

Variational Data Assimilation in A Class of Environmental Disasters: Mathematical Aspects

Paulo S. D. da Silva¹ & Kevin A. Vasconcellos²

¹ Laboratory of Mathematical Science, Northern Fluminense State University, Rio de Janeiro, Brazil

² Laboratory of Metallurgical Engineering, Northern Fluminense State University, Rio de Janeiro, Brazil

Correspondence: Paulo S. D. da Silva, Laboratório de Ciências Matemáticas, Universidade Estadual do Norte Fluminense, Avenida Alberto Lamego, 2000 Horto, Campos dos Goytacazes, Rio de Janeiro, Brazil. E-mail: paulosd@uenf.br

Received: October 10, 2014 Accepted: October 24, 2014 Online Published: January 30, 2015

doi:10.5539/esr.v4n1p75 URL: <http://dx.doi.org/10.5539/esr.v4n1p75>

Abstract

Variational data assimilation can be a powerful allied in order to estimate the determination of the trajectory of a pollutant accidentally spilled on the sea surface, when a set of observation of the resulting pollutant spot is available, as well as a computational model of the sea surface dynamics of the region of the accident. In order to solve the computational problem, the adjoint equations method will be of the paramount importance. Here it is presented the mathematical foundations of the problem, a brief discussion of some computational issues, and an very simple example is discussed.

Keywords: variational data assimilation, adjoint equations, numerical models

1. Introduction

In several accidents involving the spillage of pollutants in the sea surface, one of the main problems faced by the authorities is to determine where to combat the resulting spot. The experience have shown this is a hardy problem. For example, in 2000, there was a terrible accident involving the Brazilian company, Petrobras, in a bay known as Baía de Guanabara, in Rio de Janeiro. An large amount of oil leaked from Reduc (see Figure 1), and, despite all the efforts, it reached in less than 48 hours I do Governador, São Gonçalo and Niterói, as in a real cat and mouse game. It was estimated some mangrove with a rich biodiversity would need some 20-30 years to recover its original state, in an optimist scenario. At that time, we were trying to understand the mathematical foundations of Variational data assimilation in the meteorological context, as a way to improve the numerical weather predictions, and we thought that technique could be used to obtain a better estimation of the trajectory of the oil spot, using the available information. In fact, from the mathematical view point, both problems have the same structure, namely, a set of partial differential equations, with initial and boundary conditions, describing the problem dynamics, and also we could have a set of information of the the object we want to estimate, that is, meteorological variables, in one case, and the shape and location of the spot, in the other.

Data assimilation, in its two main approach, sequential and variational, establishes that the assimilation of all available information (roughly, observations) will improve the predictive ability of the model, and that is a passive process, in the sense that it does not interfere with the model itself.

The key point in the variational data assimilation is the assumption of the existence of an initial condition (or model state) such that, when this condition is used as the initial data of the model, the subsequent states of the model will minimize a functional that measures the discrepancy between model and observed states, and, if so, the procedure is more than adequate as a first step to solve the problem of the oil spillage problem.

Although the data Assimilation method can be invaluable in computational simulations, since observations of the systems act ont the numerical model as restriction which the solution must verify, with the exception of a few articles and books, the mathematical formalism is not always clearly presented, and this is not an appropriated situation to the development of the method, since important aspects for its future must be placed in solid theoretical foundations.

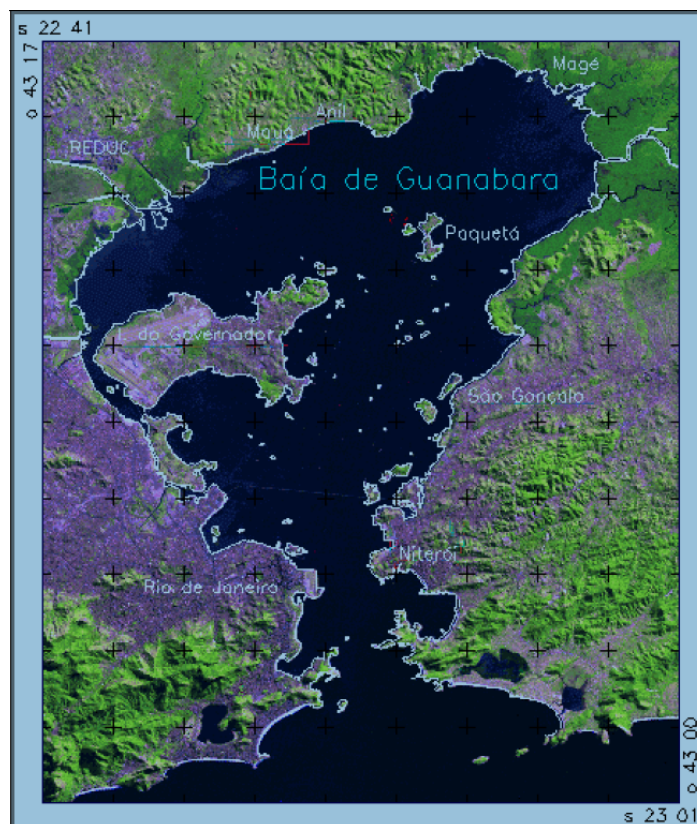


Figure 1. Guanabara Bay (Source: INPE)

Therefore, is our intention to present a rigorous mathematical formulation not only for Variational Data Assimilation but also Adjoint Equation Method as well, so as to make them perfectly understood.

