenges and opportunities coming with SoNICS for its future programs.

ONERA thus follows two avenues simultaneously with the same development and application teams: continue development of the current elsA software and start building the future SoNICS CFD software. The main project supporting these developments is named SONICE. This collaborative project between Safran Aircraft Engines, Safran Helicopter Engines, Safran Tech and ONERA is supported by the French DGAC and is led by Safran Tech. Both companies, ONERA and Safran, bring resources and expertise into this project.

In the following, we present shortly the feature scope of elsA. We then present the drivers that led us to start building SoNICS. We then outline a few key points about its innovative architecture and the scope of new features and new capabilities of SoNICS compared with elsA.

2 FEATURE SCOPE OF THE CURRENT INDUSTRIAL ELSA SOLVER

The current software elsA has been in use to simulate internal and external flows for applications ranging from turbomachinery [14][15][16] to aircraft [17][18], helicopters [19][20][21], missile and rockets for several industrial partners of the ASD domain.
elsA offers flexibility in terms of geometry and meshes: structured, unstructured, hybrid [22], chimera/overset [23], body force [24] or Immersed Boundaries Method (IBM), see fig.1.

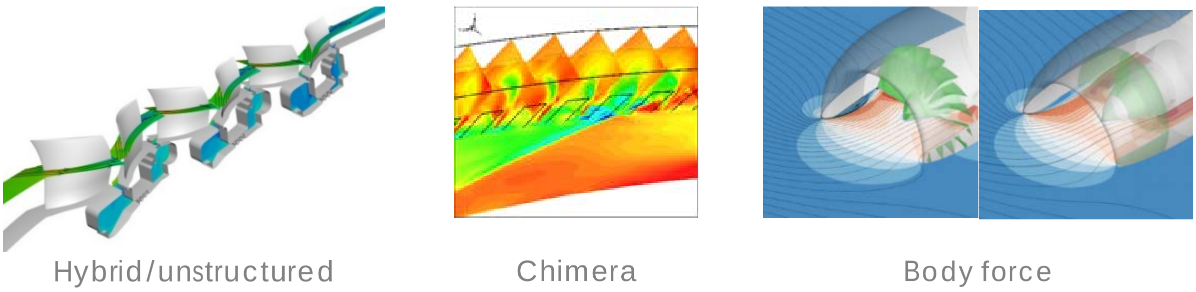

Figure 1: various meshing/geometry strategies within elsA.

elsA contains efficient and robust finite volume methods with high order schemes like *O(3)* k-exact reconstruction [4], *O(6)* Lele [5] or *O(5)* MUSCL. It offers specific boundary conditions targeted at turbomachinery or propeller applications like mixing plane, chorochronic/phase-lagged [25], time spectral methods and reduced blade count approach, see fig. 2.

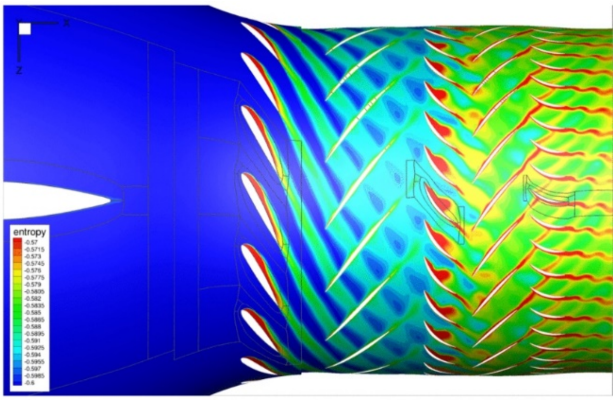

Figure 2: multi-chorochronic calculation with elsA.

elsA contains turbulence models which capture complex phenomena with various levels of precision and cost, ranging from RANS to LES modeling [26][27]. It also offers numerous laminar-turbulence transition models, see fig. 3.

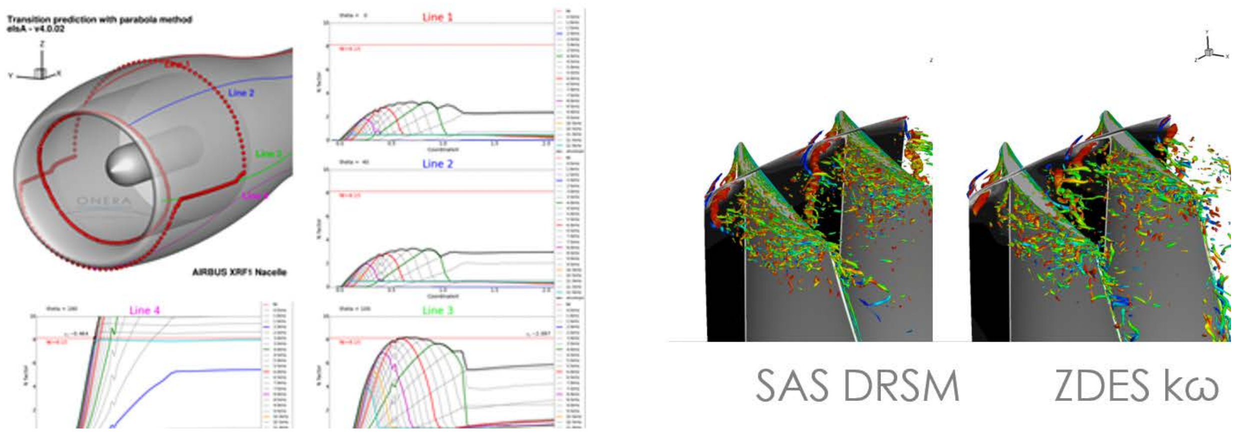

Figure 3: left: laminar/turbulent transition on nacelle using parabola model, right: comparison between SAS DRSM and ZDES model.

