A review of adjoint methods for sensitivity analysis, uncertainty quantification and optimization in numerical codes

Grégoire Allaire

A review of adjoint methods for sensitivity analysis, uncertainty quantification and optimization in numerical codes

G. Allaire[1]

1: CMAP, Ecole Polytechnique, 91128 Palaiseau, France

Abstract: The goal of this paper is to briefly recall the importance of the adjoint method in many problems of sensitivity analysis, uncertainty quantification and optimization when the model is a differential equation. We illustrate this notion with some recent examples. As is well known, from a computational point of view the adjoint method is intrusive, meaning that it requires some changes in the numerical codes. Therefore we advocate that any new software development must take into account this issue, right from its inception.

Keywords: adjoint method, sensitivity, uncertainty, optimization.

1. Introduction

The adjoint method originates in the theory of Lagrange multipliers in optimization. As an example, consider the minimization of a function J from R^n into R, under the equality constraint F, a function from R^n into R^m,

$$\min J(x) \quad \text{such that} \quad F(x) = 0 \qquad (1)$$

Introducing the Lagrange multilplier p in R^m and the Lagrangian

$$L(x,p) = J(x) + p.F(x) \qquad (2)$$

the optimality condition for (1) (under some qualification conditions) is the stationarity of the Lagragian, namely

$$J'(x^*) + p^*.F'(x^*) = 0 \qquad (3)$$

The adjoint method is an extension of this approach in the framework of optimal control theory. In this context, the variable x is the union of a state variable y and a control variable u, while the constraint F(y,u)=0 is the state equation which gives y in terms of u. In such a case the Lagrange multiplier p is called the adjoint state. This approach was developed by Pontryagin [20] for ordinary differential equations and by Lions [17] for distributed systems, i.e., partial differential equations.

2. Adjoint method

2.1 Setting of the problem

For simplicity in the exposition we consider a finite dimensional model problem. Denote by u in R^k a control or a parameter which is the ultimate optimization variable. The state of the system is denoted by y in R^n and is defined as the solution of the following state equation

$$A(u)\, y = b \qquad (4)$$

where b is a given right hand side in R^n and A(u) is an invertible n by n matrix. Since the matrix depends on u, so does the solution y. The goal is to minimize, over all admissible controls u, an objective function J(u,y) under the constraint (4). The difficulty of the problem is that y depends nonlinearly on u. We assume here that u is a continuous variable (the case of discrete variables is much more subtle). To numerically minimize the objective function J, the most efficient algorithms are by far those based on derivative informations. Therefore, a key issue is to compute the gradient of J(u,y(u)). This is the purpose of the next section.

2.2 Computation of a gradient

By the chain rule lemma the gradient of j(u)=J(u,y(u)) is

$$j'(u) = \partial_u J(u,y(u)) + y'(u)\, \partial_y J(u,y(u)) \qquad (5)$$

Differentiating (4) with respect to u yields

$$A(u)\, y'(u) = -A'(u)\, y(u) \qquad (6)$$

From (6) and (5) we deduce the so-called direct gradient

$$j'(u) = \partial_u J(u,y(u)) - A(u)^{-1}A'(u)\, y(u)\, \partial_y J(u,y(u)) \qquad (7)$$

If the dimension k of the vector variable u is small, formula (7) is viable in the sense that it is not too costly for computing j'(u). Indeed, using (7) requires k linear solving of (6) (recall that the notation ' is the gradient with respect to u, a vector in R^k). Solving (6) is a costly operation since A(u) is an n by n matrix and n is usually very large (it is the number of degrees of freedom in the state equation, ranging from 10^4 up to 10^7 in the most recent applications). However, if k is as large as n (like in structural optimization, parameter identification, data assimilation, etc.), then formula (7) is useless !