In order to realize the numerical control of the space-time evolution of a system involving a geophysical flow it is necessary to know its initial condition, which is, the numerical counterpart of the real initial state or the initial configuration of the system under study. In the case of geophysical flows, the dimension and the geographic location of the spatial domain problems make the distribution of an observational network difficult, what causes a scarcity of data about the phenomenon studied or even turns the initial configuration unavailable for computational simulations (Talagrand, 1997). As in those flows are presented phenomena of difficult representation in a numerical model, and even in analytical models, being, in certain situations, processes highly sensitive to initial conditions variations, to establish an initial condition adequate for simulations, in this case, is of the crucial importance for the results.

In fact, although we have not considered here the question, we should mention that due to the range of space-time scales present in oceanographic problems, any numerical model designed to solve one of those problems will face numerical instabilities related to the fractal structures inherent in those situations (Candy, 2009)

For an adequate initial condition of the numerical experiment, one means that one which generates the numerical model states that fit, in accordance with the desired precision, the available observations. In order to decide whether or not an initial condition is appropriated to certain simulation, one should consider, in addition to the flow dynamics, the full set of observations of the system being studied. That is exactly what is performed by the methodology known as Data Assimilation, approach in which all the knowledge on a system being analyzed, its dynamics and observations, is combined in order to produce an initial condition that minimize the error between the model generated configurations and the correspondent known configurations, in a sense to be precise later.

A problem involving geophysical flows in its discrete form produce sa very high number of variables, so that, straight away, Variational Data Assimilation is a methodology computationally intractable. A solution for this difficulty can be obtained with the use of the Adjoint Equations Method, which adequately insert the problem of the initial condition determination to simulations of geophysical flows in the domain of Optimal Control Theory, and

its mathematical foundations are analyzed here with the necessary accuracy. A second consequence of the resulting methodology of combining Variational Data Assimilation with Adjoint Equation Method is the transformation of a minimization problem with restrictions, as originally proposed by the Variational Data Assimilation formalism, in a minimization problem without, what makes it possible the use of more agile routines, as L-BFGS (Liu and Nocedal, 1989), in the determination of an optimal initial condition.

In the first section of this work, the mathematical foundations of two crucial methods are presented, Variational Data Assimilation and Adjoint Equations Method. In the second section, in order to fix ideas, one works with a one-dimensional model which permits the exploration of several aspects of the methods used and also to develop explicitly the adjoint of a given equation. Finally, in the third section, one describes a bi-dimensional numerical experiment, showing with some detail (i) an algebraic formalism, fundamental in the development of a computational program to perform the experiment, formalism that has already produced, in the Computational Linear Algebra domain, a new line of investigation, known as Automatic Differentiation and as Automatic Adjoint Generation (Faure and Papegay, 1998), and (ii) the structure of a FORTRAN code, created by the author, in order to realize the Variational Data Assimilation of a flow described by the bi-dimensional advection-diffusion equation.

2. The Mathematical Approach

Definition Let A be an open set in \mathbb{R}^n and $f : A \rightarrow \mathbb{R}$ a differentiable function. Then, $\forall x \in A$, the differential of f in $x \in A$, is the linear functional such that, $\forall v \in \mathbb{R}^n$,

$$df(x) \cdot v = \lim_{t \rightarrow 0} \frac{f(x + tv) - f(x)}{t}$$

As $df(x)$ is a linear functional in $(\mathbb{R}^n)^*$, the dual vector space of \mathbb{R}^n , it follows from Linear Algebra the existence and uniqueness of the vector $\nabla f(x) \in \mathbb{R}^n$ such that

$$df(x) \cdot v = \langle \nabla f(x), v \rangle,$$

where \langle, \rangle is the canonical inner product in \mathbb{R}^n .

2.1 The Data Assimilation Method

Let S be an evolutionary system, observed during the time interval $[t_1, t_2]$, from which is available a set of observations, all of them collected in the same time interval and distributed in the spatial domain of interest on S . Besides, it is known the dynamics of S . Based on those pieces of information, one wants to determine the initial condition of the modeled S , or the numerical representation of the configuration of S at the time t_1 , the starting moment for a computational simulation of S generates configurations to approach, within an acceptable precision, to the available observations of S at correspondent instants of time, in order to obtain a configuration of the model at $t_f > t_2$ which reproduces the real configuration of S at time t_f , also within an acceptable precision. From a conceptual point of view, this process is similar to the Best Linear Unbiased Estimation (Sorensen, 1980) of the configuration of S at t_f , given the S configurations within $[t_1, t_2]$, here solved in a deterministic approach as an optimal control problem, in which the initial conditions is the control data. At first, one considers the continuous version of the Variational Data Assimilation problem, which admits the appropriate mathematical treatment. In this case, the mathematical objects are:

- the system observations: given by $Z : [t_1, t_2] \times V \rightarrow V$, such that, $\forall t \in [t_1, t_2]$, $Z_t : V \rightarrow V$ is a differential operator defined on the vector space V provided with the inner product \langle, \rangle , where $[t_1, t_2]$ is the assimilation interval or assimilation window.
- the dynamics of the system S , described by

$$\frac{\partial X}{\partial t} = F(X), \quad (1)$$

where $X : [t_1, t_2] \times V \rightarrow V$ is the system S trajectory during the assimilation interval, $[t_1, t_2]$, $E = \{Y; Y : [t_1, t_2] \times V \rightarrow V \text{ and } Y \in C^2\}$ and $F : E \rightarrow E$ is a differential operator.