In addition, elsA offers advanced CFD-CSM coupling capabilities for a large spectrum of aero-elastic simulations [28] and it handles shape optimization using linearized and adjoint methods [29][30]. elsA also interoperates with external HPC solvers and external users’ scripts using python and the CGNS standard. Integration in industrial platforms also relies heavily on python and CGNS.

3 BUILDING AN INNOVATIVE CFD SOFTWARE

3.1 Motivations for building new software

For the past years, innovation has been mostly incremental within elsA although there have been noticeable improvements in numerical and physical modeling. However, there has been no major disruption within elsA regarding HPC and software architecture in the past years.

In the meantime, the ONERA CFD teams have acquired expertise in numerous fields. ONERA has been able to test very innovative software engineering and architecture concepts, or existing one which had not yet been used in industrial CFD software. The ONERA teams have developed prototypes in order to grasp all levels of parallelism on modern hardware architecture (distributed and shared memory, cache tiling [31], loop nest optimization, vectorization, GPU), linear algebra (Krylov based methods and their preconditioning, communication optimal linear algebra [32]), algorithmic differentiation [33][34][35], data flow graph programming [36][37], software product line [38] and automatic code generation and optimization [39][40].

It has been quite complex to test some of the innovations mentioned above in an industrial scale software like elsA. Nevertheless, ONERA has been able to develop from elsA an HPC prototype, leveraging cache tiling, loop nesting optimizations and vectorization and has succeeded at obtaining a x7 performance speedup (with exact same machine, and exact same calculation) compared with elsA on implicit calculations on unstructured grids. However, the elsA developers have quickly realized that it would be extremely long and costly to manually extend these optimizations to the entire functional scope of elsA which contains millions of lines of code, thousands of features and highly complex cross compatibilities between these features. Moreover, the developers involved both in the production of industrial scale production software elsA and in the development of these innovations could realize that elsA with its current architecture would be unable to sustain the changes coming with the future hardware architecture of the exascale and post-exascale era.

3.2 Building SoNICS architecture

ONERA has decided to build an entirely new CFD software in order to overcome the limitations of elsA's architecture. SoNICS new architecture relies on several innovations, as summarized on fig. 4, which relies heavily on automation:

- *Graph generation*: each specific calculation with SoNICS depends on each user needs in terms of physical and numerical models, parameters, boundary conditions and also on the target machine (CPU/GPU/Vector engines/TPU clusters). For this specific calculation, a workflow based on an operator graph is generated. The graph will not be the same if the user wants to perform e.g. an Euler simulation on a structured grid on a CPU cluster, or if the user chooses to perform instead e.g. a k-w RANS calculation on an unstructured grid on an heterogeneous CPU/GPU cluster. Such a graph is created by an automatic graph generator. This generator automatically handles cross compatibilities between features and provides flexibility to integrate new numerical and physical models in the future. It eases coupling between solvers and provides the possibility to create external operators developed by users using high level python scripts. These external operators are automatically inserted into the operator graph.
- *Code generation*: a code generator is built to transform generic code developed in a high-level language into specific code with optimized loops and numerous HPC optimization for hitting all targets of every specific user context (mesh, modeling, target hardware). With this ingredient, we propagate HPC optimizations into the entire code and separate the HPC optimizations (into a so called HPC layer) from the code handling models, hiding the complications of such optimizations to developers building numerical or physical models.
- *Automatic linearized and adjoint code generation*: we make use of algorithmic differentiation throughout the entire SoNICS software to obtain the linearized code by differentiation of the direct graph and to get the adjoint code for optimization by differentiating the same graph in reverse order.

