Optimal Boundary Discretization by Variational Data Assimilation

Eugene Kazantsev

▶ To cite this version:

Optimal Boundary Discretization by Variational Data Assimilation.

Eugene Kazantsev

INRIA, projet MOISE, Laboratoire Jean Kuntzmann,
BP 53, 38041 Grenoble Cedex 9, France

SUMMARY

Variational data assimilation technique applied to the identification of the optimal discretization of interpolation operators and derivatives in the nodes adjacent to the boundary of the domain is discussed in frames of the linear shallow water model. The advantage of controlling the discretization of operators near boundary rather than boundary conditions is shown. Assimilating data produced by the same model on a finer grid in a model on a coarse grid, we have shown that optimal discretization allows us to correct such errors of the numerical scheme as under-resolved boundary layer and wrong wave velocity.
Copyright © 200 John Wiley & Sons, Ltd.

KEY WORDS: Variational Data Assimilation; Boundary conditions; Shallow water model, Boundary layer, Inertia-gravity waves.

1. Introduction

It is now well known, that even the best model is not sufficient to make a good forecast. Any model depends on a number of parameters, requires initial and boundary conditions and other data that must be collected and used in the model. However, interpolating or smoothing observed data is not the best way to incorporate these data into a model. Lorenz, in his pioneer work [1] has shown that a geophysical fluid is extremely sensitive to initial conditions. This fact requires to bring the model and its initial data together, in order to work with the couple "model-data", and to identify the optimal initial data for the model taking into account simultaneously the information contained in the observational data and in the equations of the model.

Optimal control methods [2] and perturbations theory [3] applied to the data assimilation technique ([4], [5]) show the way to do it. They allow to retrieve an optimal initial state for a given model from heterogeneous observation fields. Since the early 1990's, many mathematical and geophysical teams have been involved in the development of the data assimilation strategy. One can cite many papers devoted to this problem, as in the domain of development of different techniques for the data assimilation and in the domain of its applications to the atmosphere and oceans.

However, overwhelming majority of data assimilation methods are now intended to identify and reconstruct an optimal initial state for the model. Since Lorenz [1], who has pointed out the importance of precise knowledge of the starting point of the model, essentially the starting point is considered as the control parameter and the target of data assimilation.

Of course, the model's flow is extremely sensitive to its initial point. But, it is reasonable to suppose that a geophysical model is also sensitive to many other parameters like bottom topography, boundary conditions on rigid and open boundaries, forcing fields and friction coefficients. All these parameters and values are also extracted in some way from observational data, interpolated to the model's grid and can neither be considered as exact, nor as optimal to the model. On the other hand, due to non-linearity and intrinsic instability of model's trajectory, its sensitivity to all these external parameters may also be exponential.

Numerous studies show strong dependence of the model's flow on the boundary data ([6], [7]), on the representation of the bottom topography ([8], [9], [10]), on the wind stress ([11], [12]), on diffusivity coefficients ([13]) and on fundamental parametrization like Boussinesq and hydrostatic hypotheses [14]. However, few papers are devoted to the development of data assimilation techniques intended to identify and to control these internal model's parameters. One can cite several attempts to use data assimilation in order to identify the bottom topography of simple models ([15], [16]) and in order to control open boundary conditions in coastal and regional models ([17], [18], [19]). Boundary conditions on rigid boundaries have been controlled by data assimilation for heat equation (see for example [20], [21]), but this control concerns the linear parabolic diffusion operator rather than hyperbolic transport and advection operators that are more important in geophysical models.

Studies of the possibility to control boundary conditions on rigid boundaries for equations containing hyperbolic operators can be found in [22] on the example of non-linear balance equation, in [23] on the example of the wave equation and in [24] and [25] on the example of the Burgers equation. The principal possibility to improve the model's solution controlling it's boundary values are shown in all these papers. However, as it has been noted in [24], particular attention must be paid to the discretization process which must respect several rules because it is the discretization of the model's operators takes into account the set of boundary conditions and introduces them into the model. Consequently, instead of controlling boundary conditions themself, it may be more useful to identify optimal discretization of differential operators in points adjacent to boundaries. This allows us to control directly the way the boundary conditions influence the model and to control boundary parameters in a more general way. In [23], for example, it was shown that deplacing the boundary helps to correct the error occured due to a wrong wave velocity. Adjustment of the boundary position is, however, only possible when the coefficients used in approximations of the derivatives are controlled directly by data assimilation. Boundary conditions participate in discretized operators, but considering the discretization itself, we take into account additional parameters like the position of the boundary, lack of resolution of the grid, etc.

