



Activity Report 2016

Project-Team ECUADOR

Program transformations for scientific computing

RESEARCH CENTER
Sophia Antipolis - Méditerranée

THEME
Numerical schemes and simulations

Table of contents

1. Members	1
2. Overall Objectives	2
3. Research Program	2
3.1. Algorithmic Differentiation	2
3.2. Static Analysis and Transformation of programs	3
3.3. Algorithmic Differentiation and Scientific Computing	4
4. Application Domains	5
4.1. Algorithmic Differentiation	5
4.2. Multidisciplinary optimization	5
4.3. Inverse problems and Data Assimilation	5
4.4. Linearization	7
4.5. Mesh adaptation	7
5. New Software and Platforms	7
5.1. AIRONUM	7
5.2. TAPENADE	7
6. New Results	8
6.1. AD-adjoints and C dynamic memory management	8
6.2. AD-adjoints of MPI-parallel codes	8
6.3. AD-adjoints of Iterative Processes	9
6.4. AD of mixed-language codes	9
6.5. Multirate methods	9
6.6. Application of AD to uncertainties and errors in CFD	10
6.7. Control of approximation errors	10
6.8. Turbulence models	11
7. Bilateral Contracts and Grants with Industry	11
8. Partnerships and Cooperations	11
8.1. National Initiatives	11
8.2. European Initiatives	12
8.2.1.1. AboutFlow	12
8.2.1.2. UMRIDA	12
8.3. International Initiatives	12
8.4. International Research Visitors	13
8.4.1. Visits of International Scientists	13
8.4.2. Internships	13
8.4.3. Visits to International Teams	13
9. Dissemination	13
9.1. Promoting Scientific Activities	13
9.2. Teaching - Supervision - Juries	13
9.2.1. Teaching	13
9.2.2. Supervision	13
9.2.3. Juries	13
9.3. Popularization	13
10. Bibliography	13

Project-Team ECUADOR

Creation of the Project-Team: 2014 January 01

Keywords:

Computer Science and Digital Science:

- 2.1.1. - Semantics of programming languages
- 2.2.1. - Static analysis
- 2.5. - Software engineering
- 6.1.1. - Continuous Modeling (PDE, ODE)
- 6.2.6. - Optimization
- 6.2.7. - High performance computing
- 6.3.1. - Inverse problems
- 6.3.2. - Data assimilation

Other Research Topics and Application Domains:

- 1.1.2. - Molecular biology
- 3.2. - Climate and meteorology
- 3.3.2. - Water: sea & ocean, lake & river
- 3.3.4. - Atmosphere
- 5.2.3. - Aviation
- 5.2.4. - Aerospace
- 9.5.3. - Economy, Finance

1. Members

Research Scientists

Laurent Hascoët [Team leader, Inria, Senior Researcher, HDR]
Alain Dervieux [Inria, Senior Researcher, HDR]
Valérie Pascual [Inria, Researcher]

PhD Students

Eléonore Gauci [Inria]
Emmanuelle Itam [Univ. Montpellier II, from Feb 2016]
Ala Taftaf [Inria]

Visiting Scientists

Olivier Allain [Lemma, Visiting Scientist]
Gautier Brêthes [Visiting Scientist, until Mar 2016]
Georgios Ntanakas [Rolls-Royce, Dahlewitz, Germany, Visiting Scientist, until Feb 2016]

Administrative Assistant

Christine Claux [Inria]

Others

Bruno Koobus [Univ. Montpellier II, Professor]
Stephen Wornom [Lemma, Engineer]

2. Overall Objectives

2.1. Overall Objectives

Team Ecuador studies Algorithmic Differentiation (AD) of computer programs, blending :

- **AD theory:** We study software engineering techniques, to analyze and transform programs mechanically. Algorithmic Differentiation (AD) transforms a program P that computes a function F , into a program P' that computes analytical derivatives of F . We put emphasis on the *adjoint mode* of AD, a sophisticated transformation that yields gradients for optimization at a remarkably low cost.
- **AD application to Scientific Computing:** We adapt the strategies of Scientific Computing to take full advantage of AD. We validate our work on real-size applications.

We want to produce AD code that can compete with hand-written sensitivity and adjoint programs used in the industry. We implement our algorithms into the tool Tapenade, one of the most popular AD tools now.

Our research directions :

- Efficient adjoint AD of frequent dialects e.g. Fixed-Point loops.
- Development of the adjoint AD model towards Dynamic Memory Management.
- Development of the adjoint AD model towards Parallel Languages.
- Optimal shape design and optimal control for steady and unsteady simulations. Higher-order derivatives for uncertainty quantification.
- Adjoint-driven mesh adaptation.

3. Research Program

3.1. Algorithmic Differentiation

Participants: Laurent Hascoët, Valérie Pascual, Ala Taftaf.

algorithmic differentiation (AD, aka Automatic Differentiation) Transformation of a program, that returns a new program that computes derivatives of the initial program, i.e. some combination of the partial derivatives of the program's outputs with respect to its inputs.

adjoint Mathematical manipulation of the Partial Differential Equations that define a problem, obtaining new differential equations that define the gradient of the original problem's solution.

checkpointing General trade-off technique, used in adjoint AD, that trades duplicate execution of a part of the program to save some memory space that was used to save intermediate results.

Algorithmic Differentiation (AD) differentiates *programs*. The input of AD is a source program P that, given some $X \in \mathbb{R}^n$, returns some $Y = F(X) \in \mathbb{R}^m$, for a differentiable F . AD generates a new source program P' that, given X , computes some derivatives of F [6].