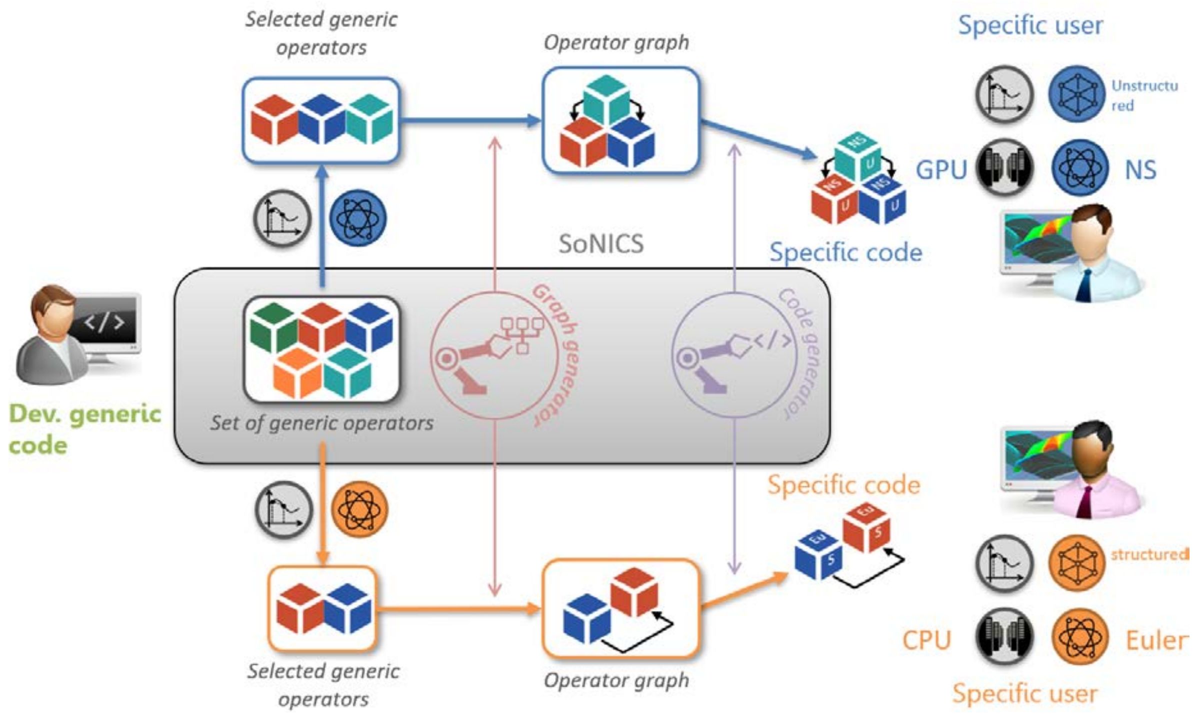

Figure 4: generating specific code from generic code.

3.3 Feature scope of new SoNICS software

SoNICS targets an enlarged application spectrum compared with elsA, and aims at new capabilities which elsA could not offer because of its older architecture.

Enlarged application spectrum:

- SoNICS considers multiple fluid species, helping us simulate configurations with higher operating temperatures and pressure,
- SoNICS offers a shorter turnaround time thanks to dynamic mesh adaptation, initializing calculation on a coarse mesh and using metric-based adaptation to refine where necessary. The introduction of node centered finite volumes will enable adaptation with anisotropic mesh cells,
- SoNICS offers advanced Krylov based linear algebra, shape optimization, flow stability, flow control based on adjoint and embedded deep learning capabilities based on backpropagation (learning optimal parameters of numerical or physical models), all of these thanks to algorithmic differentiation. SoNICS natively handles automatic differentiation for linearized and adjoint code and for backpropagation, using forward and backward differentiation of the operator graph applied to the entire solver,
- SoNICS handles more easily cross-compatibilities between features thanks to its operator graph structure.

Shorter CPU time: with SoNICS, the generated source code is automatically optimized for each existing and future hardware. In addition to the design of an HPC layer leveraging all level of parallelism, SoNICS will strongly benefit from the impressive performance of current and future GPU hardware (e.g. NVIDIA's Hopper

H100 and followers). This opens for example new possibilities for scale resolved LES and DES/ZDES simulations with SoNICS, and, together with optimization which we are applying to the entire distribution (solver plus external software tools), this will strongly reduce turnaround time at design offices for day-to-day RANS calculations in large design parameter spaces.

Precision and robustness: SoNICS's architecture eases integration of future numerical or physical models thanks to its operators graph structure and to the opportunity to fit external operators developed by users into the operator graph.

SoNICS also enables shorter development cycles because of the use of automation. The developer codes only once, and optimized source code will be generated for every type of hardware. Then algorithmic differentiation automatically delivers the linearized and adjoint code. This strongly reduces the number of lines of source code which need to be developed and maintained.

SoNICS will not only perform high fidelity calculations. It will also make use of low fidelity models like body force which we use for example for meridian initialization. It can also be coupled with low resolution models. Body force is one example of implementation of an external operator in python, which then fits in the operator graph and benefits subsequently from all properties (differentiation, HPC layer, cross compatibilities) of operators inside the operator graph.

4 BRIDGING THE GAP BETWEEN INDUSTRIAL NEEDS AND ONERA'S CFD OFFER

We have put recently a lot of efforts building a new, more agile organization in order to better satisfy Safran end users' needs. This has been possible because Safran brings sustained financial support into the organization but also directly participates in a lot of processes, in particular into industrialization.

This organization relies on three pillars presented in the following paragraphs.

4.1 Handling users' needs instead of delivering features

There exist high-level software platforms for the different industrial components (e.g. HP turbine, LP turbine, compressor, booster, nacelle, combustion chamber, …). Theses platforms have pre-tuned numerical and physical parameters and encapsulate the CFD solver and dedicated pre and post-processing tools tailored to each design need, see fig. 5. To feed the software development plan, instead of periodically centralizing a list of features to be developed in a given period of time, we have put in place an agile organization which aims at satisfying users' need. In fact, users are e.g. design engineers who are experts at their craft but less skilful in numerical analysis. In order to develop the right features in the right software layers, we organize teams around specific users' needs. Product owners collect users' needs, gather a team of developers for each software layer and select testers from the applied engineering/design offices teams.