Rather one should use the adjoint method which goes as follows. Introduce the so-called adjoint equation

$$A^T(u)\, p = -\partial_y J(u,y(u)) \qquad (8)$$

where $A^T(u)$ is the adjoint or transposed matrix. Multiplying (8) by y'(u) leads to

$$A^T(u)p.y'(u) = -y'(u).\partial_y J(u,y(u)) \qquad (9)$$

while multiplying (6) by p gives

$$A(u)y'(u).p = -A'(u)y(u).p \qquad (10)$$

Comparing (9) and (10) which have equal left hand side yields to a simplification of (5) as

$$j'(u) = \partial_u J(u,y(u)) + A'(u)y(u).p \qquad (11)$$

On the contrary of (7), formula (11) is cheap to evaluate, whatever the size of k: it requires only to solve the additional adjoint equation (8). It largely outperforms (7) and it is thus the method of choice for computing the gradient of the objective function j(u)=J(u,y(u)). There are at least two ways for introducing an adjoint in a computer code. Either, a so-called analytic adjoint, i.e. (8), is implemented. Or a program for the adjoint is obtained by automatic differentiation [14], [23] of the code solving the state equation (4).

The only drawback of (11) (and of the whole adjoint approach) is that it is intrusive from a computational software point of view, namely it requires the development of an adjoint solver. This is quite easy if you know well your software. It is impossible if your software is a black box with no access to the source code. Even, if the source files are available, it could be a nightmare if they are not documented and/or not written in a clear and systematic manner...

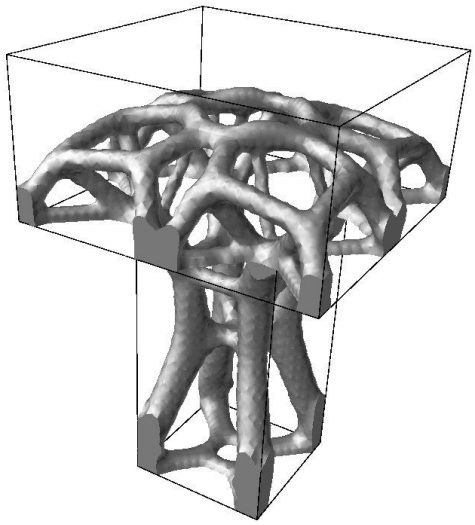

Figure 1: Optimal mast obtained with the level set method for shape and topology optimization

3. Applications

There are many applications of the adjoint approach. The most classical ones are in control theory [17], [20], optimization [1], sensitivity or perturbation analysis, uncertainty quantification, inverse problems [22] and data assimilation [15]. I will review some of these applications, in connection with my own experience, therefore not trying to be exhaustive. The goal of this list is to convince the reader that, whatever the field of application, any numerical simulation code will sooner or later need an adjoint. Therefore, it is crucial to design any new software with this view in mind.

3.1 Optimal design

A classical application of the adjoint approach is the optimal design of structures or systems. The number of optimization variables describing the system is so large that it is the only viable approach. For example in fluid mechanics, the number of variables is related to the number of parameters, necessary to describe the boundary of an airplane [13], [18]. In the context of topology optimization for mechanical structures, the number of design variables is even larger since any cell of a “hold-all” computational domain is potentially a design variable: either it is full of material or empty. For any topology optimization method (be it homogenization [4], SIMP [10], phase field [24] or level set [5]) the number of design variables is so large that again the adjoint method is the only way for computing a gradient and perform efficient optimization. Topology optimization (using an adjoint approach) has been very successful those last years and we present two results obtained with the level set method [5], [6] in Figures 1 and 2.