Although the boundary configuration of the ocean is steady and can be measured with much better accuracy than the model's initial state, it is not obvious how to represent it on the model's grid because of limited resolution. The coastal line of continents possesses a very fine structure and can only be roughly approximated by the model's grid. Consequently, boundary conditions are defined at the model grid's points which are different from the coast. Even the most evident impermeability condition being placed at a wrong point may lead to some error in the model's solution. We should accept the flux to be able to cross the boundary

in places where the boundary is in water, prescribing some integral properties on the flux.

Moreover, ocean models frequently include strong and thin boundary currents with intense velocity gradients. In this case particular attention must be payed to the approximation of the boundary layer because wrong representation of these currents may be responsible for drastic deformations in a global solution (see, for example [6]). This may lead us to control the discretization of the model's operators in the whole boundary layer rather than in adjacent to boundary nodes only.

In this paper we use 4D-Var data assimilation to control the discretization of derivatives and interpolation operators in the boundary regions. The development of the data assimilation is illustrated on the example of the linear shallow water model on the Arakawa's C-grid. The simplicity of the model allows us to clearly see technical points of the development (like the algorithm of differentiation and development of the adjoint equation) without being overwhelmed by complexity of operators and grids. The purpose of the paper is to study the possibility to control the numerical scheme by data assimilation and the particularities of this type of control in view to develop and use the data assimilation to identify optimal numerical scheme in coastal regions of ocean models.

The paper is organized as follows. The second section describes the model, its adjoint and the data assimilation procedure. The third section is devoted to the stationary solution of the model that includes the Munk boundary layer. And the fourth section discusses the data assimilation in the case of inertia-gravity and Rossby waves.

2. Linear Shallow Water Model

2.1. Model's equations and discretization

In order to test the data assimilation procedure we consider the linear shallow water model on the β-plane [26], [27]:

$$
\begin{aligned}
\frac{\partial u}{\partial t} - (f_0 + \beta y)v &= -g\frac{\partial \eta}{\partial x} - \sigma u + \mu \Delta u + \frac{\tau_x}{\rho_0 H_0} \\
\frac{\partial v}{\partial t} + (f_0 + \beta y)u &= -g\frac{\partial \eta}{\partial y} - \sigma v + \mu \Delta v + \frac{\tau_y}{\rho_0 H_0} \\
\frac{\partial \eta}{\partial t} + H_0\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) &= 0
\end{aligned}
\tag{1}
$$

where $u(x,y,t)$ and $v(x,y,t)$ are two velocity components, $\eta(x,y,t)$ is the sea surface elevation, ρ_0 is the mean density of water, H_0 is the characteristic depth of the basin and g is the reduced gravity. The model is driven by the surface wind stress with components $\tau_x(x,y)$ and $\tau_y(x,y)$ and subjected to the bottom drag that is parametrized by linear terms σu, σv and to the horizontal eddy diffusion parametrized by an harmonic operator $\mu\Delta u$ and $\mu\Delta v$. Coriolis parameter is supposed to be linear in y coordinate and is presented as $(f_0 + \beta y)$. The system is defined in a square box of side L with boundary conditions prescribed for u and v. We require that both u and v vanish on the whole boundary. No boundary conditions is prescribed to η.