- the “weight” function $W : [t_1, t_2] \rightarrow \mathcal{L}(V)$, where $\mathcal{L}(V)$ is the vector space of the linear operators in V , that results from the statistics information of the instruments used to collect the data on S , and that, each $t \in [t_1, t_2]$, associate the injective linear operator $W(t) : V \rightarrow V$.

- the quadratic functional

$$J : E \rightarrow \mathbb{R} \quad (2)$$

$$X \mapsto \frac{1}{2} \int_D \langle W(t)(X(t, x) - Z(t, x)), W(t)(X(t, x) - Z(t, x)) \rangle$$

Then the Variational Data Assimilation problem is to find a trajectory of the system states S , $X : [t_1, t_2] \times V \rightarrow V$ such that X is the solution of equation (1) which minimize the functional (2), in other words, we arrive to the following minimization problem with constraint:

Problem M_R : find the solution of equation (1) minimizing the functional (2)

It would be much more interesting (and, in many problems, the only feasible way) if instead of the solution $X \in E$ one searches the solution $X_I \in E_I$, where $E_I = \{X_I; X_I = X|_{[t_1] \times V}, X \in E \text{ and } X \in C^2\}$ is isomorph to the set of operators in V , which could be called the space of the initial configurations of S , that minimizes the restriction of the functional J to E_I .

However, it is not known an explicit relation between J and X_I , since J is not a function of X . Nevertheless, if the problem in equation (1) is well posed, in the Hadamard sense, then the knowledge of X and the knowledge X_I are equivalent. The Adjoint Equation Method provides, from the relation between X and X_I , the differential of the restriction of J to E_I .

2.2 The Adjoint Equations Method

One rewrites the functional in equation(1)3 as

$$J : E \rightarrow \mathbb{R} \quad (3)$$

$$X \mapsto \int_D T(X(t, x))dD,$$

where

$$T : E \rightarrow \mathbb{R}$$

$$X(\cdot, \cdot) \mapsto \frac{1}{2} \langle W(\cdot, \cdot)(X(\cdot, \cdot) - Z(\cdot, \cdot)), W(\cdot, \cdot)(X(\cdot, \cdot) - Z(\cdot, \cdot)) \rangle.$$

As the differential of J in $X \in E$ is a linear functional in the Hilbert space E with the inner product

$$\langle X(\cdot, \cdot), Y(\cdot, \cdot) \rangle_E = \int_D \langle W(\cdot, \cdot)X(\cdot, \cdot), W(\cdot, \cdot)Y(\cdot, \cdot) \rangle dD,$$

by using the Riesz' Representation Theorem (Kreyszig, 1989), one obtains, $\forall H \in E$,

$$dJ(H) \cdot X = \langle \nabla_X J(H), X \rangle_E = \int_D \langle \nabla_X T(t, x), H(t) \rangle dD, \quad (4)$$

where $\nabla_X J$ and $\nabla_X T$ are the gradients of J and T , respectively, with respect to X .

Now, one considers the linear version of the equation (1)

$$\frac{\partial X}{\partial t} - \frac{\partial F}{\partial X} X = 0 \quad (5)$$

resulting from the substitution of the operator F for its first order approximation in an equilibrium point, omitted from the equation, and in which it was used the notation $(\frac{\partial F}{\partial X})$ for the differential Of F , and its adjoint equation is

$$\frac{\partial X}{\partial t} - (\frac{\partial F}{\partial X})^* X + \nabla_X T = 0, \quad (6)$$

where $(\frac{\partial F}{\partial X})^*$ is the adjoint operator of $\frac{\partial F}{\partial X}$.

Then one obtains the following result:

Theorem 1: Given a solution $X \in E$ of M_R , $X_I = X|_{[t_1] \times V}$ is the solution of the following unconstrained minimization problem:

Problem M_U : find $\delta x \in E_I$ minimizing $\langle \nabla_{X_I} J, \delta x \rangle$.

Proof: Let $Y : [t_1, t_2] \times V \rightarrow V$ be the solution of (5) such that $Y(t_2) = 0$ and $\delta X \in E_I$ is a perturbation of any solution of equation (6). Then it follows that

$$\begin{aligned} \left\langle \frac{\partial X}{\partial t} - \frac{\partial F}{\partial X} \delta X, Y \right\rangle + \left\langle \frac{\partial Y}{\partial t} + \left(\frac{\partial F}{\partial X} \right)^* Y + \nabla_X T, \delta X \right\rangle &= 0 \\ \therefore \frac{\partial}{\partial t} \langle Y, \delta X \rangle + \langle \nabla_X T, \delta X \rangle &= 0 \\ \therefore \frac{\partial}{\partial t} \langle Y, \delta X \rangle &= -\langle \nabla_X T, \delta X \rangle \\ \therefore \langle Y, \delta X \rangle \Big|_{t_1}^{t_2} &= - \int_D \langle \nabla_X T, \delta X \rangle dD = -\langle \nabla_X T J, \delta X \rangle \end{aligned}$$

$$\therefore \langle Y_I, \delta X_I \rangle = \langle \nabla_X J, \delta X \rangle \quad (7)$$