Figure 2: Optimal chair obtained with the level set method and an exact mesh of the structure

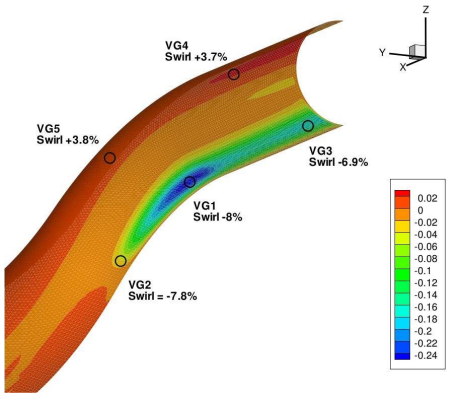

Figure 3: Topological derivative or sensitivity for adding a vortex generator in an air duct

3.2 Perturbation analysis

The adjoint approach is also very helpful for sensitivity analysis, i.e., in the determination of the effect of small perturbations of the data on the solution. This is a very old and classical topic in nuclear reactor physics [19] where the adjoint state is called the importance function. It is a classical tool in meteorology too [11].

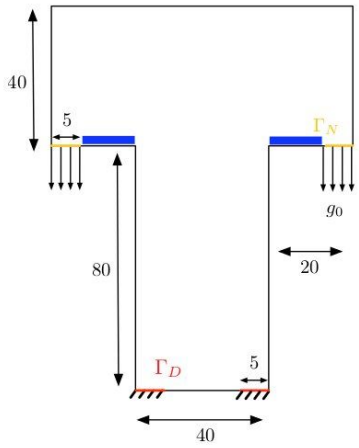

Figure 4: Boundary conditions for the mast test case

Compared to the setting of section 2, the variable u is now interpreted as a coefficient or a data in the model (4) and j(u)=J(u,y(u)) is a quantity of interest, the sensitivity of which has to be computed. Formula (11) is precisely the prediction of the sensitivity of the output j(u) with respect to perturbations of the data u.

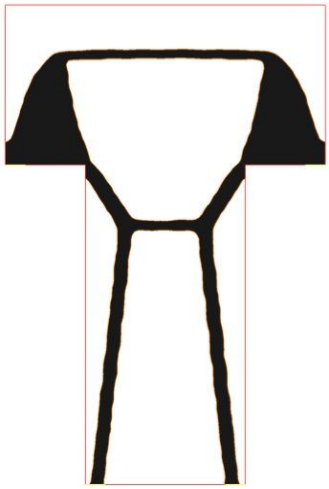

Figure 5: Optimal mast without uncertainties

A variant of this perturbation analysis appeared recently under the name of topological derivative [12], [21]. It amounts to compute a criterion which predicts if the nucleation of a small hole or inclusion is going to decrease or not the objective function. It has been very successful, as an additional topology optimization tool, in structural mechanics [3], inverse problems [8] and fluid mechanics. In figure 3 an example from [7] is displayed: the blue (negative) region indicates where it is advantageous to put a vortex generator (a small winglet) on the boundary of an air duct in order to obtain a homogeneous flow distribution at the outlet.

3.3 Uncertainty quantification

Estimating the variations of a computational results in terms of possible uncertainties of the data or parameters is a very important topic which has attracted a lot of attention those last years. Very often, this task is associated to some probabilistic representation of the parameters and of the results. Besides the direct approach of Monte-Carlo simulation (which is very CPU time-consuming) one can use spectral methods or polynomial chaos [16]. Still these methods are quite intensive in terms of CPU. Another possibility is to rely on a worst-case approach which makes the problem deterministic if the worst-case scenario can be identified (which is not always possible). Recently a linearization approach for small uncertainties was suggested in [2] and [9] which makes the problem fully deterministic and explicit. Here is the main idea. Suppose that small perturbations δ occur in the right hand side of (4), namely

$$A(u)\, y(u,\delta) = b + \delta \qquad (12)$$

The perturbed solution is $y(u,\delta) = y(u) + z$ with $z=A(u)^{-1}\delta$. Let us assume that the quantity of interest is

$$j(u,\delta) = J(u,y(u,\delta)) \qquad (13)$$

Assuming that the norm of the perturbation is bounded by a small number m>0, the worst-case function is

$$w(u) = \max_{\|\delta\| < m} j(u,\delta) \qquad (14)$$

Evaluating (14) is a difficult task in general. However, if m is small we can linearize (13) and compute (14) as