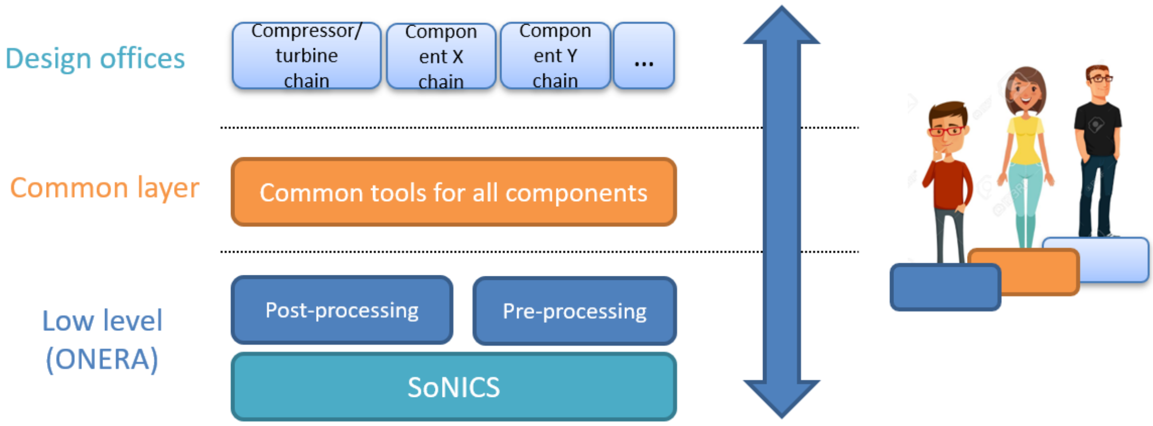

Figure 5: software layers with various types of tools. Industrial end-users have access to the top-level tools with pre-tuned parameters and models. Developing a coherent distribution of tools involves coordination between end-users, developers and testers at the various levels. Coordination is handled by product owners.

End users express their needs in the language of their craft. They ask for example "how can I perform an unsteady calculation for this multi-stage turbine with cooling holes". The product owner will decompose the need into features, and with the team of software developers and applied team testers, will decompose the resulting features into tasks to be performed in the various software layers, see fig. 6.

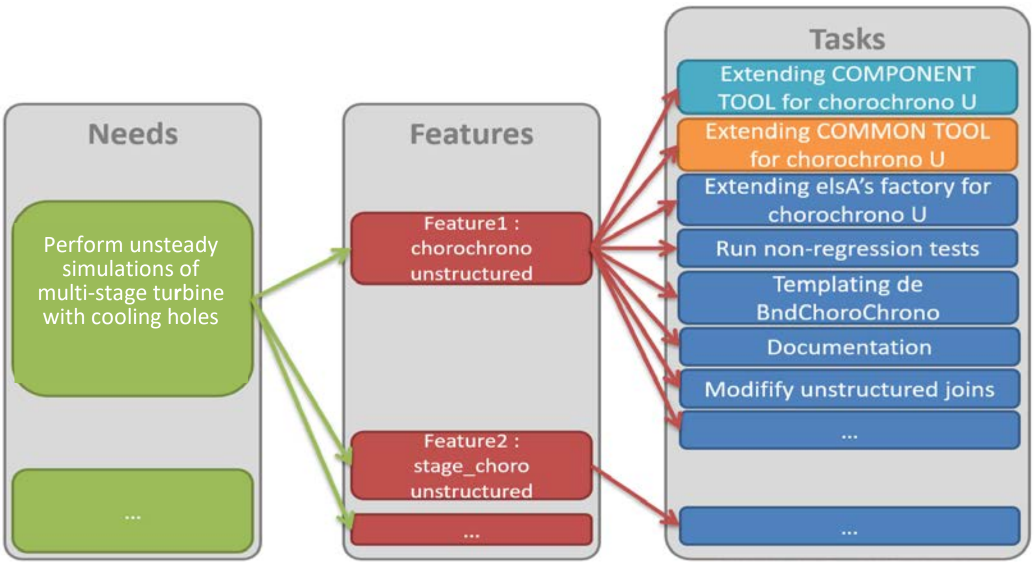

Figure 6: users' needs are split into features and tasks. Tasks are performed in various software layers.

4.2 Having the end user involved at all steps

For each need, the end user is involved into all steps of the process. A use case, consisting of a test case and testing environment, is defined at the beginning of the process. Several meetings are organized periodically for each need which gather the end user, product owner, developers of the various software layer and testers.

Acceptance tests are performed on the end user machine within the software chain of the design office dedicated to the aircraft component. Planning is performed by splitting the need into features, themselves being split into tasks, see fig 7. Tasks are assigned to individuals or group of individuals. Because individual developers and testers work in various software layers and at different organizations, coordination and synchronization by the product owner is paramount.