Any execution of P amounts to a sequence of instructions, which is identified with a composition of vector functions. Thus, if

$$\begin{aligned} P & \text{ runs } \{I_1; I_2; \dots; I_p\}, \\ F & \text{ then is } f_p \circ f_{p-1} \circ \dots \circ f_1, \end{aligned} \quad (1)$$

where each f_k is the elementary function implemented by instruction I_k . AD applies the chain rule to obtain derivatives of F . Calling X_k the values of all variables after instruction I_k , i.e. $X_0 = X$ and $X_k = f_k(X_{k-1})$, the Jacobian of F is

$$F'(X) = f'_p(X_{p-1}) \cdot f'_{p-1}(X_{p-2}) \cdot \dots \cdot f'_1(X_0) \quad (2)$$

which can be mechanically written as a sequence of instructions I'_k . This can be generalized to higher level derivatives, Taylor series, etc. Combining the I'_k with the control of P yields P' , and therefore this differentiation is piecewise.

In practice, many applications only need cheaper projections of $F'(X)$ such as:

- **Sensitivities**, defined for a given direction \dot{X} in the input space as:

$$F'(X).\dot{X} = f'_p(X_{p-1}) \cdot f'_{p-1}(X_{p-2}) \cdot \dots \cdot f'_1(X_0) \cdot \dot{X} \quad . \quad (3)$$

This expression is easily computed from right to left, interleaved with the original program instructions. This is the *tangent mode* of AD.

- **Adjoins**, defined after transposition (F'^*), for a given weighting \bar{Y} of the outputs as:

$$F'^*(X).\bar{Y} = f'^*_1(X_0) \cdot f'^*_2(X_1) \cdot \dots \cdot f'^*_{p-1}(X_{p-2}) \cdot f'^*_p(X_{p-1}) \cdot \bar{Y} \quad . \quad (4)$$

This expression is most efficiently computed from right to left, because matrix×vector products are cheaper than matrix×matrix products. This is the *adjoint mode* of AD, most effective for optimization, data assimilation [37], adjoint problems [32], or inverse problems.

Adjoint AD builds a very efficient program [34], which computes the gradient in a time independent from the number of parameters n . In contrast, computing the same gradient with the *tangent mode* would require running the tangent differentiated program n times.

However, the X_k are required in the *inverse* of their computation order. If the original program *overwrites* a part of X_k , the differentiated program must restore X_k before it is used by $f'^*_{k+1}(X_k)$. Therefore, the central research problem of adjoint AD is to make the X_k available in reverse order at the cheapest cost, using strategies that combine storage, repeated forward computation from available previous values, or even inverted computation from available later values.

Another research issue is to make the AD model cope with the constant evolution of modern language constructs. From the old days of Fortran77, novelties include pointers and dynamic allocation, modularity, structured data types, objects, vectorial notation and parallel programming. We keep developing our models and tools to handle these new constructs.

3.2. Static Analysis and Transformation of programs

Participants: Laurent Hascoët, Valérie Pascual, Ala Taftaf.

abstract syntax tree Tree representation of a computer program, that keeps only the semantically significant information and abstracts away syntactic sugar such as indentation, parentheses, or separators.

control flow graph Representation of a procedure body as a directed graph, whose nodes, known as basic blocks, each contain a sequence of instructions and whose arrows represent all possible control jumps that can occur at run-time.

abstract interpretation Model that describes program static analysis as a special sort of execution, in which all branches of control switches are taken concurrently, and where computed values are replaced by abstract values from a given *semantic domain*. Each particular analysis gives birth to a specific semantic domain.

data flow analysis Program analysis that studies how a given property of variables evolves with execution of the program. Data Flow analysis is static, therefore studying all possible run-time behaviors and making conservative approximations. A typical data-flow analysis is to detect, at any location in the source program, whether a variable is initialized or not.

The most obvious example of a program transformation tool is certainly a compiler. Other examples are program translators, that go from one language or formalism to another, or optimizers, that transform a program to make it run better. AD is just one such transformation. These tools share the technological basis that lets them implement the sophisticated analyses [25] required. In particular there are common mathematical models to specify these analyses and analyze their properties.

An important principle is *abstraction*: the core of a compiler should not bother about syntactic details of the compiled program. The optimization and code generation phases must be independent from the particular input programming language. This is generally achieved using language-specific *front-ends*, language-independent *middle-ends*, and target-specific *back-ends*. In the middle-end, analysis can concentrate on the semantics of a reduced set of constructs. This analysis operates on an abstract representation of programs made of one *call graph*, whose nodes are themselves *flow graphs* whose nodes (*basic blocks*) contain abstract *syntax trees* for the individual atomic instructions. To each level are attached symbol tables, nested to capture scoping.

Static program analysis can be defined on this internal representation, which is largely language independent. The simplest analyses on trees can be specified with inference rules [28], [35], [26]. But many *data-flow analyses* are more complex, and better defined on graphs than on trees. Since both call graphs and flow graphs may be cyclic, these global analyses will be solved iteratively. *Abstract Interpretation* [29] is a theoretical framework to study complexity and termination of these analyses.

Data flow analyses must be carefully designed to avoid or control combinatorial explosion. At the call graph level, they can run bottom-up or top-down, and they yield more accurate results when they take into account the different call sites of each procedure, which is called *context sensitivity*. At the flow graph level, they can run forwards or backwards, and yield more accurate results when they take into account only the possible execution flows resulting from possible control, which is called *flow sensitivity*.

Even then, data flow analyses are limited, because they are static and thus have very little knowledge of actual run-time values. Far before reaching the very theoretical limit of *undecidability*, one reaches practical limitations to how much information one can infer from programs that use arrays [41], [30] or pointers. Therefore, conservative *over-approximations* must be made, leading to derivative code less efficient than ideal.

3.3. Algorithmic Differentiation and Scientific Computing

Participants: Alain Dervieux, Laurent Hascoët, Bruno Koobus.

linearization In Scientific Computing, the mathematical model often consists of Partial Differential Equations, that are discretized and then solved by a computer program. Linearization of these equations, or alternatively linearization of the computer program, predict the behavior of the model when small perturbations are applied. This is useful when the perturbations are effectively small, as in acoustics, or when one wants the sensitivity of the system with respect to one parameter, as in optimization.

adjoint state Consider a system of Partial Differential Equations that define some characteristics of a system with respect to some input parameters. Consider one particular scalar characteristic. Its sensitivity, (or gradient) with respect to the input parameters can be defined as the solution of “adjoint” equations, deduced from the original equations through linearization and transposition. The solution of the adjoint equations is known as the adjoint state.