Equation (7) is the main point to be addressed by the Adjoint Equation in this context, and its meaning must be clear. The right side of the equality is the expression of the differential of J (relative to X), $dJ(X)$. As X_I is the projection of X onto E_I , the left side of equation (7) is the expression of the differential of the restriction of J (relative to X_I) to E_I . By the uniqueness of the representation of the last differential as an inner product in E_I , it follows that Y_I is precisely the gradient of the restriction of the functional J to E_I , that is, Y_I is the projection of $\nabla_X J$ onto E_I . Then, one obtains

$$Y_I = \nabla_{X_I} J \quad (8)$$

Therefore, using the Adjoint Equation Method, one projects the solution set $C^2([t_1, t_2] \times V, V)$ onto $C^2(t_1 \times V, V)$, which represents, from the computational viewpoint, a considerable reduction in the number of variables in the discrete problem. Besides, this projection transforms a constraint minimization problem in one unconstrained problem, namely, the problem of minimizing equation (2) with the restriction equation (1), becomes

$$\min_{\delta X \in X_I} \langle \nabla_{X_I} J, \delta X \rangle$$

The problem M_R , involving the quadratic functional J , defined in equation (2), has the existence of its solution guaranteed but not its uniqueness. In order to obtain the uniqueness of the solution of M_R , and of M_U , one redefines the problem using the following quadratic functional:

$$\begin{aligned} J' : E &\rightarrow \mathbb{R} \\ X &\mapsto J_B(X) + J(X) \end{aligned} \quad (9)$$

where, X_b is an available estimate of the initial configuration of X , and $B \in L(V)$ is the operator determined by the available statistical information on the estimate of X_b and

$$\begin{aligned} J' : E &\rightarrow \mathbb{R} \\ X &\mapsto \frac{1}{2} \int_D \langle B(X_1 - X_b), B(X_1 - X_b) \rangle_{E_I} dx \end{aligned} \quad (10)$$

Then, one obtains the following result:

Theorem 2: Given

Problem $M_{R'}$: find the solution of equation (1) that minimizes equation (10),

there exists a unique solution of $M_{R'}$, and $X_1 = X|_{E_I} \in E_I$ is the solution of the following unconstrained minimization problem

Problem $M_{U'}$: find the $\delta X_1 \in E_I$ that minimizes $\langle \nabla_{X_I} J', \delta X_1 \rangle_{E_I}$.

Proof: It is only necessary to redefine the inner product on E as

$$\langle X(\cdot, \cdot), Y(\cdot, \cdot) \rangle_E = \frac{1}{2} \int_D \langle B(X(t_1, \cdot)), B(Y(t_1, \cdot)) \rangle_{E_I} + \langle B(X(\cdot, \cdot)), B(Y(\cdot, \cdot)) \rangle dD$$

Let X_b be the available estimate of the initial configuration of X . The problem of optimal solution will be iteratively solved by shearching the perturbation $\delta X_I \in E_I$ of $\delta X \in E$ such that $(X_b + \delta X)$ will be the solution of $M_{U'}$.

Let us emphasize once again this very important point: we need the adjoint code (of a giving code) in order to obtain the differential of the functional J with respect to the initial conditions $X_I \in E_I$

3. Implementing the Adjoint Equation Method

Another interesting and also important feature of the implementation of Variational Data Assimilation is the development of automatic routines to perform the data assimilation using the Adjoint Equation Method. Nowadays, due to the several academic and industrial application of the method and considering, the extension of the work needed to produce such computational routine, the automatic generation of the adjoint to a given code can be realized by computational codes developed for this task (Faure and Papegay, 1998). As a good knowledge of the rules used in the preparation of the linear and the adjoint programs to a given computational code precede the correct use of such automatic routines, it is presented here some illustrative examples of the rules to be followed when manually generating these codes. Of course, the rules are the same used by automatic routines.

3.1 The Linear Code

A computational code in, exempli gratia, FORTRAN language is nothing but an ordered sequence of command lines and sometimes one or more subroutines or functions, which are also made of an ordered sequence of command lines. One begins defining a rule to obtain the adjoint instruction to a given command line, and, in the sequel, it is derived the adjoint instruction to the instruction resulting of two consecutive command lines. Let one consider the following FORTRAN command line,

$$Z=2*X+Y**3$$

which can be identified with the function

$$\begin{aligned} f: \mathbb{R}^2 &\rightarrow \mathbb{R} \\ (x, y) &\mapsto 2x + y^3 \end{aligned} \quad (11)$$

or even with the function

$$\begin{aligned} g: \mathbb{R}^3 &\rightarrow \mathbb{R}^3 \\ (x, y, z) &\mapsto (x, y, 2x + y^3), \end{aligned} \quad (12)$$

a very useful process in order to clarify the many stages in the work out of the adjoint instruction and its implementation, decreasing the possibility of errors when writing the computational code.

Although equation (11) and equation (12) are mathematically equivalents, notation in the latter is much more convenient to write the computational codes.

As g is a nonlinear operator, and nonlinear operators do not possess adjoint, when a command line involves one nonlinear operation, the first action to do is to take the linear version of the nonlinear declaration. This can be achieved simply taking the differential of the operator, or in matrix notation, much more useful for the adjoint process, considering the jacobian matrix of the operator. In the case of function g , its jacobian is given by

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3y^2 & 0 \end{pmatrix} \quad (13)$$

Then the linear of the instruction g, dg , is

$$\begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}_o = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3y_i^2 & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}_i \quad (14)$$

where the subscripts o and i denote, respectively, the output and the input amounts stored in the variables, just after and before the execution of the command line, that is,

$$\begin{cases} dx_o = dx_i \\ dy_o = dy_i \\ dz_o = 2dx_i + 3y_i^2 dy_i \end{cases} \quad (15)$$

Continuing to associate each instruction line of the code to a function, which means that a program is a composition of functions, so the linear version of the given program is obtained by using the Chain Rule of the Calculus, while the adjoint of a given code, by multiplying the reverse order the adjoint matrices that correspond to each command line. The processes, although simple, demands careful, since one outside variable in a command line can or not be defined using the same variable (in this case, seem as na inside variable), as it will be considered later.