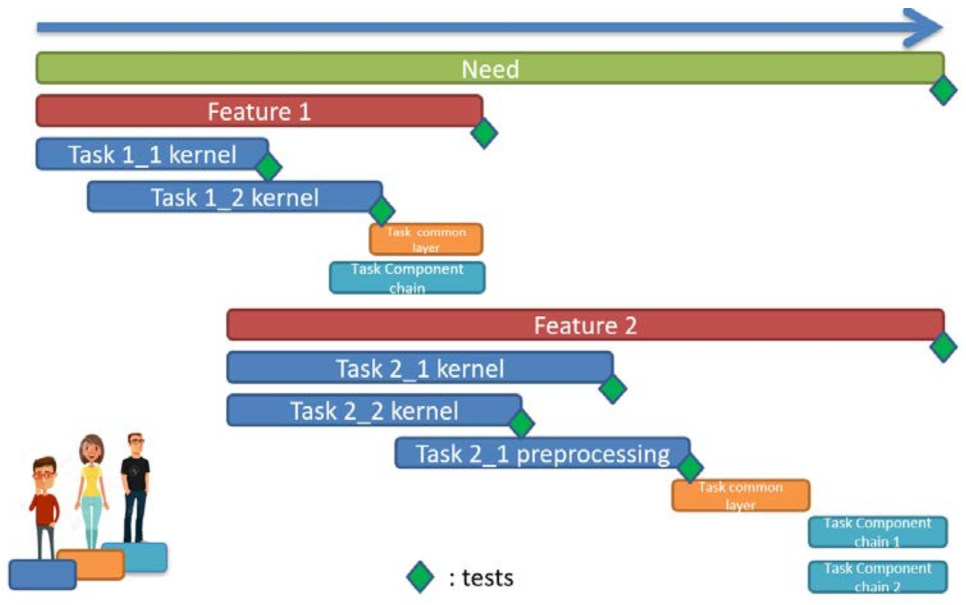

Figure 7: planning for team handling a specific need decomposed into features and tasks in the various software layers.

4.3 Testing in the industrial end users' environment

ONERA does not ship only a solver and a few tools designed by its teams, but instead must deliver to the end users an entire distribution with solver and software tools designed by teams at different organizations (ONERA, Safran Aircraft Engines, Safran Helicopter Engines, Safran Aero Boosters, Safran Tech, other industries, academics and various sub-contractors) working at remote locations. One challenge is to ship a distribution of tools at each different layer (as shown above on fig. 5) which are coherent and operate together with perfect compatibility. The version of each tool is tagged at each release of the distribution (e.g. Distribution_v2.0 = elsA_v5.1.1 – Tool1_v1.0 – Tool2_v2.1 – Tool3_v0.5) and a set of tests must be passed in order to guaranty that each feature functions correctly.

To do so, we define use cases for each need which are industrial-scale test cases plus a well defined user environment, and simplify the test case in order to perform the development of each feature in the solver and software tools in ONERA's environment. These tests are stored in various databases both at ONERA and Safran. ONERA stores the simplified development test cases in a development non-regression basis. Each developer runs the entire non-regression basis at each source code commit. At each release of the distribution, each organization runs (usually performed by each tool integrator) its own non-regression basis and validation basis. These tests are performed on different hardware/operating systems.

The use of all tools in every layer of a common pivot python/CGNS standard eases a lot the development of interfaces between tools and the exchange of use cases between organizations. Sharing the same single standard also eases support for handling bugs and anomalies in the distribution.

In order to avoid issues when shipping distributions to Safran end users, ONERA adds to its validation basis larger size industrial test cases from Safran bases which are representative of industrial needs (e.g. multi-stage axial turbines, centrifugal compressor, axial compressors, …). However, Safran cannot share with ONERA most of its configurations because of confidentiality restrictions.

While ONERA ships the distribution of its tools to Safran three times a year, the planning of these releases is not synchronized with the acceptance tests of features and needs (illustrated above on fig. 7). In order to ship and test the distribution more often, we are planning to move the organization in the near future to Continuous Integration and Continuous Deployment (CI/CD). Opening access to common source repositories across organizations and providing access to shared computing resources for running acceptance cases are necessary steps which we are tackling right now.

The three principles above have guided the development of dashboards used by teams at Safran and ONERA to plan and follow the development of elsA and SoNICS so that these solvers satisfy users' needs.

5 RECENT APPLICATIONS WITH ELSA AND SONICS MATCHING USERS NEEDS

In the following paragraphs, we showcase a few recent new features developed in elsA thanks to the new agile organization. We then show the first milestones achieved with the newly born SoNICS.

5.1 A few recent features of elsA originating from Safran end users' needs

Because our partner Safran starts ambitious programs for new aircraft motor configurations, motor designers need many additional features in elsA in order to obtain better simulations representativity (geometry and mesh strategy, physical models like turbulence, transition, real gases, boundary conditions dedicated to turbomachinery, …), precision and CPU performance. We exhibit below a few recent features among numerous other features developed during the two past years.

Non-reflecting boundary conditions

Future motor configurations will be more compact with faster rotation speed. In order to simulate such configurations, it is difficult to avoid putting boundary conditions nearby blades, and artificial reflection of waves (shocks, rarefaction waves …) occur on these boundary conditions. Following and extending the theoretical work of Giles [41], we have developed non-reflecting boundary conditions which strongly dampen spurious reflections on inlet, outlet and mixing plane boundary conditions, as shown on fig. 8. With these new boundary conditions, we obtain a strong dampening of the shock reflection together with almost negligible discrepancies on quantities of interest compared with reference calculations on a very long domain: flow rate < 0.05%, compression rate < 0.09% and efficiency < 0.47%.

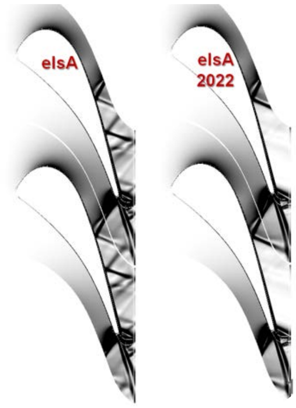

Figure 8: left: older version of elsA exhibiting reflection of shocks on outlet boundary condition located near blade row, right: April 2022 version of elsA with dampened reflection on the outlet condition

This feature is currently validated both at ONERA and Safran using increasingly complex test cases (axial and centrifugal compressors) representative of design offices' needs.