Scientific Computing provides reliable simulations of complex systems. For example it is possible to *simulate* the steady or unsteady 3D air flow around a plane that captures the physical phenomena of shocks and turbulence. Next comes *optimization*, one degree higher in complexity because it repeatedly simulates and applies gradient-based optimization steps until an optimum is reached. The next sophistication is *robustness* i.e. to detect and to lower preference to a solution which, although maybe optimal, is very sensitive to uncertainty on design parameters or on manufacturing tolerances. This makes second derivative come into play. Similarly *Uncertainty Quantification* can use second derivatives to evaluate how uncertainty on the simulation inputs imply uncertainty on its outputs.

We investigate several approaches to obtain the gradient, between two extremes:

- One can write an *adjoint system* of mathematical equations, then discretize it and program it by hand. This is time consuming. Although this looks mathematically sound [32], this does not provide the gradient of the discretized function itself, thus degrading the final convergence of gradient-descent optimization.
- One can apply adjoint AD (*cf* 3.1) on the program that discretizes and solves the direct system. This gives exactly the adjoint of the discrete function computed by the program. Theoretical results [31] guarantee convergence of these derivatives when the direct program converges. This approach is highly mechanizable, but leads to massive use of storage and may require code transformation by hand [36], [39] to reduce memory usage.

If for instance the model is steady, or when the computation uses a Fixed-Point iteration, tradeoffs exist between these two extremes [33], [27] that combine low storage consumption with possible automated adjoint generation. We advocate incorporating them into the AD model and into the AD tools.

4. Application Domains

4.1. Algorithmic Differentiation

Algorithmic Differentiation of programs gives sensitivities or gradients, useful for instance for :

- optimum shape design under constraints, multidisciplinary optimization, and more generally any algorithm based on local linearization,
- inverse problems, such as parameter estimation and in particular 4Dvar data assimilation in climate sciences (meteorology, oceanography),
- first-order linearization of complex systems, or higher-order simulations, yielding reduced models for simulation of complex systems around a given state,
- mesh adaptation and mesh optimization with gradients or adjoints,
- equation solving with the Newton method,
- sensitivity analysis, propagation of truncation errors.

4.2. Multidisciplinary optimization

A CFD program computes the flow around a shape, starting from a number of inputs that define the shape and other parameters. On this flow one can define optimization criteria e.g. the lift of an aircraft. To optimize a criterion by a gradient descent, one needs the gradient of the criterion with respect to all inputs, and possibly additional gradients when there are constraints. Adjoint AD is the most efficient way to compute these gradients.

4.3. Inverse problems and Data Assimilation

Inverse problems aim at estimating the value of hidden parameters from other measurable values, that depend on the hidden parameters through a system of equations. For example, the hidden parameter might be the shape of the ocean floor, and the measurable values of the altitude and velocities of the surface.

One particular case of inverse problems is *data assimilation* [37] in weather forecasting or in oceanography. The quality of the initial state of the simulation conditions the quality of the prediction. But this initial state is not well known. Only some measurements at arbitrary places and times are available. A good initial state is found by solving a least squares problem between the measurements and a guessed initial state which itself must verify the equations of meteorology. This boils down to solving an adjoint problem, which can be done though AD [40]. Figure 1 shows an example of a data assimilation exercise using the oceanography code OPA [38] and its AD-adjoint produced by Tapenade.

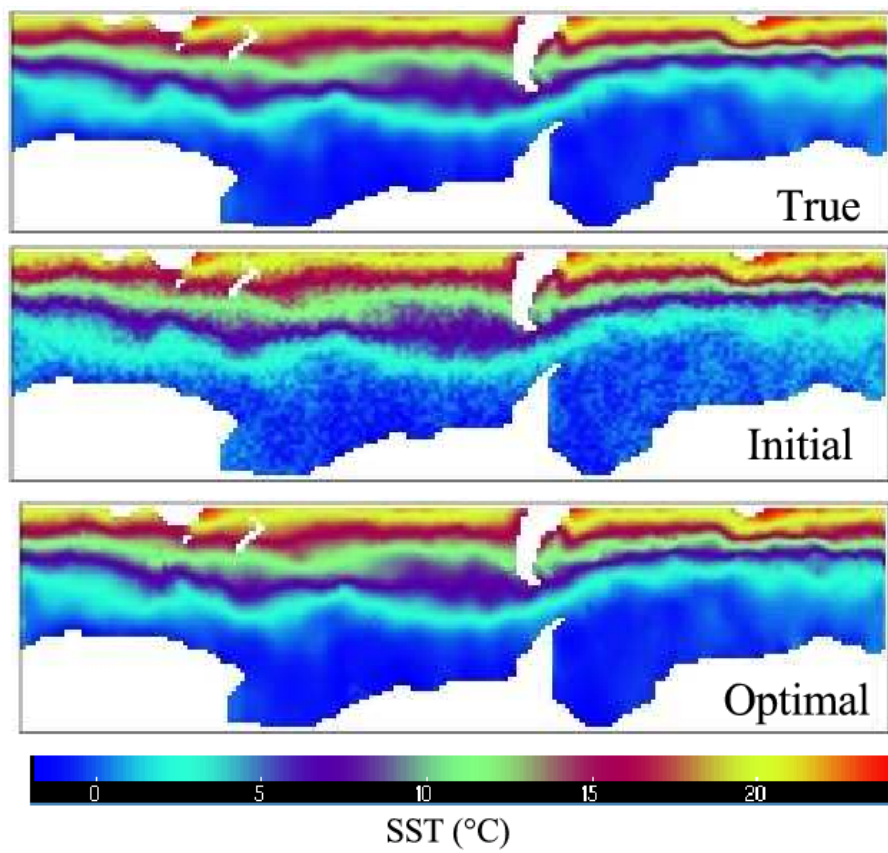


Figure 1. Twin experiment using the adjoint of OPA. Random noise, added to a simulation of the sea surface temperature around the Antarctic, is removed by minimizing the discrepancy with the physical model

The special case of *4Dvar* data assimilation is particularly challenging. The 4th dimension in “4D” is time, as available measurements are distributed over a given assimilation period. Therefore the least squares mechanism must be applied to a simulation over time that follows the time evolution model. This process gives a much better estimation of the initial state, because both position and time of measurements are taken into account. On the other hand, the adjoint problem involved is more complex, because it must run (backwards) over many time steps. This demanding application of AD justifies our efforts in reducing the runtime and memory costs of AD adjoint codes.

