

Variational methods

Maelle Nodet, Arthur Vidard

▶ **To cite this version:**

Maelle Nodet, Arthur Vidard. Variational methods. Handbook of Uncertainty Quantification, Springer International Publishing, pp.1-20, 2016, 978-3-319-11259-6. 10.1007/978-3-319-11259-6_32-1 . hal-01251720

HAL Id: hal-01251720
https://inria.hal.science/hal-01251720

Submitted on 6 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Title: Variational methods

Name: Maelle Nodet[1,2,3], Arthur Vidard[3,1,2]

Affil./Addr. 1: Univ. Grenoble Alpes

LJK, F-38000 Grenoble, France

E-mail: maelle.nodet@inria.fr, arthur.vidard@inria.fr

Affil./Addr. 2: CNRS

LJK, F-38000 Grenoble, France

Affil./Addr. 3: Inria

Variational methods

Keywords

variational sensitivity analysis, variational methods, tangent model, adjoint model, gradient, automatic differentiation, derivative, local sensitivity analysis, stability analysis, geophysical applications, meteorology, glaciology, oceanography

Abstract

This contribution presents derivative-based methods for local sensitivity analysis, called *Variational Sensitivity Analysis* (VSA). If one defines an output called the *response function*, its sensitivity to inputs variations around a nominal value can be studied using derivative (gradient) information. The main issue of VSA is then to provide an efficient way of computing gradients.

This contribution first presents the theoretical grounds of VSA: framework and problem statement, tangent and adjoint methods. Then it covers pratical means to compute derivatives, from naive to more sophisticated approaches, discussing their various

merits. Finally, applications of VSA are reviewed and some examples are presented, covering various applications fields: oceanography, glaciology, meteorology.

Introduction

This contribution presents derivative-based methods for local sensitivity analysis, gathered under the name *Variational Sensitivity Analysis* (VSA). The aim of VSA is to provide sensitivity information using derivatives. This is indeed a valuable information, as the derivative of a function at a given point gives the growth rate at that point, in other words the tendency of the function to grow (or not) when the input varies. Approximately one could say *the larger the derivative, the more sensitive the parameter*.

Note that VSA can be extended to global analysis (GSA) and the reader is refered to contribution (see Derivative-based Global sensitivity measure). Here the focus will be solely on local derivative-based sensitivity analysis. Contrary to GSA, LSA aims to compute sensitivities when the parameters vary *locally* around their nominal values, and not *globally* over a potentially large subset.

VSA is closely related to the research domain called *Data Assimilation*. This one consists in adjusting input parameters of a model so that the system state fits a given set of observations (data). Variational data assimilation translates this into an optimal control problem whose aim is to minimise the model-observation misfit. This minimisation is performed using descent methods and the gradient is computed using the so-called adjoint method, which is also at the core of VSA. Moreover, improving parameters and models through data assimilation assumes that the most sensitive parameters are known, which in turn creates the need to perform VSA beforehand.

This contribution is divided in two parts. First it will cover the methods of local VSA: the derivative is first defined, then the adjoint method is shown to provide a

powerful way to compute it, then a brief overview about practical derivatives computation is given, and finally stability analysis is mentioned, which is closely related to sensitivity analysis.

In a second part some applications are presented. VSA has been used in a wide range of domains: e.g. meteorology [1; 5; 8; 23; 33; 34], cyclones tracking [14; 22; 32], air quality [26; 25], oceanography [2; 30; 27; 31], surface hydrology [4], groundwater modelling [28], glaciology [13], agronomy [15], chemistry [24], ... Historically the adjoint method was first applied to numerical weather prediction, so that meteorology is a primary application domain, with many references on VSA. As in other geophysical domains, meteorological models are in general of very large dimension, so that GSA is mostly out of reach, which motivates the introduction of the adjoint method and VSA. This contribution chose to focus on a small number of example applications in geophysics.

Methods

Problem statement

In this section, the sensitivity of the output vector $\mathbf{y}$ with respect to the input vector $\mathbf{x}$ is considered. This output is often called the *response function*, since it represent the response of the system to variation on the input. In the variational framework, practitioners are generally interested in studying sensitivities of given numerical models $\mathcal{M}$ coming from various application domains (physics, geosciences, biology, ...), so that the output is a function of a state vector $\mathbf{u}(\mathbf{x};t) \in \mathbb{R}^p$, which depends on the input $\mathbf{x} \in \mathbb{R}^d$ and on time t and represents the state of a given system of interest:

$$\begin{cases} \dfrac{d\mathbf{u}}{dt} = \mathcal{M}(\mathbf{u};\mathbf{x}), \quad t \in [0,T] \\ \mathbf{u}(t=0) = \mathbf{u}_0(\mathbf{x}) \end{cases} \tag{1}$$

As $\mathbf{u}$ lies in $\mathbb{R}^p$, the model $\mathcal{M}$ is a (possibly non linear) operator from $\mathbb{R}^p \times \mathbb{R}^d$ to $\mathbb{R}^p$. In that case, the dependency between the input and the output reads:

$$\mathbf{y} = G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x})) \tag{2}$$

where G is the response function and $\mathcal{G}$ maps the state space into the output space.

Remark 1. In the definition of the response function G lies all the art of sensitivity analysis. It should be designed carefully depending on the aim of the study (model validation, physical phenomenon study, etc.). This is discussed in depth in the contributions Sensitivity analysis of spatial and/or temporal phenomena and Variables Weights and Importance in Arithmetic Averages: Two Stories to Tell.

Remark 2. A more general case would be $\mathbf{y} = \mathcal{G}(\mathbf{u}(\mathbf{x}), \mathbf{x})$. The theory easily extends to that case, but for readability only this simpler case will be presented here.

In order to simplify the notations and clarify the reading, scalar outputs will be considered in this chapter, *i.e.* $\mathbf{y} \in \mathbb{R}$, but vector-valued outputs can of course be considered as well.

Variational sensitivity consists in finding the sensitivity of G with respect to the variations of $\mathbf{x}$, in other words the derivative of G with respect to the vector $\mathbf{x}$:

$$\frac{dG}{d\mathbf{x}}(\mathbf{x})$$

In this framework, G is differentiable from $\mathbb{R}^d$ to $\mathbb{R}$, with continuous derivative, *i.e.* G is of class C^1. Then the derivative can be identified (using the Euclidean scalar product in $\mathbb{R}^d$) with the gradient

$$\nabla_{\mathbf{x}} G(\mathbf{x}) = \left(\frac{\partial G}{\partial x_1}(\mathbf{x}), \frac{\partial G}{\partial x_2}(\mathbf{x}), ..., \frac{\partial G}{\partial x_d}(\mathbf{x}) \right)^T$$

The partial derivatives of G are particular cases of the directional derivative, also called the Gâteau derivative, which is defined by $G'(\mathbf{x})[\mathbf{h}]$ such that:

$$G'(\mathbf{x})[\mathbf{h}] = \lim_{\alpha \to 0} \frac{G(\mathbf{x} + \alpha \mathbf{h}) - G(\mathbf{x})}{\alpha}$$

where the direction $\mathbf{h}$ is a vector of $\mathbb{R}^d$. The partial derivative is simply the directional derivative in the direction of a basis vector, it is given by:

$$G'(\mathbf{x})[\mathbf{e}_i] = \frac{\partial G}{\partial x_i}(\mathbf{x})$$

As G is a continuously differentiable function, the link between the directional derivative and the gradient is immediate:

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}} G(\mathbf{x}).\mathbf{h}$$

where "." represents the Euclidean scalar product in $\mathbb{R}^d$.