Stage reduction boundary condition on unstructured meshes

Adapting the stage reduction boundary condition to unstructured meshes involves complex computational geometry algorithms, which have to be performed in distributed parallel environment. One needs to perform optimally mesh intersections on both sides of the join and distributed HPC communication strategies must also be optimized. This type of boundary conditions enables unsteady turbomachinery calculations. A large number of validation cases has been successfully passed before shipping this feature to the design offices who will shortly use it in production. Figure 9 illustrates validation on a Vega 3 test case.

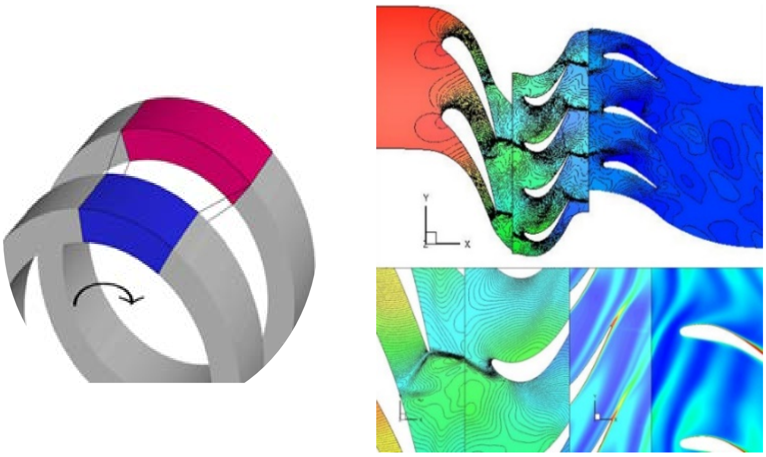

Figure 9: stage reduction join validated on Vega 3 case.

CPU acceleration of implicit calculations with elsA

We illustrate a development where elsA has benefited from innovations of an early prototype of the SoNICS software. In this prototype, we have implemented an algorithm for ordering the unknowns along wavefront hyperplanes inside the mesh. These ordering and additional vectorization optimizations led to a x7 performance speedup of implicit LU-SGS calculations on unstructured meshes with the SoNICS prototype compared with elsA. We have transferred some of the ingredients of this algorithm (not all of them because of elsA design limitations) into the elsA solver. Calculations by Safran of a Tatef compressor (fig. 10) exhibit a x2.6 performance speedup while ONERA ZDES calculations of a shear layer exhibit a x3.1 performance speedup.

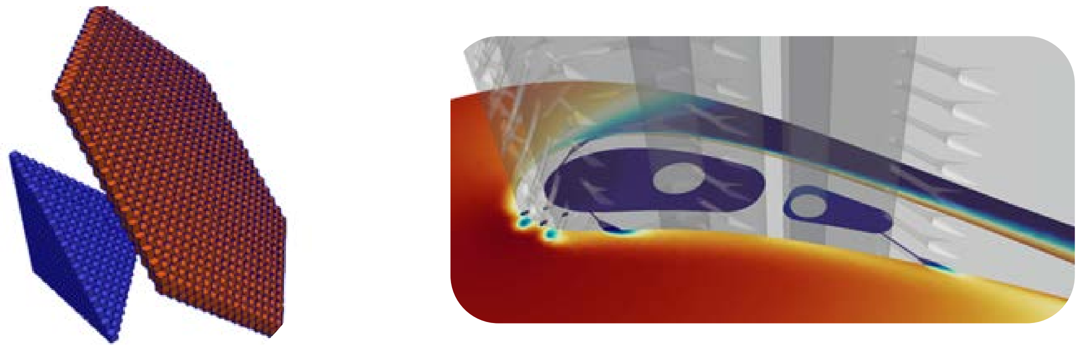

Figure 10: left : waverfront ordering of the unknowns, left : Tatef compressor calculation showing that the current version of elsA is 2.6 times faster than older (before 2020) versions

5.2 First milestones achieved with SoNICS

We have finalized in the past months many of the key elements of the SoNICS architecture. A few milestones have been successfully achieved. We have run test cases which concern HPC and numerical schemes:

- An operator graph and numerical operators have been generated both for CPU and GPU calculations.
- Two versions of the numerical schemes, one cell-centered, the other node-centered, have been set up. This second version allows for more robust calculations on anisotropic unstructured meshes which we target in our dynamic mesh adaptation strategy.

Figure 11 shows an Euler calculation with a Roe scheme on an M6 wing at low Mach number using a cell-centered discretization.

In addition, we have measured notable acceleration of the HPC performances of SoNICS compared with elsA:

- On CPU processors: we have measured a x14 acceleration between SoNICS and a 2020 version of elsA (on same hardware, same configuration, same calculation) thanks to the optimizations contained in the HPC layer.
- On GPUs: a x100 acceleration compared to the same calculation on CPUs. This is the expected result which corresponds grossly to the flops difference between the two hardware architectures.