Consider, for example, the following sequence of instructions

$$\begin{cases} Z = X + 2 * Y \\ W = Y + Z * * 2, \end{cases}$$

with the associated functions

$$\begin{aligned} f: \mathbb{R}^4 &\rightarrow \mathbb{R}^4 & g: \mathbb{R}^4 &\rightarrow \mathbb{R}^4 \\ (x, y, z, w) &\mapsto (x, y, x + 2y, w), & (x, y, z, w) &\mapsto (x, y, z, y + z^2). \end{aligned}$$

that we will refer as i_1 and i_2 , respectively.

To make clearer the processes of taking the linear (and later also the adjoint) of a given instruction, as well as the writing of the correspondent instructions in FORTRAN, one adopts the following convention: let E_i, E_{in} and E_o , respectively, the input, the intermediate and the output spaces, that is, the sequence $i_2 \circ i_1$, meaning, applying first i_1 and then i_2 , will represent by

$$\begin{aligned} f_1: E_i &\rightarrow E_{in} & f_2: E_{in} &\rightarrow E_o \\ (x, y, z, w) &\mapsto (x, y, x + 2y, w), & (x, y, z, w) &\mapsto (x, y, z, y + z^2), \end{aligned}$$

so that

$$i_2 \circ i_1 = f_2 \circ f_1(x, y, z, w) = f_2(x, y, x + 2y, w) = (x, y, x + 2y, y + (x + 2y)^2).$$

Therefore, writing $v_i \in E_i, v_{in} \in E_{in}$ and $v_o \in E_o$, and as its known the relations between them, given the perturbation data $d_i \in E_i, d_{in} \in E_{in}$ and $d_o \in E_o$, we have, by the Chain Rule, that

$$d(f_2 \circ f_1)(v_i) \cdot d_i = df_2(f_1(v_i)) \cdot df_1(v_i) \cdot d_i = df_2(v_{in}) \cdot d_{in} = d_o.$$

Computing the jacobian matrices of $df_2(v_{in})$ and $df_1(v_i)$, we obtain

$$df_2(v_{in}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 2(x + 2y) & 0 \end{pmatrix} \text{ and } df_1(v_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

So that the sequential instructions give $i_2 \circ i_1 = f_2 \circ f_1(x, y, z, w) = f_2(x, y, x + 2y, w) = (x, y, x + 2y, y + (x + 2y)^2)$. And in matricial form, the sequential instructions produce, being $d_i = (dx_i, dy_i, dz_i, dw_i)$, such that

$$d_i = (dx_i, dy_i, dz_i, dw_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 2(x+2y) & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

that is

$$\begin{cases} dx_o = dx_i \\ dy_o = dy_i \\ dz_o = dx_i + 2dy_i \\ dw_o = (2x_i + 4y_i)dx_o + (4x_i + 8y_i + 1)dy_i \end{cases}$$

Leaving the input and output indexes (which are not written in a real code) and discarding the unchanged variables, as $dx = dx$ and $dy = dy$, the linear instruction of instructions i_1 and i_2 above are the following FORTRAN instructions:

$$\begin{cases} dZ = dX + 2 * dY \\ dW = (2 * X + 4 * Y) * dX + (4 * X + 8 * Y + 1)dY, \end{cases}$$

as it was expected.

3.2 The Adjoint Code

The linear code is an intermediate step in order to obtain the adjoint of a given code. Considering again the sequence of i_1 and i_2 instructions, we have the following adjoint operators

$$[df_i(v_i)]^* = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ and } [df_2(v_{in})]^* = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2z_{in} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

As the adjoint of $d(f_2 \circ f_1)(dv_i)$, $[d(f_2 \circ f_1)]^*(dv_o)$, is given by

$$[d(f_2 \circ f_1)]^*(dv_o) = [df_1]^*(dv_{in}) \cdot [df_2]^*(dv_o)$$

Then, given the vector $dv_o = (dx_o, dy_o, dz_o, dw_o) \in E_o$, we obtains $(dx_o, dy_o, dz_o, dw_o) = [df_2(v_{in})]^*(dv_{in})$. Therefore,

$$\begin{aligned} [d(f_2 \circ f_1)]^*(dv_o) &= [df_1]^*(dv_i) \cdot [df_2]^*(dv_{in}) = \\ &= \begin{pmatrix} 1 & 0 & 1 & 2z_o \\ 0 & 1 & 2 & 4z_o \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} dx_o \\ dy_o \\ dz_o \\ dw_o \end{pmatrix} = \begin{pmatrix} dx_o + dz_o + 2z_o dw_o \\ dy_o + 2dz_o + 4z_o dw_o \\ 0 \\ 0 \end{pmatrix}, \end{aligned}$$

that is

$$\begin{cases} dx_i = dx_o + dz_o + 2z_o dw_o \\ dy_i = dy_o + 2dz_o + 4z_o dw_o. \end{cases}$$