Tangent and adjoint models

The most naive approach to track sensitive variables consists in fixing all parameters except one, increasing it by a given percentage (of its standard deviation or its absolute value) and then evaluate its impact on the output. This type of analysis can allow for a quick ranking of the variables if there are not too many of them. The reader can refer to [11] for a brief review on this subject.

A more refined approach would be to obtain a numerical approximation of the gradient using finite differences, *i.e.* computing the gradient as a limit of a growth rate (see [11] and next paragraph about practical aspects). This method is very simple but has two main drawbacks. First, its computational cost increases rapidly with the dimension d of $\mathbf{x}$. Second, the choice of δx_i is critical: if too large, the truncation error becomes large, if too small rounding error occurs.

To address this last point, one can obtain an exact calculation for the Gâteaux derivatives (FSAP, *Forward Sensitivity Analysis Procedure* in [3]'s terminology). Assuming the output is given by equations (1,2):

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{u}}\mathcal{G}(\mathbf{u}(\mathbf{x})).\mathbf{u}'(\mathbf{x})[\mathbf{h}] = \mathcal{G}'(\mathbf{u}(\mathbf{x}))[\mathbf{u}'(\mathbf{x})[\mathbf{h}]]$$

where $\mathbf{u}'(\mathbf{x})[\mathbf{h}]$ is the Gâteau derivative of $\mathbf{u}$ at $\mathbf{x}$ in the direction $\mathbf{h}$. If $\mathbf{v}$ denotes $\mathbf{u}'(\mathbf{x})[\mathbf{h}]$, then $\mathbf{v}$ is given by the following equations, called the Tangent Linear Model (TLM):

$$\begin{cases} \dfrac{d\mathbf{v}}{dt} = \dfrac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x}).\mathbf{v} + \dfrac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x}).\mathbf{h}, \quad t \in [0, T] \\ \mathbf{v}(t=0) = \mathbf{u}_0'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}}\mathbf{u}_0.\mathbf{h} \end{cases} \tag{3}$$

where $\frac{\partial \mathcal{M}}{\partial \mathbf{u}}$ and $\frac{\partial \mathcal{M}}{\partial \mathbf{x}}$ are the Jacobian matrices of the model with respect to the state $\mathbf{u}$ and the parameters $\mathbf{x}$. The Tangent Linear Model allows to compute exactly the directional derivative of the response function G, for a given direction $\mathbf{h}$. To get the entire gradient, all the partial derivatives need to be computed, therefore d integrations of the TLM are required. The accuracy problem may be solved, but for a large-dimensional set of parameters the computing cost of the FSAP method remains prohibitive.

In large-dimensional cases however, an adjoint method can be used to compute the gradient (ASAP, *Adjoint Sensitivity Analysis Procedure* in [3]'s terminology). As the derivation of the adjoint model is tedious in the general abstract case, this contribution will focus on common examples.

One will first consider the case where G and $\mathcal{G}$ are given by:

$$G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x})) = \int_{t=0}^{t=T} \mathcal{H}(\mathbf{u}(\mathbf{x}; t))\, dt \tag{4}$$

where $\mathcal{H}$ is the single time output function, from $\mathbb{R}^p$ to $\mathbb{R}$.

Then the Gâteau derivative of G with respect to $\mathbf{x}$ in the direction $\mathbf{h}$ is given by

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}}G(\mathbf{x}).\mathbf{h} = \int_{t=0}^{t=T} \nabla_{\mathbf{u}}\mathcal{H}(\mathbf{u}(\mathbf{x}; t)).\mathbf{v}\, dt \tag{5}$$

where $\mathbf{v}$ is as before $\mathbf{u}'(\mathbf{x})[\mathbf{h}]$.

Now the adjoint method consists in introducing a wisely chosen so-called *adjoint variable* so that the previous gradient can be formulated without the tangent variable

$\mathbf{v}$. To do so, the tangent model (3) is multiplied by a variable $\mathbf{p}(t)$ and an integration by parts is performed in order to obtain a formula $\int_{t=0}^{t=T}(...).\mathbf{v}\,dt$, which will later be identified with (5). So first a multiplication by $\mathbf{p}$ is performed and then an integration:

$$0 = -\int_0^T \mathbf{p}\frac{d\mathbf{v}}{dt}\,dt + \int_0^T \mathbf{p}\left(\frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x});\mathbf{x}).\mathbf{v} + \frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x});\mathbf{x}).\mathbf{h}\right)\,dt$$

Then the integration by parts gives:

$$\begin{aligned} 0 = &-\mathbf{p}(T)\mathbf{v}(T) + \mathbf{p}(0)\mathbf{v}(0) + \int_0^T \left(\frac{d\mathbf{p}}{dt} + \frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x});\mathbf{x})^T\mathbf{p}\right).\mathbf{v}\,dt \\ &+ \int_0^T \left(\left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x});\mathbf{x})\right)^T\mathbf{p}\right).\mathbf{h}\,dt \end{aligned}$$

Using the initial condition (3) on $\mathbf{v}$, it becomes:

$$\begin{aligned} &\mathbf{p}(T)\mathbf{v}(T) - \int_0^T \left(\frac{d\mathbf{p}}{dt} + \frac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x});\mathbf{x})^T\mathbf{p}\right).\mathbf{v}\,dt \\ &\qquad = \mathbf{p}(0)\nabla_{\mathbf{x}}\mathbf{u}_0.\mathbf{h} + \int_0^T \left(\left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x});\mathbf{x})\right)^T\mathbf{p}\right).\mathbf{h}\,dt \\ &\qquad = \left(\mathbf{p}(0)\nabla_{\mathbf{x}}\mathbf{u}_0 + \int_0^T \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x});\mathbf{x})\right)^T\mathbf{p}\,dt\right).\mathbf{h} \end{aligned} \tag{6}$$

As equation (5) needs to be re-written, the following backward equation for $\mathbf{p}$ is set and called the adjoint model:

$$\begin{cases} -\dfrac{d\mathbf{p}}{dt} = \left[\dfrac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x});\mathbf{x})\right]^T.\mathbf{p} + \nabla_{\mathbf{u}}\mathcal{H}(\mathbf{u}(\mathbf{x};t)), \quad t \in [0,T] \\ \mathbf{p}(t=T) = 0 \end{cases} \tag{7}$$

Combining (5), (6) and (7) the following formula for the Gâteau derivative is obtained:

$$G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}}G(\mathbf{x}).\mathbf{h} = \left(\mathbf{p}(0)\nabla_{\mathbf{x}}\mathbf{u}_0 + \int_0^T \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x});\mathbf{x})\right)^T\mathbf{p}\,dt\right).\mathbf{h}$$

leading to the gradient:

$$\nabla_{\mathbf{x}}G(\mathbf{x}) = \left[\nabla_{\mathbf{x}}\mathbf{u}_0\right]^T\mathbf{p}(t=0) + \int_{t=0}^{t=T}\left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x});\mathbf{x})\right]^T\mathbf{p}(t)\,dt \tag{8}$$

which does not involve variable $\mathbf{v}$ anymore. Thus the computing cost is independent from the number of parameters and $\nabla_{\mathbf{x}}G(\mathbf{x})$ can be computed *exactly* using one integration of the direct model and one backward integration of the adjoint model. Note that if the model is linear, only the adjoint integration is needed.