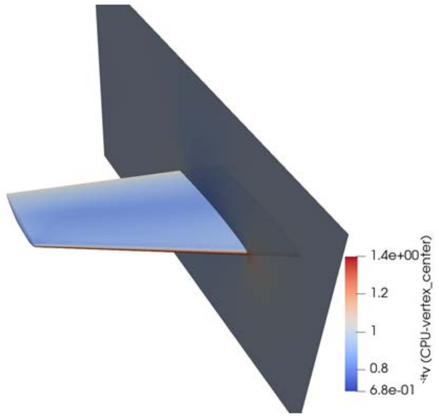

Figure 11: Euler calculation of M6 wing using SoNICS

6 CONCLUSIONS

We have shown in this paper the drivers which have led ONERA to switch to the development of an innovative CFD software named SoNICS, while continuing development of the current elsA solver for the medium term needs of our partner Safran. While we progressively scale down elsA, SoNICS will evolve to an industrial-scale software with a feature scope larger than elsA. SoNICS will also bring breakthrough capabilities which we believe will provide a competitive edge to Safran for its future motor programs. The robust organization around software development and industrialization will help aligning the software distribution to users' needs. While we continue on our roadmap and increase momentum by switching more skilled developers and application engineers to SoNICS, progressively achieving major development milestones, we are confident that the differentiating capabilities brought by SoNICS will help Safran and other ONERA partners achieve successes in future industrial programs.

Acknowledgments: we thank the participants of the SONICE project and Safran and ONERA management for valuable input and discussions. Especially, we thank all those involved in the development of elsA and SoNICS, who worked on producing the results and many of the illustrations presented in this paper. We finally thank DGAC for its support within the framework of *France Relance*.

REFERENCES

[1] L. Cambier, S. Heib, S. Plot. "The Onera elsA CFD software: input from research and feedback from industry". Mechanics & Industry, 14(3):159–174, 2013.

[2] A. Lerat, P. Cinnella, B. Michel, F. Falissard, "High order residual-based compact schemes for aerodynamics and aeroacoustics", Computers & Fluids, Vol. 61, pp. 31-38, 2012.

[3] F. Sicot, G. Dufour, N. Gourdain, "A time-domain harmonic balance method for rotor/stator interactions", J. of Turbomachinery, 134(1):011001 (13 pp.), 2012.

[4] G. Pont, P. Brenner, P. Cinnella, B. Maugars, J.-C. Robinet, Multiple-correction hybrid k-exact schemes for high-order compressible RANS-LES simulations on fully unstructured grids, Journal of Computational Physics, Volume 350, pp 45-83, 2017.

[5] S. Le Bras, H. Deniau, C. Bogey, G. Daviller, "Development of Compressible Large-Eddy Simulations Combining High-Order Schemes and Wall Modeling", AIAA Journal Vol. 55, No. 4, pp. 1152-1163, Avril 2017.

[6] B. Aupoix, D. Arnal, H. Bézard, B. Chaouat, F. Chedevergne, S. Deck, V. Gleize, P. Grenard, E. Laroche,"Turbulence and transition modeling", Aerospace Lab, Issue 2, 2011.

[7] H. Bézard, T. Daris, "Calibrating the Length Scale Equation with an Explicit Algebraic Reynolds Stress Constitutive Relation", 6th Int. Symp. on Engineering Turbulence Modelling and Measurements, W. Rodi and M. Mulas editors, Elsevier, pp. 77-86, 2005.

[8] C. Content, R. Houdeville., "Application of the γ-Reθt laminar-turbulent transition model in Navier-Stokes computations". AIAA, 40th Fluid Dynamics Conference and Exhibit, 2010-4445, June 2010.

[9] G. Bégou, H. Deniau, O. Vermeersch, G. Casalis, "Database Approach for Laminar-Turbulent Transition Prediction: Navier-Stokes Compatible Reformulation", AIAA Journal, Vol. 55, No. 11 (2017), pp. 3648-3660.

[10] L. Jecker, O. Vermeersch, H. Deniau, E. Croner, G. Casalis, "A laminar kinetic energy model based on the Klebanoff-mode dynamics to predict bypass transition", European Journal of Mechanics / B Fluids, Vol. 74, pp 265-279, 2019

[11] Bouchard, M., Marty, J., Deck, S., Costes, M., "Validation of correlations-based transition modeling strategies applied to the Spalart-Allmaras turbulence model for the computation of separation-induced transition", Aerospace Science and Technology", 119,107045, 2021.

[12] A. Dugeai, Y. Mauffrey, F. Sicot, "Aeroelastic capabilities of the elsA solver for rotating machines applications", International Forum on Aeroelasticity and Structural Dynamics, Paris, 2011

[13] Huvelin, F., Lepage, A., Dequand, S., "Experimental and numerical investigations of a 2D aeroelastic airfoil encountering a gust in transonic conditions", CEAS Aeronautical Journal, 10(4), pp. 1101-1120, 2019.

[14] S. Plot. "The High Level of Maturity of the elsA CFD Software for Turbomachinery Aerodynamics Applications". In: ISABE 2019. 2019.

[15] Baretter, A., Godard, B., Joseph, P., Roussette, O., Romano, F. , Barrier, R., Dazin, A., “Experimental and numerical analysis of a compressor stage under flow distortion”, International Journal of Turbomachinery, Propulsion and Power, 6(4),43, 2021.

[16] Barrier R., Billonnet G., Marty J., Pascal L., “Linear Stability Analysis in Rotating Frames for Fan Blade Transition Prediction”, AIAA Journal - Vol. 58, No. 8, pp. 3264-3271, 2020.