We discretize all variables of this equation on the regular Arakawa's C-grid with constant
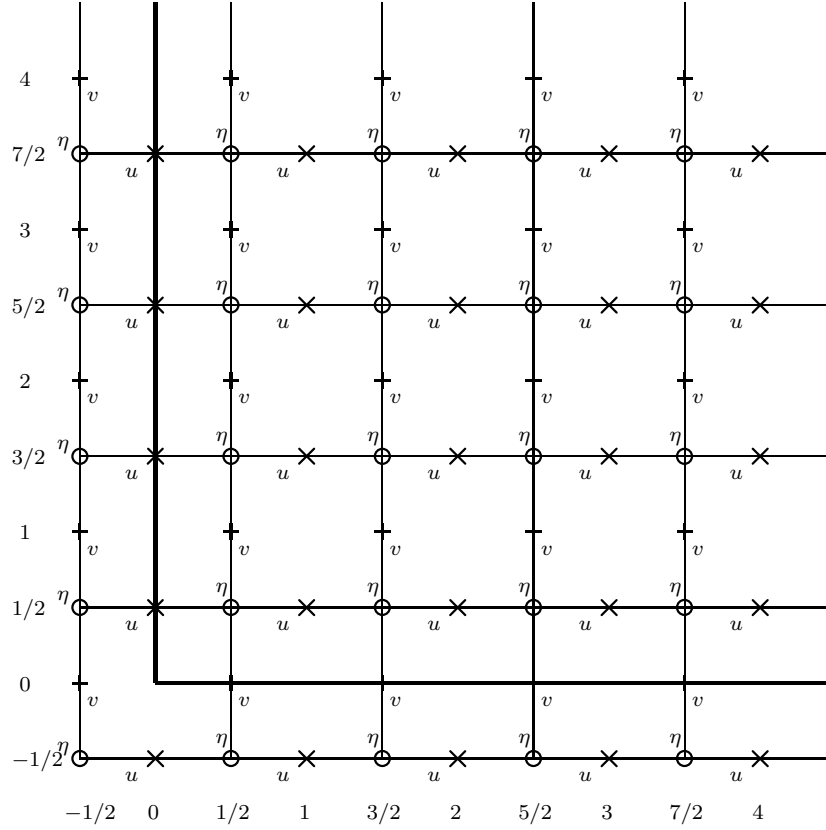
Figure 1. Arakawa C-grid

grid step $h = \frac{L}{N}$ in both x and y directions (see fig.1)

$$
\begin{aligned}
u_{i,j-1/2} &= u(ih, jh - h/2) \text{ for } i = 0, \dots N, j = 0, \dots, N+1 \\
v_{i-1/2,j} &= v(ih - h/2, jh) \text{ for } i = 0, \dots N+1, j = 0, \dots N \\
\eta_{i-1/2,j-1/2} &= \eta(ih - h/2, jh - h/2) \text{ for } i = 0, \dots N+1, j = 0, \dots N+1
\end{aligned}
$$

The system (1) contains several spatial operators that must be approximated on this grid. First of all, this is the divergence operator in the third equation. We shall note the discretization of derivatives in the divergence by $D_x u$ and $D_y v$. Second, the gradient operator that is present in the first two equations is denoted by $G_x \eta, G_y \eta$. Third, two interpolation operators are necessary to calculate variables u and v at points where v and u are defined respectively. These operators are composed of two subsequent interpolations: $S_x u$ and $S_y v$ that provide their results at points where η is defined, and $S_y \eta$ and $S_x \eta$ that give the resulting interpolation

at required points. It should be noted that operators $S_y\eta$ and $S_x\eta$ are never applied to the variable η itself, but to the results of interpolation $S_x u$ and $S_y v$ which are defined at the same points as η. And finally, the fourth operator is the discrete Laplacian that is denoted by Δ^h.

Using these notations, the discretized system (1) can be written

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= (f_0 + \beta y)S_x S_y v - gG_x\eta - \sigma u + \mu\Delta^h u + \frac{\tau_x}{\rho_0 H} \\
\frac{\partial v}{\partial t} &= -(f_0 + \beta y)S_y S_x u - gG_y\eta - \sigma v + \mu\Delta^h v + \frac{\tau_y}{\rho_0 H} \\
\frac{\partial \eta}{\partial t} &= -H_0(D_x u + D_y v)
\end{aligned}
\tag{2}
$$

We define discretized operators $D_x u, D_y v, G_x\eta, G_y\eta, S_x u, S_y v, S_y\eta$ and $S_x\eta$ at all internal points of the domain as linear combination of the variable's values at four adjacent grid points. For example, the derivative and the interpolation of the variable u defined at corresponding points writes

$$
(D_x u)_{i-1/2, j-1/2} = \frac{1}{h}\sum_{k=-2}^{1} a_k^D u_{i+k, j-1/2} \text{ for } i = M+1, \ldots, N-M
$$

$$
\tag{3}
$$

$$
(S_x u)_{i-1/2, j-1/2} = \frac{1}{h}\sum_{k=-2}^{1} a_k^S u_{i+k, j-1/2} \text{ for } i = M+1, \ldots, N-M
$$

That means the derivative at the point $i-1/2$ is presented as linear combination of four values of u taken at points from $i-2$ to $i+1$ with coefficients a_k^D.