Similarly the Adjoint Model can be computed for the following type of output:

$$G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x}; t_1)) \tag{9}$$

for $t_1 \in]0; T[$, by noticing that it can be written as

$$G(\mathbf{x}) = \int_{t=0}^{t=T} \mathcal{H}(\mathbf{u}(\mathbf{x}; t))\delta_{t=t_1}(t)\, dt$$

where $\delta_{t=t_1}(t)$ is the Dirac function of t at point t_1. The Adjoint Model is then:

$$\begin{cases} -\dfrac{d\mathbf{p}}{dt} = \left[\dfrac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x})\right]^T .\mathbf{p} + \nabla_{\mathbf{u}}\mathcal{H}(\mathbf{u}(\mathbf{x}; t))\delta_{t=t_1}(t), \quad t \in [0, T] \\ \mathbf{p}(t = T) = 0 \end{cases} \tag{10}$$

Notice here that the adjoint variable is equal to zero up to time t_1. Computationally speaking, it is therefore sufficient to run the adjoint model from t_1 to 0..

Then the gradient is unchanged:

$$\nabla_{\mathbf{x}} G(\mathbf{x}) = \left[\nabla_{\mathbf{x}} \mathbf{u}_0\right]^T \mathbf{p}(t = 0) + \int_{t=0}^{t=T} \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x})\right]^T \mathbf{p}(t)\, dt \tag{11}$$

One can also be interested in an output at final time:

$$G(\mathbf{x}) = \mathcal{G}(\mathbf{u}(\mathbf{x}; T)), \qquad G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{u}}\mathcal{H}(\mathbf{u}(\mathbf{x}; T)).\mathbf{v}(T)$$

Then the adjoint model writes:

$$\begin{cases} -\dfrac{d\mathbf{p}}{dt} = \left[\dfrac{\partial \mathcal{M}}{\partial \mathbf{u}}(\mathbf{u}(\mathbf{x}); \mathbf{x})\right]^T .\mathbf{p}, \quad t \in [0, T] \\ \mathbf{p}(t = T) = \nabla_{\mathbf{u}}\mathcal{H}(\mathbf{u}(\mathbf{x}; T)) \end{cases} \tag{12}$$

and the gradient is unchanged:

$$\nabla_{\mathbf{x}} G(\mathbf{x}) = \left[\nabla_{\mathbf{x}} \mathbf{u}_0\right]^T \mathbf{p}(t = 0) + \int_{t=0}^{t=T} \left[\frac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{u}(\mathbf{x}); \mathbf{x})\right]^T \mathbf{p}(t)\, dt \tag{13}$$

Note here that this adjoint method allows to obtain directly every component of the gradient vector with a single run of the adjoint model, thus avoiding the curse of dimensionality on the parameter set dimension.

Practical gradient computation

Practical aspects are an important incentive for the choice of either of gradient computations methods presented above. For a small dimension problem, the finite difference approximation route is pretty straightforward and unless a high precision is required, it is probably the better choice. For larger dimension problems however either choice will require some efforts in term of computing cost and/or code developments. This section presents practical aspects of such developments, as well as a further approximation of the finite differences method tailored for high dimension problems.

Finite differences approximation

By definition, one has:

$$\frac{\partial G}{\partial x_i}(\mathbf{x}) = \lim_{\alpha_i \to 0} \frac{G(\mathbf{x} + (0, ..., \alpha_i, ..., 0)^T) - G(\mathbf{x})}{\alpha_i}$$

Thus the partial derivative (and therefore the gradient) can be numerically approximated by

$$\frac{\partial G}{\partial x_i}(\mathbf{x}) \approx \frac{G(\mathbf{x} + (0, ..., \delta x_i, ..., 0)^T) - G(\mathbf{x})}{\delta x_i}$$

where δx_i is "small", see e.g. [11]. As it has been mentioned before, this method, although very simple, has two main drawbacks (computational cost, rounding issues), which motivates the need for the adjoint code.

Discrete vs. continuous adjoint

There are two approaches to obtain an adjoint code:

- discretize the continuous direct model, and then write the adjoint of the discrete direct model. This is generally called the *discrete adjoint* approach, see *e.g.* [17].
- write the continuous adjoint from the continuous direct model (as explained before), then discretize the continuous adjoint equations. This is called the *continuous adjoint approach.*

The two approaches are not equivalent. As an example, a simple ordinary differential equation can be considered:

$$c'(t) = F(t).c(t)$$

where F is a time dependant linear operator acting on the vector $c(t)$. If this model is discretized using a forward Euler scheme, a typical time step writes as

$$c_{n+1} = (I + \Delta_t F_n)c_n$$

where I is the identity matrix and n the time index. Then the adjoint of this discrete equation would give the following typical time step (*discrete adjoint*)

$$c^*_n = (I + \Delta_t F_n)^T c^*_{n+1}$$

where $*$ denotes the adjoint variable. On the other hand, the continuous adjoint equation is

$$-c^{*\prime}(t) = F(t)^T.c^*(t)$$

If the same forward Euler explicit scheme is chosen, one obtains for the *continuous adjoint*:

$$c^*_n = (I + \Delta_t F_{n+1})^T c^*_{n+1}$$

and the time dependency on F then implies that both approaches are not identical. This illustrates the fact that discretisation and transposition (the word "adjointization" does not exist, the process of obtaining an adjoint code is called "transposition" or "derivation") do not, in general, commute.

The choice of the discrete adjoint approach should be immediate for two main reasons:

1. The response function $G(\mathbf{x})$ is computed through the discrete direct model, its gradient is therefore given by the *discrete adjoint*. The *continuous adjoint* gives only an approximation of this gradient.

2. The discrete adjoint approach allows for the use of automatic adjoint compilers (software that takes in input the direct model code and produces as output the tangent and adjoint codes, e.g. Tapenade [12]).