Considering that $v_o = (x_i + 2y_i)$ and leaving the indexes, one obtains the adjoint instruction in FORTRAN as:

$$\begin{cases} dX = dX + dZ + 2 * (X + 2 * Y) * dW \\ dY = dY + 2 * dZ + 4 * (X + 2 * Y) * dW. \end{cases}$$

A more complex situation, demanding a careful scrutiny, arises when one of the variables appears in one command line as an output variable and as an input variable in the following command line, will be considered in the sequel. So, let i_3 and i_4 be, respectively, the instructions:

$$\begin{cases} X = X * Y * * 2 \\ Y = Y * * 3 * x * * 2. \end{cases}$$

Consider again the input, intermediate and output spaces, E_i , E_{in} and E_o , all of them isomorphs to \mathbb{R}^2 , and the functions representing the instructions i_3 and i_4

$$\begin{aligned} f_3 : E_i &\rightarrow E_{in} & f_4 : E_{in} &\rightarrow E_o \\ (x, y) &\mapsto (x, xy^2), & (x, y) &\mapsto (x, x^2y^3). \end{aligned}$$

We define the function associate with the sequential action of i_3 and i_4 instructions, which gives, for any $v_i \in E_i$,

$$\begin{aligned} f : E_i &\rightarrow E_o \\ v_i &\mapsto v_o = (f_4 \circ f_3)(v_i). \end{aligned}$$

And, applying the Chain Rule, we obtain, for a given vector $d_i = (dx_i, dy_i)$,

$$df(v_i) = df_4(f_3(v_i))(dv_i)df_3(v_i)(dv_i), \quad (16)$$

$$df_3(v_i) = \begin{pmatrix} 1 & 0 \\ y^2 & 2xy \end{pmatrix} \text{ and } df_4(f_3(v_i)) = \begin{pmatrix} 1 & 0 \\ 2x(xy^2)^3 & 3x^2(xy^2)^2 \end{pmatrix}. \quad (17)$$

Therefore, expression (16) is given by

$$\begin{pmatrix} 1 & 0 \\ 2x(xy^2)^3 & 3x^2(xy^2)^2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ y^2 & 2xy \end{pmatrix} \begin{pmatrix} dx_i \\ dy_i \end{pmatrix} = \begin{pmatrix} dx_o \\ dy_o \end{pmatrix},$$

that is

$$\begin{cases} dx_o = dx_i \\ dy_o = 5x_i^4 y_i^6 dx_i + 6x_i^5 y_i^5 dy_i. \end{cases}$$

So, in FORTRAN, the linear of the commands i_3 and i_4 is given by

$$\begin{cases} dY = Y * 2 * dX + 2 * X * Y * dY \\ Y = X * Y * 2 \\ dY = 2 * X * Y * 3 * dX + 3 * Y * 2 * X * 2 * dY \\ Y = Y * 3 * X * 2. \end{cases}$$

Now we can proceed to derive the adjoint instructions of i_3 and i_4 . Taking the adjoint of matrices in (17), we have

$$[df_4]^*(v_o) : E_o \rightarrow E_{in} \text{ and } [df_3]^*(v_{in}) : E_{in} \rightarrow E_i$$

such that

$$df_3(v_i) = \begin{pmatrix} 1 & y^2 \\ 0 & 2xy \end{pmatrix} \text{ and } df_4(f_3(v_i)) = \begin{pmatrix} 1 & 2x(xy^2)^3 \\ 0 & 3x^2(xy^2)^2 \end{pmatrix},$$

and, then, we obtain

$$\begin{pmatrix} 1 & y^2 \\ 0 & 2xy \end{pmatrix} \begin{pmatrix} 1 & 2x(xy^2)^3 \\ 0 & 3x^2(xy^2)^2 \end{pmatrix} \begin{pmatrix} dx_o \\ dy_o \end{pmatrix} = \begin{pmatrix} dx_i \\ dy_i \end{pmatrix},$$

that is, in FORTRAN language,

$$\begin{cases} I_0 = Y \\ Y = X * Y * 2 \\ I_1 = Y \\ Y = Y * 3 * X * 2 \\ \\ Y = I_1 \\ dX = dX + 2 * X * Y * 3 * dY \\ dY = 3 * Y * 2 * X * 2 * dY \\ \\ Y = I_0 \\ dX = dX + Y * 2 * dY \\ dY = 2 * Y * X * dY. \end{cases}$$

4. Results

4.1 A Very Simple Numerical Experiment

One studied the oil stain trajectory determination based on a *a posteriori* simulation of the Guanabara Bay accident in 2000. We use a number of 16 computer generated oil spot as the the observation within the time interval $[t_1, t_2]$, were $t_1 = 19 : 00h$ (January, 18) and $t_2 = 19 : 00h$ (January, 22).

The transport of the oil spot was described by the bi-dimensional advection-diffusion

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} + v \frac{\partial C}{\partial y} - D \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) = 0, \quad (18)$$

which is the universal transport model of oil in the sea surface (Lehr and Cekirge, 1980).