[17] Carrier, G., Blondeau, C., Moëns F., Carini, M., Cadillon, J., Le Garrec, T., Mincu, D., “Illustrations of multi-disciplinary high-fidelity analysis capabilities for aero-structural and aero-acoustic aircraft design”, AIAA Scitech 2021 Forum, pp. 1-33.

[18] Sartor, F., Martin, M., Moreno, C.R., Dandois, J., "Numerical simulations of transonic buffet on a half wing-body configuration”, 32nd Congress of the International Council of the Aeronautical Sciences, 2021.

[19] C. Lienard, I. Salah El Din, T. Renaud, R. Fukari “RACER high-speed demonstrator: Rotor and rotor-head wake interactions with tail unit”, American Helicopter Society, 74th Annual Forum, Phoenix, Arizona, May 14-17, 2018.

[20] Castells C., Richez F., Costes M., “A Numerical Investigation of the Influence of the Blade–Vortex Interaction on the Dynamic Stall Onset”, Journal of the American Helicopter Society, 2021.

[21] Boisard, R., “Aerodynamic Analysis of Rotor/Propeller Wakes Interactions on High Speed Compound Helicopter”, 47th European Rotorcraft Forum, 2021.

[22] Castillon, L., Soismier, M., Le Pape, M.C., Maugars, B., Michel, B., “A hybrid structured/unstructured grid strategy for the CFD modeling of technological effects on complex turbomachinery applications,- ISABE congress 2019.

[23] L. Castillon; G. Billonnet; J. Riou; S. Péron; C. Benoit, "A Technological Effect Modeling on Complex Turbomachinery Applications With an Overset Grid Numerical Method", Journal of Turbomachinery , October 2014.

[24] Awes, A., Dufour, G., Daon, R., Marty, J., Barrier, R., Carbonneau, X., “Unsteady Body Force Methodology for Fan Operability Assessment under Clean and Distorted Inflow Conditions”, AIAA Scitech forum 2021.

[25] Castillon, L., Gourdain, N., and Ottavy, X., "Multiple-Frequency Phase-Lagged Unsteady Simulations of Experimental Axial Compressor", J. of Propulsion and Power, (2015), accessed January 06, 2015, volume 31, issue 1 on pages 444-455.

[26] Marty, J., Uribe, C., “Impact of Underlying RANS Turbulence Models in Zonal Detached Eddy Simulation: Application to a Compressor Rotor”, International Journal of Turbomachinery, Propulsion and Power - Vol. 5, No. 3, p.22., 2020.

[27] Benjamin F., Barrier, R., Polacsek, C., “Zonal Detached Eddy Simulation of the Fan-OGV Stage of a Modern Turbofan Engine”, ASME Turbo Expo, 2020.

[28] Placzek, A., “Assessment of single passage simulations for the aeroelastic stability of a multistage centrifugal compressor”, 18th Int. Symp. on Transport Phenomena and Dynamics of Rotating Machinery, 2020

[29] Carini, M., Blondeau, C., Fabbiane, N., Meheut, M., Abu-Zurayk, M., Feldwisch, J., Ilic, C., Merle, A., “Towards industrial aero-structural aircraft optimization via coupled-adjoint derivatives”, AIAA Aviation 2021 Forum, 2021.

[30] Wilke G., Bailly, J., Kimura, K., Tanabe, Y., “JAXA-ONERA-DLR cooperation: results from rotor optimization in hover”, CEAS Journal, 2021.

[31] Song, Y., & Li, Z. (1999). New tiling techniques to improve cache temporal locality. ACM SIGPLAN Notices, 34(5), 215-228.

[32] Demmel, J., Grigori, L., Hoemmen, M., & Langou, J. (2012). Communication-optimal parallel and sequential QR and LU factorizations. SIAM Journal on Scientific Computing, 34(1), A206-A239.

[33] Hascoet, L., & Pascual, V. (2013). The Tapenade automatic differentiation tool: principles, model, and specification. ACM Transactions on Mathematical Software (TOMS), 39(3), 1-43.

[34] Moses, W. S., Churavy, V., Paehler, L., Hückelheim, J., Narayanan, S. H. K., Schanen, M., & Doerfert, J. (2021, November). Reverse-mode automatic differentiation and optimization of GPU kernels via enzyme. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-16).

[35] J. Bradbury et al., JAX: Composable transformations of Python+NumPy programs. http://github.com/google/jax. Deposited 15 December 2020

[36] Wolf, Marilyn. Computers as components: principles of embedded computing system design. Elsevier, 2012.

[37] Ferrante, J., Ottenstein, K. J., & Warren, J. D. The program dependence graph and its use in optimization. ACM Transactions on Programming Languages and Systems (TOPLAS), 9(3), 319-349. 1987.

[38] F. Damiani, M. Lienhardt, L. Paolini, “A formal model for multi software product lines”. Science of Computer Programming, 172, 203-231, 2019.

[39] Chelini, L., Drebes, A., Zinenko, O., Cohen, A., Vasilache, N., Grosser, T., & Corporaal, H., “Progressive raising in multi-level IR”. In 2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO) (pp. 15-26). IEEE, 2021.

[40] C. Lattner et al., "MLIR: Scaling Compiler Infrastructure for Domain Specific Computation". In 2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO),, pp. 2-14, IEEE, 2021.

[41] M. B. Giles, “Nonreflecting boundary conditions for Euler equation calculations”. AIAA journal, 28(12), 2050-205. 1990.