However, it must be noted here that, for large complex systems, obtaining the discrete adjoint can be a time- and expertise-demanding task. Therefore, if one has limited time and experience in adjoint coding, and if one can be satisfied with just an approximation of the gradient, the continuous adjoint approach can be considered. Moreover, for complex non-linear or non differentiable equations, the discrete adjoint has been shown in [29] to present problems to compute the sensitivities, so that other approaches may be considered. Similarly [18] presents an application with adaptive mesh, where it is preferable to go through the continuous adjoint route. In any case, the validation of the gradient (see below) should give a good idea about the quality of the chosen gradient.

Adjoint code derivation

As it has been mentioned above, obtaining a *discrete* adjoint code is a complex task. A numerical code is ultimately a sequence of single-line instructions. In other words, a code can be seen as a composition of a (large) number of function, each line code representing one function in this composition. To obtain the tangent code (and then the adjoint code, by transposition), it is necessary to apply the chain rule to this function composition. Because of non-linearities, dependencies, inputs/outputs, applying the chain rule to a large code is very complex. There exist recipes for adjoint code construction, explaining in details how to get the adjoint for various instruction type (assignment, loop, conditional statements, inputs/outputs, and so on), see *e.g.* [9; 10]. Code differentiation can be done by hand following these recipes.

An alternative to adjoint hand-writing is to use specially designed softwares that automate the writing of the adjoint code, called automatic differentiation tools,

such as the software Tapenade [12]. These tools are powerful, always improving, and can now derive large computer codes in various programming languages. It should be mentioned, however, that these tools can not be used completely in black box mode and may require some preparatory work on the direct code. Despite these difficulties, some large-scale usages of automatic differentiation have been performed, e.g. for the ocean model of the MIT (MITgcm, see [20]) or a Greenland ice model (see below the Applications section and [13]).

Monte-Carlo approximation

Monte-Carlo approximation allows to obtain an approximate gradient using a reasonable number of response function evaluations. Here the approximation proposed in [1] is presented.

Assume that $\mathbf{h}$ is small enough, so that the following approximation holds:

$$\boldsymbol{\delta} G = G(\mathbf{x}+\mathbf{h}) - G(\mathbf{x}) \approx G'(\mathbf{x})[\mathbf{h}] = \nabla_{\mathbf{x}} G(\mathbf{x}).\mathbf{h} \tag{14}$$

By right-multiplying by $\mathbf{h}^T$ both sides of equation (14) one gets

$$\boldsymbol{\delta} G \mathbf{h}^T \approx (\nabla_{\mathbf{x}} G(\mathbf{x}).\mathbf{h})\mathbf{h}^T$$

Considering that $\mathbf{h}$ is a stochastic perturbation, one can take the expectation

$$\mathbb{E}(\boldsymbol{\delta} G \mathbf{h}^T) \approx \mathbb{E}([\nabla_{\mathbf{x}} G(\mathbf{x}).\mathbf{h}]\mathbf{h}^T) = \nabla_{\mathbf{x}} G(\mathbf{x}).\mathbf{A}$$

where $\mathbf{A} = \mathbb{E}(\mathbf{h}\mathbf{h}^T)$ is the covariance matrix of $\mathbf{h}$. Therefore

$$\nabla_{\mathbf{x}} G(\mathbf{x}) \approx \mathbb{E}(\boldsymbol{\delta} G \mathbf{h}^T)\mathbf{A}^{-1} \tag{15}$$

In practice $\mathbf{A}$ is a large matrix and may be difficult to inverse, therefore the authors propose to replace $\mathbf{A}$ by its diagonal. In the end, an approximate gradient is given by the formula:

$$\widetilde{\nabla_{\mathbf{x}} G(\mathbf{x})} = \mathbb{E}(\boldsymbol{\delta} G \mathbf{h}^T) \left(\mathrm{Diag}(\mathbf{A})\right)^{-1} \tag{16}$$

where the expectation is computed using a Monte-Carlo method, requiring a certain number of model runs. This formula is simple enough, but should be handled with care, indeed it holds three successive approximations: finite differences instead of true gradient, Monte-Carlo approximation (usually carried over with a small sample size), and A replaced by $\mathrm{Diag}(A)$. However, this kind of approach has been successfully used in data assimilation, to obtain gradient-like information without resorting to adjoint code construction, see e.g. [19] [7].

Gradient code validation

First order test

The first order test is very simple, the idea is just to check the following approximation at the first order in $\alpha \to 0$:

$$\frac{G(\mathbf{x} + \alpha.\mathbf{h}) - G(\mathbf{x})}{\alpha} = \big(\nabla G, \mathbf{h}\big) + o(1)$$

The principle of the test is then to compute, for various perturbation directions $\mathbf{h}$ and various values of α (with $\alpha \to 0$, *e.g.* $\alpha = 10^{-n}, n = 1..8$), the two following quantities: first one computes

$$\tau(\alpha, \mathbf{h}) = \frac{G(\mathbf{x} + \alpha.\mathbf{h}) - G(\mathbf{x})}{\alpha}$$

with the direct code, and then one computes $\delta(\mathbf{h}) = \big(\nabla G, \mathbf{h}\big)$, where ∇G is given by the adjoint code. Then one just has to measure the relative error

$$\varepsilon(\alpha, \mathbf{h}) = \frac{|\tau(\alpha, \mathbf{h}) - \delta(\mathbf{h})|}{|\delta(\mathbf{h})|}$$

and check that $\varepsilon(\alpha, \mathbf{h})$ tends to 0 with α for various directions $\mathbf{h}$.

Second order test

For this test the Taylor expansion at second order is written:

$$G(\mathbf{x}+\alpha\mathbf{h}) = G(\mathbf{x}) + \alpha\big(\nabla G, \mathbf{h}\big) + \frac{\alpha^2}{2}\big(\mathbf{h}, \nabla^2 G\mathbf{h}\big) + o(\alpha^2)$$

When $\mathbf{h}$ is given, the last term is therefore a constant: $G(\mathbf{x}+\alpha\mathbf{h}) = G(\mathbf{x}) + \alpha\big(\nabla G, \mathbf{h}\big) + \frac{\alpha^2}{2}C(\mathbf{h}) + o(\alpha^2)$. In that case, the second order test writes as follows. Let $\tau(\alpha, \mathbf{h})$ be defined as previously:

$$\tau(\alpha, \mathbf{h}) = \frac{G(\mathbf{x}+\alpha.\mathbf{h}) - G(\mathbf{x})}{\alpha} \quad \text{and} \quad \delta(\mathbf{h}) = \big(\nabla G, \mathbf{h}\big)$$

The taylor expansion gives $\frac{\tau(\alpha,\mathbf{h})-\delta(\mathbf{h})}{\alpha} = \frac{1}{2}C(\mathbf{h}) + o(1)$. The test consists in computing the quantity $r(\alpha, \mathbf{h})$:

$$r(\alpha, \mathbf{h}) = \frac{\tau(\alpha, \mathbf{h}) - \delta(\mathbf{h})}{\alpha}$$

for various directions $\mathbf{h}$ and various α and check that it tends to a constant (depending on $\mathbf{h}$) when $\alpha \to 0$.