4.4. Linearization

Simulating a complex system often requires solving a system of Partial Differential Equations. This can be too expensive, in particular for real-time simulations. When one wants to simulate the reaction of this complex system to small perturbations around a fixed set of parameters, there is an efficient approximation: just suppose that the system is linear in a small neighborhood of the current set of parameters. The reaction of the system is thus approximated by a simple product of the variation of the parameters with the Jacobian matrix of the system. This Jacobian matrix can be obtained by AD. This is especially cheap when the Jacobian matrix is sparse. The simulation can be improved further by introducing higher-order derivatives, such as Taylor expansions, which can also be computed through AD. The result is often called a *reduced model*.

4.5. Mesh adaptation

Some approximation errors can be expressed by an adjoint state. Mesh adaptation can benefit from this. The classical optimization step can give an optimization direction not only for the control parameters, but also for the approximation parameters, and in particular the mesh geometry. The ultimate goal is to obtain optimal control parameters up to a precision prescribed in advance.

5. New Software and Platforms

5.1. AIRONUM

Aironum is an experimental software that solves the unsteady compressible Navier-Stokes equations with k-, LES-VMS (Large Eddy Simulation - Variational Multi-Scale) and hybrid turbulence modelling on parallel platforms, using MPI. The mesh model is unstructured tetrahedrization, with possible mesh motion.

Aironum was developed by Inria and University of Montpellier. It is used by Inria, University of Montpellier and University of Pisa. Aironum is used as an experimental platform for:

- Numerical approximation of compressible flows, such as upwind mixed element volume approximation with superconvergence on regular meshes.
- Numerical solution algorithms for the implicit time advancing of the compressible Navier-Stokes equations, such as parallel scalable deflated additive Schwarz algorithms.
- Turbulence modelling such as the Variational Multiscale Large eddy Simulation and its hybridization with RANS (Reynolds Averaged Navier-Stokes) statistical models.
- Participant: Alain Dervieux
- Contact: Alain Dervieux
- URL: <http://www-sop.inria.fr/tropics/aironum>

5.2. TAPENADE

KEYWORDS: Static analysis - Optimization - Compilation - Gradients

Tapenade [10] is an Algorithmic Differentiation tool that transforms an original program into a new program that computes derivatives of the original program. Being an AD tool, Tapenade produces analytical derivatives exact up to machine precision, and in adjoint mode computes gradients at a cost independent of the number of input variables.

Tapenade implements the results of our research about models and static analyses for AD. Tapenade can be downloaded and installed on most architectures. Alternatively, it can be used as a web server. Higher-order derivatives can be obtained through repeated application. Tapenade accepts source programs written in Fortran77, Fortran90, or C. It provides differentiation in the following modes: tangent, vector tangent, adjoint, and vector adjoint.

Tapenade performs sophisticated data-flow analysis, flow-sensitive and context-sensitive, on the complete source program to produce an efficient differentiated code. Analyses performed are Type-Checking, Read-Write analysis, Pointer analysis, and AD-specific analyses including Activity analysis, Adjoint Liveness analysis, and TBR analysis.

- Participants: Laurent Hascoët, Valérie Pascual
- Contact: Laurent Hascoët
- URL: <http://www-sop.inria.fr/tropics/tapenade.html>

6. New Results

6.1. AD-adjoints and C dynamic memory management

Participants: Laurent Hascoët, Sri Hari Krishna Narayanan [Argonne National Lab. (Illinois, USA)], Mathieu Morlighem [University of California at Irvine (USA)].

One of the current frontiers of AD research is the definition of an adjoint AD model that can cope with dynamic memory management. This research is central in our ongoing effort towards adjoint AD of C, and more remotely towards AD of C++. This research is conducted in collaboration with the MCS department of Argonne National Lab. Our partnership is formalized by joint participation in the Inria joint lab JLESC, and partly funded by the Partner University Fund (PUF) of the French embassy in the USA.

Adjoint AD must reproduce in reverse order the control decisions of the original code. In languages such as C, allocation of dynamic memory and pointer management form a significant part of these control decisions. Reproducing memory allocation in reverse means reallocating memory, possibly receiving a different memory chunk. Reproducing pointer addresses in reverse thus require to convert addresses in the former memory chunks into equivalent addresses in the new reallocated chunks. Together with Krishna Narayanan from Argonne, we experiment on real applications to find the most efficient solution to this address conversion problem. We jointly develop a library (called ADMM, ADjoint Memory Management) whose primitives are used in AD adjoint code to handle this address conversion. Both our AD tool Tapenade and Argonne's tool OpenAD use ADMM in the adjoint code they produce.

This year, ADMM was instrumental in the successful generation of the adjoint code of "ALIF" (formerly called "SEISM") by Tapenade. The "ALIF" code is developed by Mathieu Morlighem from UC Irvine, jointly with Eric Larour from JPL. This glaciology code is a C clone of the C++ "ISSM" code from JPL. One objective of this work is to clarify the C programming style that allows AD to perform better. Another objective is to make progress in the direction of generating adjoints of C++ code. Although ADMM has already been used with success for the adjoint of several small- to medium-size applications, and now on the large-size code "ALIF", we are still considering alternative implementation strategies. This work was presented at the AD2016 conference in Oxford [16], and an article is submitted to journal "Optimization Methods and Software".