Coefficients a_k are supposed to be known because we intend to control the approximations near the boundary only. In this paper, we use either the sequence $a_k^D = \frac{1}{h}(0, -1, 1, 0)$ for $k = -2, -1, 0, 1$, or $a_k^D = \frac{1}{24h}(1, -27, 27, -1)$. One can easily see that corresponding approximations of derivatives are of the second and of the fourth order

$$
\frac{u_i - u_{i-1}}{h} = \left(\frac{\partial u}{\partial x}\right)_{i-1/2} + \frac{h^2}{24}\left(\frac{\partial^3 u}{\partial x^3}\right)_{i-1/2} + O(h^3)
$$

$$
\tag{4}
$$

$$
\frac{u_{i-2} - 27u_{i-1} + 27u_i - u_{i+1}}{24h} = \left(\frac{\partial u}{\partial x}\right)_{i-1/2} - \frac{3h^4}{640}\left(\frac{\partial^5 u}{\partial x^5}\right)_{i-1/2} + O(h^5)
$$

To discretize interpolation operators, we use either $a_k^S = (0, 1/2, 1/2, 0)$ for $k = (-2, -1, 0, 1)$, or $a_k^S = \frac{1}{8}(-1, 9, 9, -1)$ that also provide the second and the fourth order interpolation of the variable.

All other operators are discretized similarly at all internal points. All derivatives use the same sequence a_k^D and all interpolation operators are discretized with the same sequence a_k^S. We shall note in each particular case whether the second or the fourth order scheme is used.

However, operators D and S are not defined in the boundary region that is considered as a band of M grid nodes width that follows the boundary. Discretizations of operators in this

band are supposed to be different from (3) and represent the control variables in this study. Moreover, schemes (3) can not be used at all for the fourth order approximation because they require function's values beyond the boundary. We can, of course, extrapolate variables beyond the domain with the necessary order and substitute extrapolated values in (3), but it is not obvious what extrapolation formula is the best for this purpose. So, in order to obtain an optimal boundary approximation assimilating external data, we suppose nothing about derivatives near the boundary and write them in a general form

$$(D_x u)_{m-1/2,j-1/2} = \frac{1}{h} \left(\alpha_{0,m}^{D_x u} + \sum_{k=1}^{K} \alpha_{k,m}^{D_x u} u_{k-1,j-1/2} \right) \text{ for } m = 1, 2, \ldots, M \qquad (5)$$

This formula represents also a linear combination of values of u at K points adjacent to the boundary, but with the coefficients $\alpha \neq a$ which are considered as particular for each operator and for each variable. The constant $\alpha_0^{D_x u}$, which is also particular for each operator, may be added in some cases to simulate non uniform boundary conditions like $u(0, y) = \alpha_0^{D_x u} \neq 0$. All coefficients α are considered as control parameters in this study. They are allowed to vary in the variational assimilation procedure intended to identify their optimal set. The number of points K used in linear combination is not fixed. We keep the possibility to vary K and to study its influence on the assimilation results. In addition to this, we can control derivative and interpolation operators at M points adjacent to the boundary. The vicinity of the boundary is not restricted to just one point, we use the parameter M to define how many points near the boundary will be controlled by the algorithm.

Moreover, we can require looking for particular α for each point near the boundary. That means we may allow the dependence of α_0, $\alpha_{k,m}$ on the latitude j in the expression (5). The question of using either uniform coefficients or particular α for each latitude will be discussed later in this paper.