Stability analysis

Stability analysis is the study of how perturbations on the system will grow. Such tools, and in particular the so-called singular vectors, can be used for sensitivity analysis. Indeed looking at extremal perturbations gives an input on sensitivities of the system (see [22; 27; 23] for application examples).

The growth rate of a given perturbation $\mathbf{h}_0$ of, say, the initial condition $\mathbf{u}_0$ of the model is classically defined by

$$\rho\left(\mathbf{h}_0\right) = \frac{\|\mathcal{M}\left(\mathbf{u}_0 + \mathbf{h}_0, T\right) - \mathcal{M}\left(\mathbf{u}_0, T\right)\|}{\|\mathbf{h}_0\|} \tag{17}$$

where $\|.\|$ is a given norm.

One can then define the optimal perturbation $\mathbf{h}_0^1$ so that $\rho\left(\mathbf{h}_0^1\right) = \max_{\mathbf{h}_0} \rho\left(\mathbf{h}_0\right)$ and then deduce a family of maximum growth vectors:

$$\rho\left(\mathbf{h}_0^i\right) = \max_{\mathbf{h}_0 \perp \mathrm{Span}\left(\mathbf{h}_0^1,\ldots,\mathbf{h}_0^{i-1}\right)} \rho\left(\mathbf{h}_0\right) \qquad , i \geq 2 \tag{18}$$

By restricting the study to the linear part of the perturbation behaviour, the growth rate becomes (denoting $\mathbf{L} = \frac{\partial \mathcal{M}}{\partial \mathbf{u}}$ for clarity):

$$\begin{aligned} \rho^2\left(\mathbf{h}_0\right) = \frac{\|\mathbf{L}\mathbf{h}_0\|^2}{\|\mathbf{h}_0\|^2} &= \frac{<\mathbf{L}\mathbf{h}_0, \mathbf{L}\mathbf{h}_0>}{<\mathbf{h}_0, \mathbf{h}_0>} \\ &= \frac{<\mathbf{h}_0, \mathbf{L}^*\mathbf{L}\mathbf{h}_0>}{<\mathbf{h}_0, \mathbf{h}_0>} \end{aligned} \tag{19}$$

$\mathbf{L}^*\mathbf{L}$ being a symmetric positive definite matrix, its eigenvalues are non-negative real and its eigenvectors are (or can be chosen) orthonormal. The strongest growth vectors are the eigenvectors of $\mathbf{L}^*\mathbf{L}$ which correspond to the greatest eigenvalues. They are called forward singular vectors and denoted f_i^+:

$$\mathbf{L}^*\mathbf{L} f_i^+ = \mu_i f_i^+ \tag{20}$$

One can notice then that $\mathbf{L}f_i^+$ is an eigenvector of $\mathbf{L}\mathbf{L}^*$, which allows to define the backward singular vectors, noted f_i^-, as:

$$\mathbf{L}f_i^+ = \sqrt{\mu_i}\, f_i^-$$

the eigenvalue corresponding to f_i^- is μ_i as well. Forward singular vectors represent the directions of perturbation that will grow fastest, while backward singular vector represent the directions of perturbation that have grown the most.

The computation of the f_i^+ and f_i^- generally requires numerous matrix-vector multiplications, *i.e.* direct integrations of the model and backward adjoint integrations. The result of these calculations depends on the norm used, the time window and the initial state if the model is nonlinear. For an infinite time window, singular vectors converge toward Llyapunov vectors. The full non linear model can be retained in equation (17) leading to the computation of so called non-linear singular vectors. They are obtained by optimising directly equation (17). However due to non-linear dissipation,

they tend to converge toward infinitesimal perturbations as the time window lengthen, this can be sorted out by adding some constraints on the norm of the perturbation [21].

Applications

Applications of VSA can be generally divided into two classes: sensitivity to initial or boundary conditions changes and sensitivity to parameter changes. However one can extend the notion of sensitivity analysis to second order sensitivity and stability analysis. This section is organised following this classification. Even though VSA has been used extensively in a wide range of problems, examples given here come from geophysical applications. Indeed, in that case the control vector is generally of very large dimension therefore global SA techniques are out of reach. Moreover quite often tangent and/or adjoint models are already available since used for data assimilation.

Sensitivity to initial or boundary conditions changes

Sensitivity to initial conditions changes ($\mathbf{x} = \mathbf{u}_0$) is routinely used in Numerical Weather Prediction systems. In that case the model (1) describes the evolution of the atmosphere, initialised with $\mathbf{u}_0$. The parameter vector $\mathbf{x} = \mathbf{u}_0$ represents the full state of the atmosphere. In modern atmospheric models that amounts to $10^9 - 10^{10}$ unknowns that are correlated.

The response function follows the form of (4), with

$$\mathcal{H}(\mathbf{u}(\mathbf{u}_0)) = \| \mathbf{z}^{obs}(t) - H[\mathbf{u}(\mathbf{u}_0; t)] \|^2_{\mathcal{O}}$$

where $\mathbf{z}^{obs}$ are observations of the system, H maps the state vector to the observation space and $\| \, . \, \|_{\mathcal{O}}$ is typically a weighted L^2 norm. Following (4), the response function is then

$$G(\mathbf{u}_0) = \frac{1}{2}\int_0^T \parallel \mathbf{z}^{obs}(t) - H[\mathbf{u}(\mathbf{u}_0;t)] \parallel^2_{\mathcal{O}} dt \tag{21}$$

Local sensitivities of such functions can be very useful to understand the behaviour of a system and has been extensively used in geosciences (see for instance [28], [8], [34] or [2]).

One can find an example of application of such methods on the Mercator-ocean's 1/4° global ocean model in [31]. The initial objective was to try to estimate the influences of geographical areas to reduce the forecast error using an adjoint method to compute the sensitivities. A preliminary study has been conducted by considering the misfit to observations as a proxy of the forecast error and sought to determine the sensitivity of this misfit to regional changes in the initial condition and/or to forcing. That should give an indication about the important phenomena to consider to improve this system.

The most easily interpreted case in this study is to consider a sensitivity criterion coming from the difference in sea surface temperature (SST) maps at the final instant of the assimilation cycle, because of its dense coverage in space. The response function (21) is a discrete version of the time integral, in which the operator H (mapping the state vector $\mathbf{u}$ to the observation space) simply extracts the SST of the state vector. This can be translated into computing the gradient:

$$G(\mathbf{u}_0, \mathbf{q}) = \frac{1}{2}\sum_{n=1}^{N_{SST}} \parallel H_{SST}(\mathbf{u}_n) - SST^{obs} \parallel^2_{\mathbf{R}^{-1}} \tag{22}$$

with a parameter vector $\mathbf{x} = (\mathbf{u}_0, \mathbf{q})$ made of $\mathbf{u}_0 = (u_0, v_0, T_0, S_0, \eta_0)^T$ the initial state vector (current velocity components, temperature, salinity and sea level) and of $\mathbf{q} = (q_{sr}, q_{ns}, emp)^T$ (radiative fluxes, total heat fluxes, fresh water fluxes) and SST^{obs} observations of SST.