6.2. AD-adjoints of MPI-parallel codes

Participants: Laurent Hascoët, Ala Taftaf, Georgios Ntanakas [Rolls-Royce, Dahlewitz, Germany], Sri Hari Krishna Narayanan [Argonne National Lab. (Illinois, USA)].

We have a long-standing collaboration with Argonne National Lab on the question of adjoint AD of message-passing parallel codes. We continued joint development of the Adjoinable-MPI library (AMPI) that provides efficient tangent and adjoint AD for MPI-parallel codes, independently of the AD tool used (now AdolC, dco, OpenAD, Tapenade).

Ala Taftaf considers the question of checkpointing applied to the AD-adjoint of an MPI-parallel code. Checkpointing is a memory/runtime tradeoff which is essential for adjoint AD of large codes, in particular parallel codes. However, for MPI codes this question has always been addressed by ad-hoc hand manipulations of the differentiated code, and with no formal assurance of correctness. Ala Taftaf studies these past experiments and proposes more general strategies. Ala Taftaf presented her results [20], [23] at the Eccomas 2016 conference (Crete) in June and at the NOED 2016 conference (Munich) in July [22].

During his secondment with our team, PhD student Georgios Ntanakas from Rolls-Royce studied possible extension of Tapenade to handle the parallel constructs in Rolls-Royce's "Hydra" code, which rely on a special parallel library named "OPlus".

6.3. AD-adjoints of Iterative Processes

Participants: Laurent Hascoët, Ala Taftaf, Sri Hari Krishna Narayanan [Argonne National Lab. (Illinois, USA)], Daniel Goldberg [University of Edinburgh, UK].

Ala Taftaf continued her work on the adjoint of iterative Fixed-Point loops. This year she studied refinements of the AD-specific data-flow analyses to adapt them to the specific shape of this adjoint code, proposed by Bruce Christianson [27]. She also proposed an efficient "warm-start" mechanism, that provides a good initial guess for the fixed-point loop that computes the adjoint, in the case where this fixed-point loop is itself enclosed in another loop. These results are described in her PhD document, to be defended in January 2017.

We published a journal article [13] on our joint work with Krishna Narayanan from ANL and Dan Goldberg from University of Edinburgh (UK), which applies in particular this fixed-point adjoint strategy to a glaciology configuration of the MIT GCM code.

6.4. AD of mixed-language codes

Participants: Valérie Pascual, Tom Verstraete [VKI, Brussels, Belgium], Laurent Hascoët.

In collaboration with Tom Verstraete, Valérie Pascual is applying Tapenade to the library "Calculix", whose implementation mixes Fortran and C. This library is well fit for Tapenade differentiation, as the internal representation that we use for codes is language-independent. We can thus load both Fortran and C source into Tapenade and differentiate the complete code transparently. Obviously, since this is the first application of Tapenade to a real-size mixed-language code, interesting problems arise mostly about parameter-passing strategies. Valérie Pascual presented her first results at the AD2016 conference in Oxford [21].

6.5. Multirate methods

Participants: Alain Dervieux, Bruno Koobus, Emmanuelle Itam, Stephen Wornom.

This study is performed in collaboration with IMAG-Montpellier II. It addresses an important complexity issue in unsteady mesh adaptation and takes place in the work done in the ANR Maidesc. Unsteady high-Reynolds computations are strongly penalized by the very small time-step imposed by accuracy requirements on regions involving small space-time scales. Unfortunately, this is also true for sophisticated unsteady mesh adaptive calculations. This small time-step is an important computational penalty for mesh adaptive methods of AMR type. This is also the case for the Unsteady Fixed-Point mesh-adaptive methods developed by Ecuador in cooperation with the Gamma3 team of Inria-Saclay. In the latter method, the loss of efficiency is even more crucial when the anisotropic mesh is locally strongly stretched. This loss is evaluated as limiting the numerical convergence order for discontinuities to 8/5 instead of second-order convergence. An obvious remedy is to design time-consistent methods using different time steps on different parts of the mesh, as far as they are efficient and not too complex. The family of time-advancing methods in which unsteady

phenomena are computed with different time steps in different regions is referred to as the multirate methods. In our cooperation with university of Montpellier, a novel multirate method using cell agglomeration has been designed and developed in our AIRONUM CFD platform. A series of large-scale test cases show that the new method is much more efficient than an explicit method, while retaining a similar time accuracy over the whole computational domain. The comparison with an implicit scheme shows that the implicit scheme is in some cases one order less accurate due to higher time steps and higher dissipation. A communication has been presented at ECCOMAS [17] and an article is submitted to a journal.

6.6. Application of AD to uncertainties and errors in CFD

Participants: Valérie Pascual, Laurent Hascoët, Alain Dervieux.

An important application of AD is the creation of uncertainty management tools, as first and second derivatives are used for the assembly of perturbation-based models for Uncertainty Quantification.

During the FP7 project UMRIDA, finished in september 2016, Inria has assisted Alenia-Aermacchi and WUT (Warsaw) in applying Tapenade to a CFD software for perturbation-based models.

We contributed the following chapters to the UMRIDA monography [24]:

- II.5.0 Introduction to Intrusive Perturbation Methods
- II.5.1 Algorithmic Differentiation for second derivatives
- III.a.4 Introduction to Intrusive Perturbation Methods and their range of applicability
- IV.3 Use of Automatic Differentiation tools at the example of Tapenade

6.7. Control of approximation errors

Participants: Gautier Brèthes, Éléonore Gauci, Alain Dervieux, Adrien Loseille [Gamma3 team, Inria-Rocquencourt], Frédéric Alauzet [Gamma3 team, Inria-Rocquencourt], Loïc Frazza [Gamma3 team, Inria-Saclay], Stephen Wornom, Anca Belme [university of Paris 6].