Here, we can emphasize the choice of controlling the numerical scheme in the boundary region rather than boundary conditions. The general form of boundary conditions that may be prescribed for u variable in this model is

$$u(t) \mid_{boundary} -r_1 \frac{\partial u}{\partial x}(t) \bigg|_{boundary} = r_2. \qquad (6)$$

We can not write more complex boundary conditions (with second derivatives, for example) because we obtain a system with no solution at all. Consequently, we can control only two parameters, r_1 and r_2. It may be sufficient in particular cases, but, as we shall see further, is not sufficient in general. Controlling all coefficients of the numerical scheme (5), we are free to choose as many $\alpha_{k,m}$ as we need defining appropriate values of parameters K and M.

We distinguish α for different variables and different operators allowing different controls of derivatives because of the different nature of these variables and different boundary conditions prescribed for them. It is obvious, for example, that the approximation of the derivative of η in the gradient may differ from the approximation of the derivative of u in the divergence. Although both operators represent a derivative, boundary conditions for u and η are different, these derivatives are defined at different points, at different distance from the boundary. Consequently, it is reasonable to let them be controlled separately and to assume that their optimal approximation may be different with distinct coefficients $\alpha^{D_x u}$ and $\alpha^{D_x \eta}$.

Interpolations and derivatives on the opposite side of the square are also assumed to be different. They are calculated with coefficients $\tilde{\alpha}$ which are also considered as unknown control

parameters:

$$(D_x u)_{N-m+1/2, j-1/2} = \frac{1}{h}\left(\tilde{\alpha}_{0,m}^{D_x u} + \sum_{k=0}^{K} \tilde{\alpha}_{k,m}^{D_x u} u_{N-k+1/2, j-1/2} \right) \text{ for } m = 1, 2, \ldots, M$$

Time stepping of this model is performed by the leap-frog scheme. The first time step is splitted into two Runge-Kutta stages in order to ensure the second order approximation.

2.2. Tangent and adjoint models

The approximation of the derivative introduced by (3) and (5) depends on control variables α. Operators are allowed to change their properties near boundaries in order to find the best fit with requirements of the model and data. To assign variables α we shall perform data assimilation procedure and find their optimal values. Variational data assimilation is usually performed by minimization of the specially introduced cost function. The minimization is achieved using the gradient of the cost function that is usually determined by the run of the adjoint to the tangent linear model.

We start from development of the tangent linear model that is equal to the Gateaux derivative of the original model (2) with respect to the control parameters. To calculate this derivative we suppose that the control variables can have small variations and we determine how these variations will perturb the model's solution. Thus, we suppose that all α are replaced by some $\alpha + \delta\alpha$ such that $\|\delta\alpha\| << \|\alpha\|$. Let the model with $\alpha + \delta\alpha$ have a new solution $(u + \delta u, \ v + \delta v, \ \eta + \delta\eta)$. We get the equation for $\delta u, \ \delta v, \ \delta p$ as the difference between the model with coefficients $\alpha + \delta\alpha$ and the model with coefficients α. To obtain this equation, we introduce variations of operators. Using the gradient G as an example, we denote the difference $G(\alpha + \delta\alpha) - G(\alpha)$ by δG and get

$$G(\alpha + \delta\alpha)(\eta + \delta\eta) - G(\alpha)\eta = (\delta G)\eta + G\delta\eta + (\delta G)\delta\eta$$

As it was defined, G is a finite difference approximation of a derivative. This matrix is block-bidiagonal (quadridiagonal if the fourth order scheme is used), i.e. each block is composed of two or fourth diagonals filled by coefficients a_k^D from (3). However, M lines in the beginning and M lines at the end of each block have a different structure. They are intended to approximate the gradient near the boundary and count each K non-zero elements α as defined in (5). It is these lines only that depend on the control parameter and there are non-zero elements $\delta\alpha$ in the matrix δG on these lines only. All other lines of δG are filled by zeroes because the approximation of the derivative in the interior part of the domain does not depend on the controls α.