One can see an example of sensitivity to initial temperature (surface and 100m) as shown in the two bottom panels of Figure 1. High sensitivity will give a signal similar

to the gap in observations (top left), while low sensitivity will show a white area. In this example it is clear that the SST misfit is highly sensitive to changes in surface temperature where the initial mixed layer depth (top right) is low and insensitive elsewhere. The opposite conclusion can be drawn from the sensitivity to the initial temperature at 100m. This is obviously not a surprise, and corresponds more to the purpose of verification of the model rather than system improvement. However it highlights the importance of having a good estimate of the vertical mixing. Other components of the gradient show the important role of atmospheric forcing (again this could have been anticipated) and ways to improve the system also appear to point to that direction.

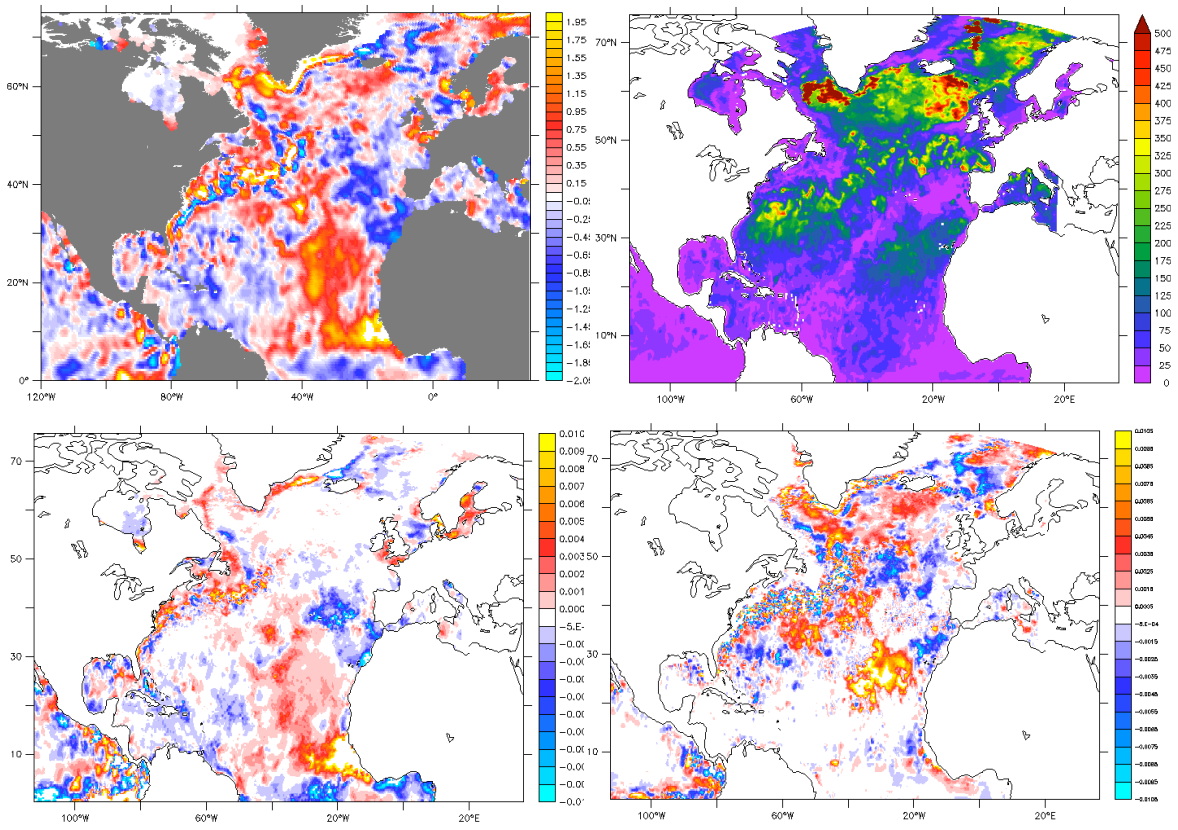


Fig. 1. Top: misfit between forecast and observed SST (left) and mixed layer depth (right). Bottom: sensitivity to one week lead time SST error with respect to variations in initial surface (left) and 100m (right) temperature (from [31]).

This kind of study is also routinely used to target observations. For example, in order to be able to track tropical cyclones it is possible to use the so-called Adjoint-Derived Sensitivity Steering Vectors (ADSSV, [32; 14; 5]). In that case, the model equation (1) represents the evolution of the atmosphere, starting from an initial state vector $\mathbf{u}_0$. Some technicalities allow to choose the parameter vector $\mathbf{x}$ equal to the vorticity at initial time $\mathbf{x} = \xi_0 = \frac{\partial v_0}{\partial x} - \frac{\partial u_0}{\partial y}$. Then the response function follows the form (9), it is a two-criteria functional at a given verification time t_1:

$$G(\xi_0) = \left(\frac{1}{|A|} \int_A u(t_1)\, dx, \frac{1}{|A|} \int_A v(t_1)\, dx \right)^T \tag{23}$$

where $\frac{1}{|A|} \int_A u\, dx$ and $\frac{1}{|A|} \int_A v\, dx$ are the zonal and meridional averaged wind velocities over a given area of interest A. By looking at the sensitivities to ξ_0: $\frac{\partial G_u}{\partial \xi_0}$ and $\frac{\partial G_v}{\partial \xi_0}$ one will get information about the way the tropical cyclone is likely to go. For example, if at a given forecast time at one particular grid point the ADSSV vector points to the east, an increase in the vorticity at this very point at the observing time would be associated with an increase in the eastward steering flow of the storm at the verifying time [32]. This information, in turn, helps to decide where to launch useful observations.

Parameter sensitivity

The previous section focuses on variable input quantities (initial/boundary conditions or forcings), however most of numerical models also rely on a set of physical parameters. They are generally only approximatively known and their settings depend on the studied case. Methods to measure sensitivities to parameter changes are the same as before, differences mostly lie in the parameter set nature: it is generally of small to medium size and elements are mostly uncorrelated. To both respect, they may be better suited for GSA. However in some cases these parameters can for instance vary spatially and therefore be out of reach of global analysis.

Examples of spatially varying parameters are quite common in geophysics, and an example in glaciology will be given here. In the framework of global change and in particular sea-level change, the volume evolution of the two main ice caps (Antarctica and Greenland) is of crucial interest. In ice-cap modelling, experts consider that the basal characteristics of the ice cap are particularly important: basal melt rate and basal sliding coefficient (linked to the sliding velocity). These basal characteritics can be considered as parameters (they are intrinsic to the modelled system) while being spatially varying, and their influence on the ice cap volume must be quantified in order to better understand and predict the future volume evolution.