Reducing approximation errors as much as possible is a particular kind of optimal control problem. We formulate it exactly this way when we look for the optimal metric of the mesh, which minimizes a user-specified functional (goal-oriented mesh adaptation). In that case, the usual methods of optimal control apply, using adjoint states that can be produced by Algorithmic Differentiation.

Our theoretical studies in mesh adaptation are supported by the ANR project MAIDESC coordinated by ECUADOR and Gamma3, which deals with meshes for interfaces, third-order accuracy, meshes for boundary layers, and curved meshes.

The thesis of Éléonore Gauci on the goal-oriented criteria for CFD and coupled CSM-CFD systems is continuing. Éléonore Gauci gave a presentation at ECCOMAS in Crete.

Further studies of mesh adaptation for viscous flows are currently performed and a paper in collaboration with Gamma3 and university of Paris 6 (Anca Belme) is being written for a Journal.

An important novelty in mesh adaption is the norm-oriented AA method. The method relies on the definition of ad hoc correctors. It has been developed in the academic platform “FMG” for elliptic problems. Gautier Brèthes gave several presentations in conferences, a journal article has been published [12]. The introduction of the norm-oriented idea considerably amplifies the impact of adjoint-based AA. The applied mathematician and the engineer now have methods when faced to mesh adaptation for the simulation of a complex PDE system, since they can specify which error norm level they wish, and for which norm. Another version is developed jointly with Inria team Gamma3 for the compressible Euler model.

A work of extension of a different standpoint, the tensorial metric method was started during the thesis of Gautier Brèthes and has been submitted to a journal.

CFD application are supported by the European FP7 project UMRIDA which deals with the application of AA to approximation error modelling and control.

This involves an extensive work on a series of RANS (Reynolds Averaged Navier-Stokes) adaptive computations relying on the multi-scale method on the one hand, and on the other hand on further development by Gamma3 and Ecuador of the novel norm-oriented method for the compressible Euler model. This will be first published as a chapter contributed to the UMRIDA monography [24]: II.1.4 Numerical uncertainties estimation and mitigation by mesh adaptation Frédéric Alauzet, Alain Dervieux, Loïc Frazza and Adrien Loseille.

6.8. Turbulence models

Participants: Alain Dervieux, Bruno Koobus, Emmanuelle Itam, Marianna Braza [CNRS-IMFT at Toulouse], Stephen Wornom, Bruno Sainte-Rose [Lemma].

Modeling turbulence is an essential aspect of CFD. The purpose of our work in hybrid RANS/LES (Reynolds Averaged Navier-Stokes / Large Eddy Simulation) is to develop new approaches for industrial applications of LES-based analyses. In the applications targeted (aeronautics, hydraulics), the Reynolds number can be as high as several tenth millions, far too high for pure LES models. However, certain regions in the flow can be better predicted with LES than with usual statistical RANS (Reynolds averaged Navier-Stokes) models. These are mainly vortical separated regions as assumed in one of the most popular hybrid model, the hybrid Detached Eddy Simulation model. Here, “hybrid” means that a blending is applied between LES and RANS. An important difference between a real life flow and a wind tunnel or basin is that the turbulence of the flow upstream of each body is not well known.

This year, we have validated and experimented for various test cases the integration of the boundary layer by adding the so-called Menter correction imposing the Bradshaw law. We have studied these improvements on multiple-body flows. An emblematic case is the interaction between two parallel cylinders, one being in the wake of the other.

The development of hybrid models, in particular DES in the literature has raised the question of the domain of validity of these models. According to theory, these models should not be applied to flow involving laminar boundary layers (BL). But industrial flows are complex flows and often present regions of laminar BL, regions of fully developed turbulent BL and regions of non-equilibrium vortical BL. It is then mandatory for industrial use that the new hybrid models give a reasonable prediction for all these types of flow. This year, we concentrated on evaluating the behavior of hybrid models for laminar BL and for vortical wakes. While less predictive than pure LES on laminar BL, some hybrid models still give reasonable predictions for rather low Reynolds numbers. A little surprisingly, the prediction of vortical wakes needs some improvement. For this improvement, we propose a hybrid formulation involving locally a sophisticated LES-VMS (Large Eddy Simulation - Variational Multi-Scale) model combined with the dynamic local limitation of Germano-Piomelli. Several standard options together with the new model have been compared for a series of test cases: a communication has been presented in a conference [18] and an article is in preparation.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

- Ecuador and Lemma share the results of Gautier Brèthes’ thesis, which is partly supported by Lemma, the other part being supported by a PACA region fellowship.
- Ecuador and Lemma have a bilateral contract to share the results of Stephen Wornom, Lemma engineer provided to Inria and hosted by Inria under a Inria-Lemma contract.

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

8.1.1.1. MAIDESC

Ecuador is coordinator of the ANR project MAIDESC, with Inria team Gamma3, University of Montpellier II, CEMEF-Ecole des Mines, Inria-Bordeaux, Lemma and Transvalor. MAIDESC concentrates on mesh adaptation and in particular meshes for interfaces, third-order accuracy, meshes for boundary layers, and curved meshes.

8.2. European Initiatives

8.2.1. FP7 & H2020 Projects

8.2.1.1. AboutFlow

Type: PEOPLE

Instrument: Initial Training Network

Duration: 2012-2016

Coordinator: Jens-Dominik Mueller

Partner: Queen Mary University of London (UK)

Inria contact: Laurent Hascoët

Abstract: The aim of AboutFlow is to develop robust gradient-based optimisation methods using adjoint sensitivities for numerical optimisation of flows. <http://aboutflow.sems.qmul.ac.uk/>

8.2.1.2. UMRIDA

Type:AAT

Instrument:Aeronautics and Air Transport

Duration: 2013-2016

Coordinator: Charles Hirsch

Partner: Numeca S.A. (Belgium)

Inria contact: Alain Dervieux

Abstract: UMRIDA addresses major research challenges in Uncertainty Quantification and Robust Design: develop new methods that handle large numbers of simultaneous uncertainties and generalized geometrical uncertainties. Apply these methods to representative industrial configurations.