So far, both δG and $\delta\eta$ are supposed to be small, we neglect their product in the right-hand-side of this equation. Variation of the divergence operator is written in a similar way with already neglected nonlinear term:

$$D_x(\alpha + \delta\alpha)(u + \delta u) - D_x(\alpha)u = (\delta D_x)u + D_x \delta u$$

The interpolation operator in (2) is composed by two successive interpolations. The linear part of it's variation counts, consequently, three terms

$$S_x(\alpha + \delta\alpha)S_y(\alpha + \delta\alpha)(v + \delta v) - S_x(\alpha)S_y(\alpha)v = (\delta S_x)S_y v + S_x(\delta S_y)v + S_x S_y \delta v$$

All these matrices (δS_x, δS_y, δD_x, δD_y etc.) have a similar structure. Their only non-zero elements are situated in the beginning and at the end of each block.

Combining variances of approximations of all differential and interpolation operators, we write the system that governs the evolution of perturbations of u, v and η:

$$
\begin{aligned}
\frac{\partial \delta u}{\partial t} &= \underline{(f_0 + \beta y)S_x S_y \delta v - gG_x \delta \eta - \sigma \delta u + \mu \Delta^h \delta u +} \\
&+ \underline{\underline{(f_0 + \beta y)(\delta S_x S_y v + S_x \delta S_y v) - g\delta G_x \eta}} \\
\frac{\partial \delta v}{\partial t} &= \underline{-(f_0 + \beta y)S_y S_x \delta u - gG_y \delta \eta - \sigma \delta v + \mu \Delta^h \delta v -} \\
&- \underline{\underline{(f_0 + \beta y)(\delta S_y S_x u + S_y \delta S_x u) - g\delta G_y \eta}} \\
\frac{\partial \delta \eta}{\partial t} &= \underline{-H_0(D_x \delta u + D_y \delta v)} - \underline{\underline{H_0(\delta D_x u + \delta D_y v)}}
\end{aligned}
\tag{7}
$$

with the same zero boundary conditions for δu and δv as for u and v. No boundary conditions is prescribed for $\delta \eta$ as well as for η. At initial time both $\delta u, \delta v$ and $\delta \eta$ vanish because our study is confined at the control of the discretization of operators near the boundary only rather than joint control of a boundary scheme and initial conditions of the model. We are interested in the evolution of a pure perturbation due to boundary scheme.

We have intentionally distinguished two types of operators in the right-hand-side of (7) (underlined by one or two lines). Operators of the first type act in the space of perturbations of u, v and η. Both the argument and the result of these operators are in the space of δu, δv, $\delta \eta$. Operator G_x, for example, brings the variable $\delta \eta$ to the space of δu, operator $S_y S_x$ transforms the variable δu to the space of δv, etc.

The second type of operators distinguished in (7) (underlined by two parallel lines) are of implicit structure. Their argument in the present form is contained in the matrix itself. Let us consider, for example, the expression $\delta G_x \eta$. True argument of this expression is contained in the matrix δG_x rather than in the vector η. In this form, it is the matrix that depends on the variations of the control parameter $\delta \alpha$. The vector η, that can be wrongly considered as an argument in this expression, is the solution of the original shallow-water model (1). This vector depends implicitly on the control parameters α, but does not depend on variations $\delta \alpha$.

Expressions like $\delta G_x \eta$ are not convenient to carry out further development. Writing an adjoint operator, we would better have a constant operator (which does not depend on $\delta \alpha$) multiplied by a variable vector (which depends on $\delta \alpha$). It would be more convenient to rewrite products like this in a more explicit form

$$
\delta G_x \eta = \widehat{\eta} \vec{\delta \alpha}
$$

where the operator $\hat{\eta}$ is constructed from the solution η of the original equation and the vector $\vec{\delta \alpha}$ is extracted from the matrix δG as it is shown in [23]. All other products of the second type ($\delta D_x u$ $\delta D_y v$, $\delta S_x u$, etc) are also rewritten in this, more explicit form. We shall further use hats to denote matrices that have been constructed from vectors. These matrices are also block-matrices. All elements of their blocks are equal to zero except M lines in the beginning and M lines at the end of each block. These lines are composed of K values of corresponding vector, and, namely, values of approximated function in K nodes near the boundary.