This has been studied for example in [13], where the authors use adjoint methods to compute the sensitivities of the ice volume V over Greenland to perturbations on the basal sliding c_b and the basal melt rate q_{bm}. Let us note here that in this case the adjoint model has been obtained using automatic differentiation tools. Figure 2 shows that sliding sensitivities exhibit significant regional variations and are mostly located in the coastal areas whereas melt-rate sensitivities are either fairly uniform or they completely vanish. This kind of information is of prime importance when tuning the model parameter and/or designing an observation campaign for measuring said parameters. For instance, in this particular system, there is no gain to be expected by focusing on the interior of the domain.

Sensitivity of complex systems

Previously presented examples focus on looking for sensitivities of a given model to perturbations. However these approaches can be extended to more complex problems such as coupled models for instance, or even a forecasting system *i.e.* a modelling system that also includes an initialisation scheme. Most of the time this initialisation is done through the so called data assimilation techniques, where observations from the past are

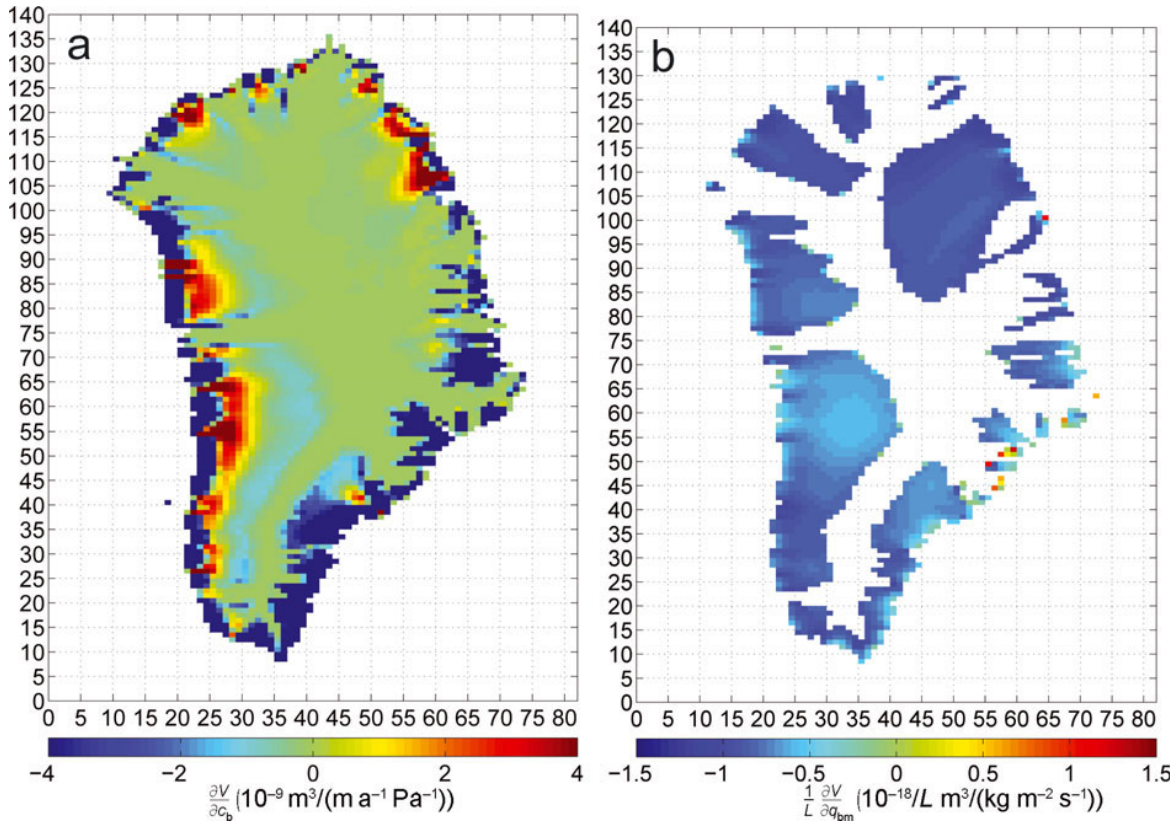


Fig. 2. Adjoint sensitivity maps of the total Greenland ice volume V related to (a) basal sliding c_b and (b) basal melt rate q_{bm}. (from [13], copyright International Glaciological Society).

used to adjust the present state of the system. There are two kinds of such techniques, either based on filtering approaches or variational methods. Filtering techniques aim at bringing the model trajectory closer to observations through a sequence of prediction and correction steps. *i.e.* the model is corrected as follows

$$\frac{d\mathbf{u}}{dt} = \mathcal{M}(\mathbf{u};\mathbf{x}) + \mathbf{K}\left(H(\mathbf{u}(t)) - \mathbf{y}^{obs}(t)\right) \tag{24}$$

where $\mathbf{K}$ is a gain matrix. For the simplest versions of filtering data assimilation, $\mathbf{K}$ does no depend on $\mathbf{u}(t)$, so computing sensitivities to such system can be done similarly as before (see [33] for an example). However in more sophisticated approach, like variational data assimilation, it is less straightforward. In that case the data assimilation problem is solved through the minimisation of a cost function that is typically equation (21). Looking for local sensitivities on the forecasting system would mean to look for sensitivities to the optimal solution of the minimisation of (21), that is to say

to compute the gradient of the whole optimality system (direct model, adjoint model, cost function); doing so by adjoint method may require the second order adjoint model [16; 6].

Another example of complex system is to perform a sensitivity analysis on a stability analysis (*i.e.* how given perturbations will affect the stability of a system).

In [27] the authors use stability analysis to study the sensitivity of the thermohaline oceanic circulation (large scale circulation, mostly dominated by density variations, *i.e.* temperature and salinity variations). To do so, they look for the optimal initial perturbation of the sea surface salinity that induces the largest variation of the thermohaline circulation.

In [23] the authors are interested in the moist predictability in meteorological models. Since this is a very non linear process they propose to use non-linear singular vectors as response function G:

$$G(\mathbf{u}_0) = \underset{\|\mathbf{h}_0\|=E_0}{\arg\max} \left(\frac{\|\mathcal{M}(\mathbf{u}_0 + \mathbf{h}_0, T) - \mathcal{M}(\mathbf{u}_0, T)\|}{\|\mathbf{h}_0\|} \right) \tag{25}$$

This tells which variation in the initial condition will affect the most the optimal perturbations, and then the predictability. Note that in that case, computing $\nabla_{\mathbf{u}_0} G(\mathbf{u}_0)$ also requires the second order adjoint.

Obviously, these are only a handful of possible applications among many; as long as one defines a response function, one could be interested by studying its sensitivities.

Conclusion

Variational methods are local sensitivity analysis techniques, they gather a set of methods from very basic to sophisticated as presented above. The main advantage is that they can be used for very large dimension problems if using adjoint methods, the downside is that it may require some heavy developments. This burden

can be reduced however by the use of automatic differentiation tools. They have been used for a very wide range of applications, and even on a daily basis in operational numerical weather prediction. Although they are local by essence adjoint based-variational methods can be extended to global sensitivity analysis as will be presented in Derivative-based Global sensitivity measure.