8.3. International Initiatives

8.3.1. Inria International Labs

Ecuador participates in the Joint Laboratory for Exascale Computing (JLESC) together with colleagues at Argonne National Laboratory. Laurent Hascoët attended the JLESC meeting in Lyon, France, June 27-29.

8.4. International Research Visitors

8.4.1. Visits of International Scientists

- Krishna Narayanan from Argonne National Laboratory, june 29-july 1.

8.4.2. Internships

- Georgios Ntanakas from Rolls-Royce, Germany, january 18-30.
- Ala Taftaf to Rolls-Royce, Germany, may 6-27.

8.4.3. Visits to International Teams

- Laurent Hascoët visited Argonne National Laboratory, november 14-22.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. Member of the organizing committees

- Laurent Hascoët is on the organizing committee of the EuroAD Workshops on Algorithmic Differentiation.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master : Laurent Hascoët, Optimisation avancée, 15 h, M2, University of Nice

9.2.2. Supervision

PhD in progress : Ala Taftaf, “Extensions of Algorithmic Differentiation by Source Transformation to meet some needs of Scientific Computing”, started july 2013, advisor L. Hascoët.

PhD in progress : Éléonore Gauci, “Norm-oriented criteria for CFD and coupled CSM-CFD systems”, started october 2014, advisor A. Dervieux

9.2.3. Juries

- Alain Dervieux, jury, PhD defense of Laure Billon, Mines Paristech, december 9.
- Laurent Hascoët, jury, PhD defense of Vladimir Groza, University of Nice, november 9.

9.3. Popularization

Laurent Hascoët wrote an article about AD for the blog “binaire”, hosted by “Le Monde”. May 9.

10. Bibliography

Major publications by the team in recent years

- [1] F. COURTY, A. DERVIEUX. *Multilevel functional Preconditioning for shape optimisation*, in "International Journal of CFD", 2006, vol. 20, n^o 7, pp. 481-490

- [2] F. COURTY, A. DERVIEUX, B. KOOBUS, L. HASCOËT. *Reverse automatic differentiation for optimum design: from adjoint state assembly to gradient computation*, in "Optimization Methods and Software", 2003, vol. 18, n^o 5, pp. 615-627
- [3] B. DAUVERGNE, L. HASCOËT. *The Data-Flow Equations of Checkpointing in reverse Automatic Differentiation*, in "International Conference on Computational Science, ICCS 2006, Reading, UK", 2006
- [4] L. HASCOËT, M. ARAYA-POLO. *The Adjoint Data-Flow Analyses: Formalization, Properties, and Applications*, in "Automatic Differentiation: Applications, Theory, and Tools", H. M. BÜCKER, G. CORLISS, P. HOVLAND, U. NAUMANN, B. NORRIS (editors), Lecture Notes in Computational Science and Engineering, Springer, 2005
- [5] L. HASCOËT, S. FIDANOVA, C. HELD. *Adjoining Independent Computations*, in "Automatic Differentiation of Algorithms: From Simulation to Optimization", New York, NY, G. CORLISS, C. FAURE, A. GRIEWANK, L. HASCOËT, U. NAUMANN (editors), Computer and Information Science, Springer, 2001, chap. 35, pp. 299-304
- [6] L. HASCOËT. *Adjoint by Automatic Differentiation*, in "Advanced data assimilation for geosciences", Oxford University Press, 2014, <https://hal.inria.fr/hal-01109881>
- [7] L. HASCOËT, U. NAUMANN, V. PASCUAL. "To Be Recorded" *Analysis in Reverse-Mode Automatic Differentiation*, in "Future Generation Computer Systems", 2004, vol. 21, n^o 8
- [8] L. HASCOËT, J. UTKE, U. NAUMANN. *Cheaper Adjoint by Reversing Address Computations*, in "Scientific Programming", 2008, vol. 16, n^o 1, pp. 81-92
- [9] L. HASCOËT, M. VÁZQUEZ, B. KOOBUS, A. DERVIEUX. *A Framework for Adjoint-based Shape Design and Error Control*, in "Computational Fluid Dynamics Journal", 2008, vol. 16, n^o 4, pp. 454-464
- [10] L. HASCOËT, V. PASCUAL. *The Tapenade Automatic Differentiation tool: Principles, Model, and Specification*, in "ACM Transactions On Mathematical Software", 2013, vol. 39, n^o 3, <http://dx.doi.org/10.1145/2450153.2450158>
- [11] M. VÁZQUEZ, A. DERVIEUX, B. KOOBUS. *Multilevel optimization of a supersonic aircraft*, in "Finite Elements in Analysis and Design", 2004, vol. 40, pp. 2101-2124

Publications of the year

Articles in International Peer-Reviewed Journals

- [12] G. BRËTHES, A. DERVIEUX. *Anisotropic norm-oriented mesh adaptation for a Poisson problem*, in "Journal of Computational Physics", 2016, vol. 322, pp. 804 - 826 [DOI : 10.1016/J.JCP.2016.07.008], <https://hal.inria.fr/hal-01413431>
- [13] D. GOLDBERG, S. H. K. NARAYANAN, L. HASCOËT, J. UTKE. *An optimized treatment for algorithmic differentiation of an important glaciological fixed-point problem*, in "Geoscientific Model Development", 2016, vol. 9, n^o 5, 27 p. , <https://hal.inria.fr/hal-01413295>

- [14] L. HASCOËT, J. UTKE. *Programming language features, usage patterns, and the efficiency of generated adjoint code*, in "Optimization Methods and Software", 2016, vol. 31, pp. 885 - 903 [DOI : 10.1080/10556788.2016.1146269], <https://hal.inria.fr/hal-01413332>
- [15] J. C. HUECKELHEIM, L. HASCOËT, J.-D. MÜLLER. *Algorithmic differentiation of code with multiple context-specific activities*, in "ACM Transactions on Mathematical Software", 2016, <https://hal.inria.fr/hal-01413321>