$$w(u) \approx j(u,0) + \max_{\|\delta\| < m} \partial_y J(u,y(u,0)) \,.\, z \qquad (15)$$

which yields, since $z=A(u)^{-1}\delta$,

$$w(u) \approx j(u) + m \,\|\, A(u)^{-T} \partial_y J(u,y(u)) \,\| \qquad (16)$$

As usual inverting the matrix $A^T(u)$ is too costly, so we resort to the adjoint approach again. Multiplying the adjoint equation (8) by z and comparing with the equation A(u)z=δ multiplied by p leads to

$$-\partial_y J(u,y(u,0)) \,.\, z = \delta \,.\, p \qquad (17)$$

which in turn gives the worst-case function

$$w(u) \approx j(u) + m \,\|\, p \,\| \qquad (18)$$

In other words the worst case deviates from the standard ouput by the norm of the adjoint p multiplied by the magnitude m of the uncertainties. This is a very simple (albeit linearized) way of assessing and quantifying uncertainties which, again, relies on the notion of adjoint.

In [2] we used formula (18) to perform shape and topology optimization of mechanical structures under various type of perturbations (on loads, material

properties or geometry). In Figures 4, 5 and 6 are displayed boundary conditions and optimal shapes for minimal compliance under a volume constraint when uncertain vertical loads are applied in the blue zone. As can be seen here, these perturbations can have a drastic effect on the optimal topology.

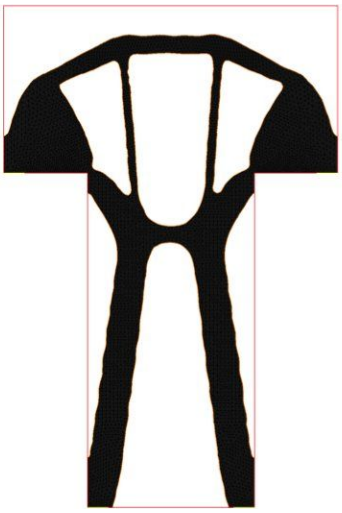

Figure 6: Optimal mast with uncertainties

4. Conclusion

In this paper we briefly recalled the adjoint theory for computing derivatives or sensitivities of an objective function depending on the solution of a state equation. In an increasingly large variety of cases (as illustrated in the figures above), the number of variables is so large that there is no viable alternative to the adjoint method. We also collected various problems where the adjoint is a key technique: of course optimal design, control and inverse problems, but also sensitivity analysis and uncertainty quantification. These types of problems occur in virtually all fields of science and engineering. Despite its efficiency the adjoint approach suffers from one major drawback which is the difficulty of implementing an adjoint solver (be it analytic or by automatic differentiation) in a pre-existing code which was not designed for accommodating an adjoint solver. Note however that programming an adjoint along the early development of a numerical software is a rather easy task. Therefore we advocate that any new software planning should incorporate, in its early stages, all the necessary tools for later incorporating an adjoint (when the need will be felt and, surely, it will happen at some times). These ingredients include various data structures, the possibility of defining “dummy” loads, similar to the right hand sides of the adjoint equation (8), the definition of the linearized adjoint operator (if an analytic adjoint is going to be chosen) or the “clean” programming of the routines that will be submitted to an automatic differentiation tool like Tapenade [23]. These minor precautions will later save a huge amount of development time, not to speak of the even larger savings in computational efficiency compared to any other method, either not relying on gradients or computing mere approximations (like finite differences).

5. Acknowledgement

Part of this work was performed in the framework of the RODIN project (sponsored by FUI AAP 13). This work was done in collaboration with many colleagues, among them M. Albertelli, Ch. Dapogny, G. Delgado, P. Frey, F. Jouve, G. Michailidis. The author is also a member of the DEFI project at INRIA Saclay Ile-de-France.