In order to write the tangent model in a matrix form we add one additional equation to the system (7) expressing that control coefficients α are stationary: $\frac{\partial \delta \alpha}{\partial t} = 0$. Using these notations we write:

$$\frac{\partial}{\partial t} \begin{pmatrix} \delta u \\ \delta v \\ \delta \eta \\ \delta \alpha \end{pmatrix} = \begin{pmatrix} -\sigma + \mu \Delta^h & (f_0 + \beta y) S_x S_y & -g G_x & R_u \\ -(f_0 + \beta y) S_y S_x & -\sigma + \mu \Delta^h & -g G_y & R_v \\ -H_0 D_x & -H_0 D_y & 0 & R_\eta \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \\ \delta \eta \\ \delta \alpha \end{pmatrix} \qquad (8)$$

where

$$\begin{aligned} R_u \delta \alpha &= (f_0 + \beta y)(\delta S_x S_y v + S_x \delta S_y v) - g \delta G_x \eta = [(f_0 + \beta y)(\widehat{S_y v} + S_x \widehat{v}) - g\widehat{\eta}]\delta \alpha \\ R_v \delta \alpha &= -(f_0 + \beta y)(\delta S_y S_x u + S_y \delta S_x u) - g \delta G_y \eta = [-(f_0 + \beta y)(\widehat{S_x u} + S_y \widehat{u}) - g\widehat{\eta}]\delta \alpha \\ R_\eta \delta \alpha &= [-H_0(\delta D_x u + \delta D_y v) = -H_0(\widehat{u} + \widehat{v})]\delta \alpha \end{aligned} \qquad (9)$$

It has to be noted, that operators R_u, R_v and R_η act from the space of the control variable α to the space ofvariations of the model's solution δu, δv or $\delta \eta$. Their matrices, consequently, are rectangular. The number of lines in their matrices corresponds to dimensions of variables in physical space, i.e. $N \times (N - 1)$ for u and v and $(N - 1)^2$ for η. The number of columns corresponds to the dimension of the control variable's space or to the total number of control coefficients α. This number is equal to the number of controlled operators (eight in the present configuration) multiplied by the number of boundaries to control for each operator (usually two: either East-West if the operator is in x direction, or North-South if the operator acts in y) and multiplied by the number of control coefficients for each boundary $(M \times (K + 1)$ as defined by (5)). The smallest number of control coefficients in the present configuration is equal to 48 $(K = 2,\ M = 1)$. Of course, this small number is not always sufficient. In general case, we have to use more controls to get better results, but in any case the number of control coefficients is much less than the dimension of the variables space.

This fact may be taken into account in the analysis of the complexity of variational data assimilation in this case and in the making choice of the practical way of calculation of the cost function. We can see, the matrix of the tangent linear model (8) is composed by two parts: the 3×3 block composed of operators acting in the space of the model's variables and the fourth column composed of operators R. The 3×3 block is responsible for the evolution of a small perturbation by the model's dynamics, while the column determines the way how this perturbation is introduced into the model. The first term is similar for any data assimilation, while the second one is specific to the particular variable under control. This term is absent when the goal is to identify the initial conditions of the model because the uncertainty in initial conditions is introduced only once, at the beginning of the model integration. But, when the uncertainty is presented in an internal model parameter, like in this case, the perturbation is introduced at each time step of the model.

So, if we intend to identify an optimal boundary parametrization for a model with an existing adjoint developed for data assimilation and identification of its initial state, we can use this adjoint as 3×3 block because this part is common for any data assimilation. However, the fourth column of the matrix (8) must be developed from the beginning because it is specific to the particular control parameter. This development may be technically difficult. One can see, writing adjoint of operators R (9) is as complex as the adjoint of the 3×3 block. If we add nonlinear terms to the model, writing the fourth column will become even more complex

than the development of the 3×3 block. Including, for example, $v \frac{\partial u}{\partial y} = S_x S_y v \ S_y D_y u$ term in the first model's equation will add six additional terms in the tangent model: two terms in the 3×3 block and four terms in the fourth column.

Regarding the supplementary work to be done developing the tangent and adjoint models and taking into account small number of control parameters, it may be reasonable to try to calculate the gradient of the cost function by some other method beginning with the simplest finite difference method. Of course, this will be more expensive computationally, but the gain in the development procedure may compensate this excessive computational cost.