Methods presented here are dedicated to first order local sensitivity analysis. This can be extended to interaction studies by using the second order derivatives (Hessian), which can be computed similarly using so-called second order adjoint models.

Readers interested in going further could start with clearly written and easy to read papers such as [4; 13; 17]. To go further, the book [3] and the application paper [2] are recommended. And finally, about second order derivatives, [16] provides a nice introduction, and [23] offers an advanced application.

Cross-References

Derivative-based Global sensitivity measure

Sensitivity analysis of spatial and/or temporal phenomena

Variables' Weights and Importance in Arithmetic Averages: Two Stories to Tell

References

1. Ancell B, Hakim GJ (2007) Comparing Adjoint- and Ensemble-Sensitivity Analysis with Applications to Observation Targeting. Monthly Weather Review 135(12):4117–4134
2. Ayoub N (2006) Estimation of boundary values in a North Atlantic circulation model using an adjoint method. Ocean Modelling 12(3-4):319–347
3. Cacuci DG (2005) Sensitivity and Uncertainty Analysis: Theory. CRC Press

4. Castaings W, Dartus D, Le Dimet FX, Saulnier GM (2009) Sensitivity analysis and parameter estimation for distributed hydrological modeling: potential of variational methods. Hydrology & Earth System Sciences 13(4)
5. Chen SG, Wu CC, Chen JH, Chou KH (2011) Validation and Interpretation of Adjoint-Derived Sensitivity Steering Vector as Targeted Observation Guidance. Monthly Weather Review pp 1608–1625
6. Daescu DN, Navon IM (2008) Reduced-Order Observation Sensitivity In 4D-Var Data Assimilation. In: American Meteorological Society 88th AMS Annual Meeting New Orleans, LA
7. Desroziers G, Camino JT, Berre L (2014) 4DEnVar: link with 4D state formulation of variational assimilation and different possible implementations. Quarterly Journal of the Royal Meteorological Society (140):2097–2110
8. Errico RM, Vukicevic T (1992) Sensitivity Analysis Using an Adjoint of the PSU-NCAR mesoseale Model. Monthly Weather Review 120(8):1644–1660
9. Giering R, Kaminski T (1998) Recipes for adjoint code construction. ACM Transactions on Mathematical Software (TOMS) 24(4):437–474
10. Griewank A, Walther A (2008) Evaluating derivatives: principles and techniques of algorithmic differentiation. Siam
11. Hamby DM (1994) A review of techniques for parameter sensitivity analysis of environmental models. Environmental Monitoring and Assessment 32(2):135–154
12. Hascoet L, Pascual V (2013) The Tapenade Automatic Differentiation tool: principles, model, and specification. ACM Transactions on Mathematical Software (TOMS) 39(3):20
13. Heimbach P, Bugnion V (2009) Greenland ice-sheet volume sensitivity to basal, surface and initial conditions derived from an adjoint model. Annals of Glaciology 50:67–80
14. Hoover BT, Morgan MC (2011) Dynamical Sensitivity Analysis of Tropical Cyclone Steering Using an Adjoint Model. Monthly Weather Review (139):2761–2775
15. Lauvernet C, Hascoët L, Dimet FXL, Baret F (2012) Using automatic differentiation to study the sensitivity of a crop model. In: Forth S, Hovland P, Phipps E, Utke J, Walther A (eds) Recent Advances in Algorithmic Differentiation, Lecture Notes in Computational Science and Engineering, vol 87, Springer, Berlin, pp 59–69
16. Le Dimet FX, Ngodock HE, Luong B, Verron J (1997) Sensitivity analysis in variational data assimilation. Journal-Meteorological Society of Japan Series 2 75:135–145

17. Lellouche JM, Devenon JL, Dekeyser I (1994) Boundary control of Burgers' equation—a numerical approach. Computers & Mathematics with Applications 28(5):33–34
18. Li S, Petzold L (2004) Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement. Journal of Computational Physics 198(1):310–325
19. Liu C, Xiao Q, Wang B (2008) An Ensemble-Based Four-Dimensional Variational Data Assimilation Scheme. Part I: Technical Formulation and Preliminary Test. Monthly Weather Review 136(9):3363–3373
20. Marotzke J, Wunsch C, Giering R, Zhang K, Stammer D, Hill C, Lee T (1999) Construction of the adjoint MIT ocean general circulation model and application to atlantic heat transport sensitivity. J Geophys Res 104(29):529–29
21. Mu M, Duan W, Wang B (2003) Conditional nonlinear optimal perturbation and its applications. Nonlin Processes Geophys 10(6):493–501
22. Qin X, Mu M (2011) Influence of conditional nonlinear optimal perturbations sensitivity on typhoon track forecasts. QJR Meteorol Soc 138:185–197
23. Rivière O, Lapeyre G, Talagrand O (2009) A novel technique for nonlinear sensitivity analysis: application to moist predictability. QJR Meteorol Soc 135(643):1520–1537
24. Saltelli A, Ratto M, Tarantola S, Campolongo F (2005) Sensitivity analysis for chemical models. Chemical reviews 105(7):2811–2828
25. Sandu A, Daescu DN, Carmichael GR (2003) Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: Part I—theory and software tools. Atmospheric Environment 37(36):5083–5096
26. Sandu A, Daescu DN, Carmichael GR, Chai T (2005) Adjoint sensitivity analysis of regional air quality models. Journal of Computational Physics 204(1):222–252
27. Sévellec F (2007) Optimal Surface Salinity Perturbations Influencing the Thermohaline Circulation. Journal of Physical Oceanography 37(12):2789–2808
28. Sykes JF, Wilson JL, Andrews RW (1985) Sensitivity analysis for steady state groundwater flow using adjoint operators. Water Resources Research 21(3):359–371
29. Thuburn J, Haine TWN (2001) Adjoints of Nonoscillatory Advection Schemes. Journal of Computational Physics 171(2):616–631
30. Vidard A (2012) Data assimilation and adjoint methods for geophysical applications. PhD thesis, Université de Grenoble, Habilitation thesis

31. Vidard A, Rémy E, Greiner E (2011) Sensitivity analysis through adjoint method: application to the GLORYS reanalysis. Contrat n° 08/D43, Mercator Océan

32. Wu CC, Chen JH, Lin PH, Chou KH (2007) Targeted Observations of Tropical Cyclone Movement Based on the Adjoint-Derived Sensitivity Steering Vector. J Atmos Sci 64(7):2611–2626

33. Zhu Y, Gelaro R (2008) Observation sensitivity calculations using the adjoint of the Gridpoint Statistical Interpolation (GSI) analysis system. Monthly Weather Review 136(1):335–351

34. Zou X, Barcilon A, Navon IM, Whitaker J, Cacuci DG (1993) An Adjoint Sensitivity Study of Blocking in a Two-Layer Isentropic Model. Monthly Weather Review 121:2833–2857