International Conferences with Proceedings

- [16] L. HASCOËT, M. MORLIGHEM. *Source-transformation Differentiation of a C++-like Ice Sheet model*, in "AD2016 - 7th International Conference on Algorithmic Differentiation", Oxford, United Kingdom, September 2016, 4 p. , <https://hal.inria.fr/hal-01413381>
- [17] E. ITAM, S. WORNOM, B. KOOBUS, A. DERVIEUX. *A Volume-Agglomeration Multirate Time Advancing Approach*, in "ECCOMAS Congress 2016 VII European Congress on Computational Methods in Applied Sciences and Engineering", Hernalssios, Crete, Greece, June 2016, <https://hal.inria.fr/hal-01413449>
- [18] E. ITAM, S. WORNOM, B. KOOBUS, A. DERVIEUX. *Hybrid simulation of high-Reynolds number flows relying on a variational multiscale model*, in "6th Symposium on Hybrid RANS-LES Methods", Strasbourg, France, September 2016, <https://hal.inria.fr/hal-01413453>
- [19] S. H. K. NARAYANAN, L. HASCOËT. *Interfacing OpenAD and Tapenade*, in "AD2016 - 7th International Conference on Algorithmic Differentiation", Oxford, United Kingdom, September 2016, 4 p. , <https://hal.inria.fr/hal-01413363>
- [20] A. TAFTAF, L. HASCOËT. *On the correct application of AD checkpointing to adjoint MPI-parallel programs*, in "VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016)", Crete, Greece, VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016), June 2016, <https://hal.inria.fr/hal-01413394>

Conferences without Proceedings

- [21] V. PASCUAL, L. HASCOËT. *Mixed-language Automatic Differentiation*, in "AD2016 - 7th International Conference on Algorithmic Differentiation", Oxford, United Kingdom, September 2016, <https://hal.archives-ouvertes.fr/hal-01393424>
- [22] A. TAFTAF, L. HASCOËT. *Experiments on Checkpointing Adjoint MPI Programs*, in "11th ASMO UK/ISSMO/NOED2016: International Conference on Numerical Optimisation Methods for Engineering Design", Munich, Germany, July 2016, <https://hal.inria.fr/hal-01413402>

Research Reports

- [23] L. HASCOËT, A. TAFTAF. *On the correct application of AD checkpointing to adjoint MPI-parallel programs*, Inria Sophia Antipolis, February 2016, n^o RR-8864, <https://hal.inria.fr/hal-01277449>

References in notes

-
- [24] C. HIRSCH, G. BUGEDA, J. SZUMBARSKI (editors). *UMRIDA: Uncertainty Management for Robust Industrial Design in Aeronautics including Best Practice Guide on Uncertainty Quantification Applications*, Springer, 2017, to appear
- [25] A. AHO, R. SETHI, J. ULLMAN. *Compilers: Principles, Techniques and Tools*, Addison-Wesley, 1986
- [26] I. ATTALI, V. PASCUAL, C. ROUDET. *A language and an integrated environment for program transformations*, Inria, 1997, n^o 3313, <http://hal.inria.fr/inria-00073376>
- [27] B. CHRISTIANSON. *Reverse accumulation and implicit functions*, in "Optimization Methods and Software", 1998, vol. 9, n^o 4, pp. 307–322
- [28] D. CLÉMENT, J. DESPEYROUX, L. HASCOËT, G. KAHN. *Natural semantics on the computer*, in "Proceedings, France-Japan AI and CS Symposium, ICOT", 1986, pp. 49-89, Also, Information Processing Society of Japan, Technical Memorandum PL-86-6. Also Inria research report # 416, <http://hal.inria.fr/inria-00076140>
- [29] P. COUSOT. *Abstract Interpretation*, in "ACM Computing Surveys", 1996, vol. 28, n^o 1, pp. 324-328
- [30] B. CREUSILLET, F. IRIGOIN. *Interprocedural Array Region Analyses*, in "International Journal of Parallel Programming", 1996, vol. 24, n^o 6, pp. 513–546
- [31] J. GILBERT. *Automatic differentiation and iterative processes*, in "Optimization Methods and Software", 1992, vol. 1, pp. 13–21
- [32] M.-B. GILES. *Adjoint methods for aeronautical design*, in "Proceedings of the ECCOMAS CFD Conference", 2001
- [33] A. GRIEWANK, C. FAURE. *Reduced Gradients and Hessians from Fixed Point Iteration for State Equations*, in "Numerical Algorithms", 2002, vol. 30(2), pp. 113–139
- [34] A. GRIEWANK, A. WALTHER. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, 2nd, SIAM, Other Titles in Applied Mathematics, 2008
- [35] L. HASCOËT. *Transformations automatiques de spécifications sémantiques: application: Un vérificateur de types incremental*, Université de Nice Sophia-Antipolis, 1987
- [36] P. HOVLAND, B. MOHAMMADI, C. BISCHOF. *Automatic Differentiation of Navier-Stokes computations*, Argonne National Laboratory, 1997, n^o MCS-P687-0997
- [37] F.-X. LE DIMET, O. TALAGRAND. *Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects*, in "Tellus", 1986, vol. 38A, pp. 97-110
- [38] G. MADEC, P. DELECLUSE, M. IMBARD, C. LEVY. *OPA8.1 ocean general circulation model reference manual*, Pole de Modelisation, IPSL, 1998
- [39] B. MOHAMMADI. *Practical application to fluid flows of automatic differentiation for design problems*, in "Von Karman Lecture Series", 1997

- [40] N. ROSTAING. *Différentiation Automatique: application à un problème d'optimisation en météorologie*, université de Nice Sophia-Antipolis, 1993

- [41] R. RUGINA, M. RINARD. *Symbolic Bounds Analysis of Pointers, Array Indices, and Accessed Memory Regions*, in "Proceedings of the ACM SIGPLAN'00 Conference on Programming Language Design and Implementation", ACM, 2000