One has integrated then the Equation (18) using as initial condition the artificial oil spill (Figure 2), and, after the integration of Equation (18) (in $G(U)$), we storage these configurations as observations of the oil stain, for example, the configurations obtained at the times $t = 6 \times i, i = 1, \dots, 16$, which is one observation (for $t = 19 : 00h$, January, 19) and is shown in (Figure 3). After we performed the direct integration, we use a perturbed initial configuration, shown in (Figure 4), as new initial condition, integrate Equation (18), again using $G(U)$ routine, and we compare the discrepancies between the correspondents configurations, that is, "observational" configurations and new computer generated with the perturbed initial conditions at the same time in $J(G(U))$ routine. If the value measured after $J(G(U))$ routine is above the *a priori* tolerance, the main program calls the subroutine $[D(G(U))]$ *, in order to obtain the integration backward in time of the adjoint Equation (6), which gives the gradient of the functional J in relation to the initial conditions of the problem, which is then sent to the minimization subroutine (L-BFGS), giving a new initial configuration U which, after integration in $G(U)$, possible will decrease the value of $J(G(U))$. When the error generated by the two configurations, model generated and observations, measured in $J(G(U))$, is bellow the tolerance, the process of Data Assimilation stopped, producing the optimal initial condition (Figure 5).

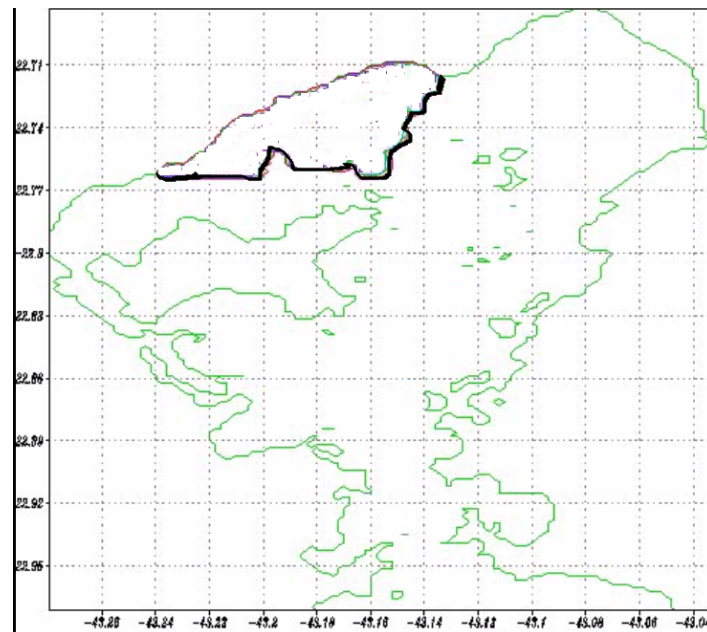


Figure 2. Guanabara Bay at instant $t = 19 : 00h$; January, 18

As we can see in Figure , the error between initial and retrieved configuration is acceptable.