6. References

[1] G. Allaire: "*Conception optimale de structures*", Collection: Mathématiques et Applications, Vol. 58, Springer, 2007.

[2] G. Allaire, Ch. Dapogny: "*A linearized approach to worst-case design in parametric and geometric shape optimization*", M3AS, 24(11) pp.2199-2257, 2014.

[3] G. Allaire, F. de Gournay, F. Jouve, A.-M. Toader: "*Structural optimization using topological and shape sensitivity via a level set method*", Control and Cybernetics 34, 59-80, 2005.

[4] G. Allaire: "*Shape optimization by the homogenization method*", Springer, 2001.

[5] G. Allaire, F. Jouve, A.-M. Toader: "*Structural optimization using sensitivity analysis and a level-set method*", J. Comp. Phys. Vol 194/1, 363-393, 2004.

[6] G. Allaire, Ch. Dapogny, P. Frey: "*Shape optimization with a level set based mesh evolution method*", CMAME 282, 22-53, 2014.

[7] G. Allaire, J. Chetboun: "*Flow Control of Curved Air Ducts using Topological Derivatives*", Proceedings of the 8th World Congress on Structural and Multidisciplinary Optimization, H.C. Rodrigues et al. eds., ISSMO, 2009.

[8] H. Ammari: *"An introduction to mathematics of emerging biomedical imaging"*, Collection: Mathématiques et Applications, Vol. 62, Springer, 2008.

[9] I. Babuška, F. Nobile and R. Tempone: "*Worst case scenario analysis for elliptic problems with uncertainty*", Numer. Math., 101(2), pp.185-219, 2005.

[10] M. Bendsoe, O. Sigmund: "*Topology Optimization. Theory, Methods, and Applications*", Springer Verlag, 2003.

[11] R. Errico: *"What is an adjoint model ?"*, Bulletin of the American Meteorological Society, 78, 2577-2591, 1997.

[12] S. Garreau, P. Guillaume, M. Masmoudi: "*The topological asymptotic for PDE systems: the*

elasticity case", SIAM J. Control Optim., 39(6), 1756-1778, 2001.

[13] M. Giles, N. Pierce: "*An Introduction to the Adjoint Approach to Design*", Flow, Turbulence and Combustion 65: 393-415, 2000.

[14] A. Griewank, A. Walther: *"Evaluating derivatives. Principles and techniques of algorithmic differentiation"*, Second edition, Society for Industrial and Applied Mathematics (SIAM), 2008.

[15] F.X. Le Dimet, O. Talagrand: "*Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects*", Tellus A, 38A(2), 97-110, 1986.

[16] O. Le Maître, O. Knio: "*Spectral methods for uncertainty quantification. With applications to computational fluid dynamics*", Scientific computation, Springer, New York, 2010.

[17] J.L. Lions: *"Optimal control of systems governed by partial differential equations",* Die Grundlehren der mathematischen Wissenschaften, 170, Springer-Verlag, 1971.

[18] B. Mohammadi, O. Pironneau: "*Applied shape optimization for fluids*", Clarendon Press, Oxford, 2001.

[19] J. Planchard: "*Méthodes mathématiques en neutronique*", Collection de la Direction des Etudes et Recherches d'EDF, Eyrolles, 1995.

[20] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze: *"The mathematical theory of optimal processes"*, Pergamon Press, Oxford, 1964.

[21] J. Sokolowski, A. Zochowski: "*On the topological derivative in shape optimization*", SIAM J. Control Optim. 37, 1251--1272 , 1999.

[22] A. Tikhonov, V. Arsenin: "*Solutions of ill-posed problems*", John Wiley, 1977.

[23] L. Hascoët, V. Pascual: "*The Tapenade automatic differentiation tool: principles, model, and specification*", ACM Transactions On Mathematical Software, 39(3), 2013.

[24] A. Chambolle, B. Bourdin: "*Design-dependent loads in topology optimization*", ESAIM Control Optim. Calc. Var., 9, 19-48, 2003.