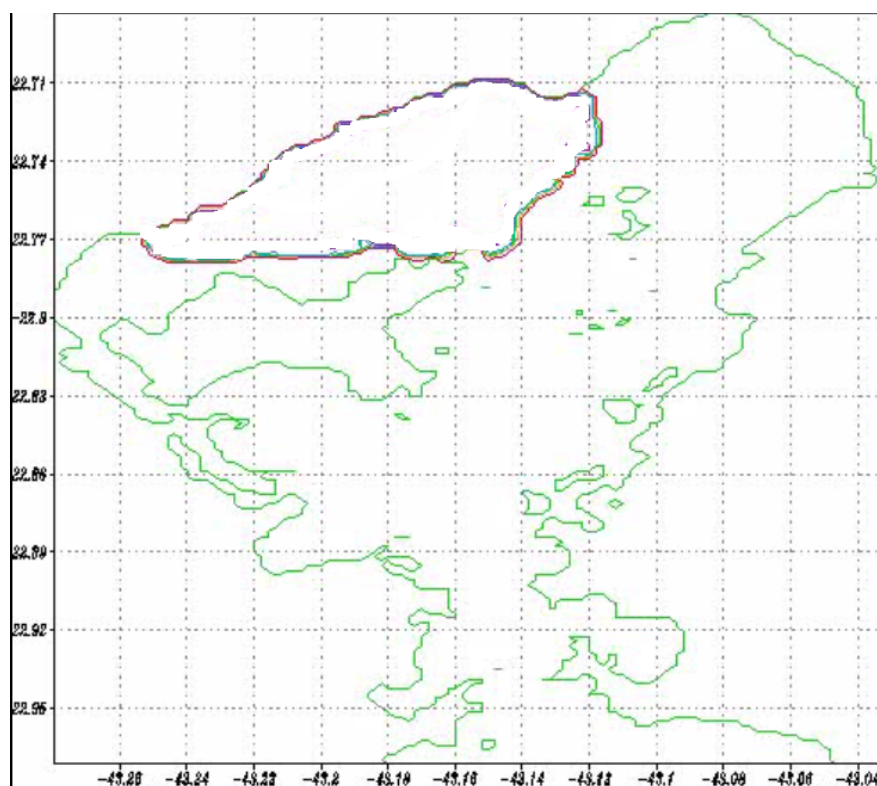


Figure 3. Guanabara Bay at instant $t = 19 : 00h$; January; 19

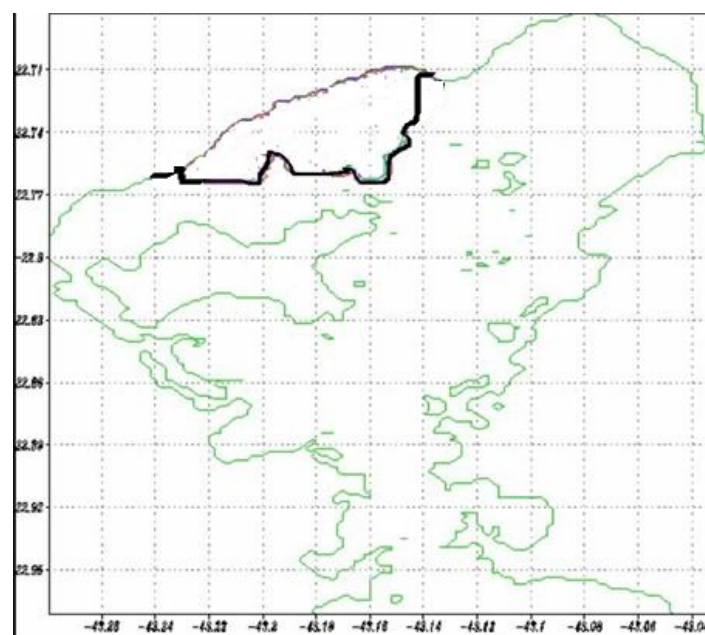


Figure 4. Configuration of Guanabara Bay at instant $t = 19 : 00h$; January; 18 perturbed

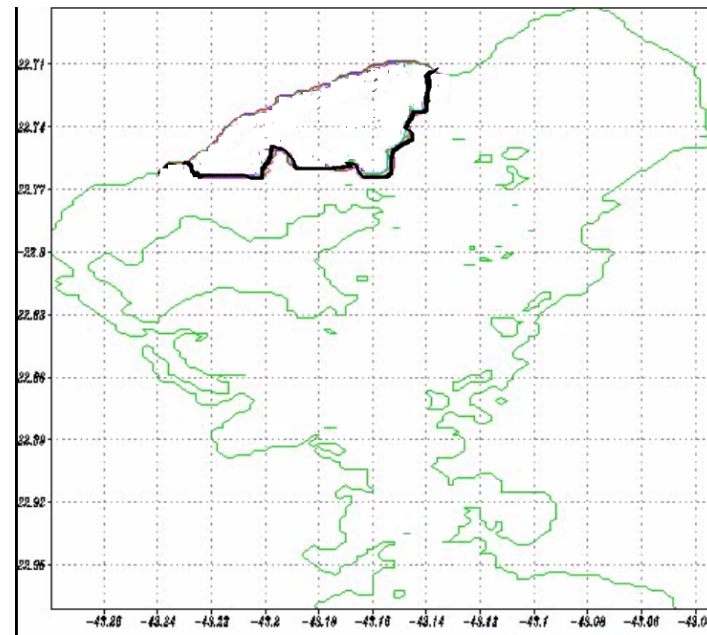


Figure 5. Initial configuration of Guanabara Bay, at instant $t = 19 : 00h$; January; 18, retrieved (compare with Figure 2)

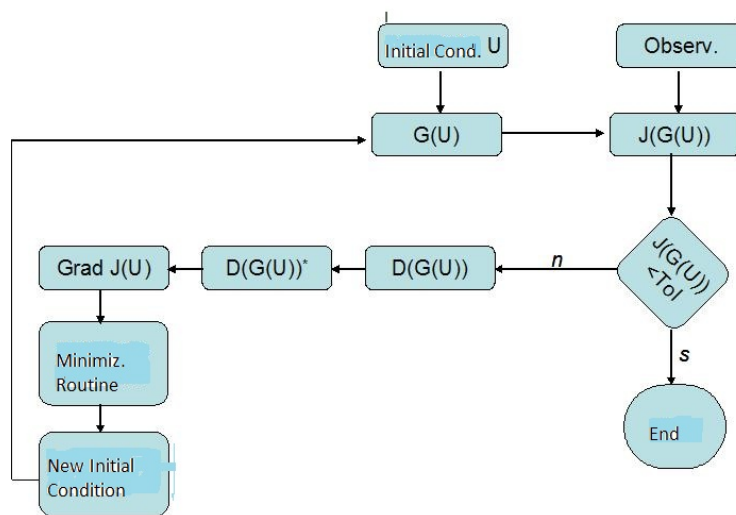


Figure 6. Flux of FORTRAN routines used in the simulations

5. Discussion

Despite the simplicity of the numerical experiment we realized, Data Assimilation, process in which observations of the studied systems are considered as conditions to the computational simulation must verify, together with Adjoint Equation Method, produce a methodology to obtain reliable and highly accurate computational configurations of a system at a feasible computational cost and within a time interval that allows an effective decision in an oil spill. A further question, also in the realm of the formalism presented here, is the optimal location of measurement instruments, enhancing an existing network.

Acknowledgements

The *a posteriori* simulation was performed by the LAMMA/UFRJ group, used here in order to obtain "observations" of the oil spillage in Guanabara Bay in 2000.

And now we present the scheme of the flux we have used in the simulation.

References

- Candy, J. V. (2009). *Bayesian Signal Processing: Classical, Modern and Particle Filtering Methods*, Wiley-Interscience.
- Faure, C., & Papegay, Y. (1998). *Odyssee User's Guide, Version 1.7*. INRIA, France.
- Kreyszig, E. (1989). *Introductory Functional Analysis with Applications*, John Wiley and Sons.
- Lehr, W. J., & Cekirge, H. M. (1980). Oil slick movements in the Arabian Gulf. In *Proceeds of Petroleum and Marine Environment*, Monaco.
- Liu, D., & Nocedal, D. (1989). On the limited memory BFGS method for large scale minimization. *Mathematical Programming*, 45, 503-528.
- Talagrand, O. (1997). Assimilation of observations, an introduction. *Journal of Meteorological Society of Japan*, 75(1B), 191-209.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).