Moreover, boundary conditions and discretizations of operators near boundary are internal model's parameters. They must be identified once for all model runs, while initial conditions are external parameters and must be identified for each particular model run. Consequently, we have to identify the discretization near boundary on the stage of the development of the model while choosing all other internal parameters. Hence, we can accept more time consuming data assimilation procedure under condition that it requires less time for it's development.

However, in this paper we proceed with the calculation of the gradient of the cost function using the adjoint of the tangent linear model (8). In order to develop the adjoint model, we need to introduce the scalar product in the space defined by tangent model. Each element in this space is composed of discretized functions u, v and η and also the whole set of the control coefficients α. A vector in this space has four components $\phi = (u, v, \eta, \alpha)$.

In this paper we consider a weighted Euclidien scalar product in this space

$$
\ll \phi, \phi^* \gg \;\; = \;\; \ll \begin{pmatrix} u \\ v \\ \eta \\ \alpha \end{pmatrix}, \begin{pmatrix} u^* \\ v^* \\ \eta^* \\ \alpha^* \end{pmatrix} \gg =
\tag{10}
$$

$$
= \;\; w_u^2 \sum_{i,j} u_{i,j} u_{i,j}^* + w_v^2 \sum_{i,j} v_{i,j} v_{i,j}^* + w_\eta^2 \sum_{i,j} \eta_{i,j} \eta_{i,j}^* + \sum_{k,m,operator} \alpha_{k,m}^{operator} (\alpha_{k,m}^{operator})^*
$$

The sums in the scalar product is performed over all nodes i, j of the grid of all model's variables u, v and η. The sum of control coefficients α is performed over all $k, m : 1 \leq k \leq K$, $1 \leq m \leq M$ (5) and over all operators "*operator*" controlled by these coefficients. Weights $w_u = w_v = \frac{1}{\sqrt{gH_0}}$ and $w_\eta = \frac{1}{H_0}$ are introduced to bring all variables to dimensionless form.

Denoting the matrix of the tangent model (8) by A, we write it in a short form:

$$
\frac{\partial \delta \phi}{\partial t} - A \delta \phi = 0, \quad \delta \phi \mid_{t=0} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \delta \alpha \end{pmatrix}
$$

Following [3], [5], we multiply the tangent model by some $\phi^*(t)$ and integrate this product in time from 0 to some T.

$$
0 \;\; = \;\; \int_0^T \ll (\frac{\partial \delta \phi}{\partial t} - A \delta \phi), \phi^*(t) \gg dt = \int_0^T \ll \frac{\partial \delta \phi}{\partial t}, \phi^*(t) \gg dt - \int_0^T \ll A \delta \phi, \phi^* \gg dt =
$$

$$
= \;\; \ll \delta \phi(T), \phi^*(T) \gg - \ll \delta \phi(0), \phi^*(0) \gg + \int_0^T \ll \delta \phi(t), -\frac{\partial \phi^*}{\partial t} - A^* \phi^* \gg dt
\tag{11}
$$

where A^* is the adjoint matrix to A. Consequently, if ϕ^* satisfies the adjoint model $-\frac{\partial \phi^*}{\partial t} - A^*\phi^* = 0$, then we have an equality

$$\ll \delta\phi(T), \phi^*(T) \gg = \ll \delta\phi(0), \phi^*(0) \gg \qquad (12)$$

that we shall use to calculate the gradient of the cost function.

Complete matricial formulation of the adjoint model, hence, writes:

$$-\frac{\partial}{\partial t}\begin{pmatrix} u^* \\ v^* \\ \eta^* \\ \alpha^* \end{pmatrix} = \begin{pmatrix} -\sigma + \mu\Delta^h & (f_0 + \beta y)S_x^* S_y^* & -H_0 D_x^* & 0 \\ -(f_0 + \beta y)S_y^* S_x^* & -\sigma + \mu\Delta^h & -H_0 D_y^* & 0 \\ gG_x^* & gG_y^* & 0 & 0 \\ R_u^* & R_v^* & R_\eta^* & 0 \end{pmatrix}\begin{pmatrix} u^* \\ v^* \\ \eta^* \\ \alpha^* \end{pmatrix} \qquad (13)$$

where

$$\begin{aligned} R_u^* &= (f_0 + \beta y)(\widehat{S_y v}^* + \widehat{v}^* S_x^*) + g\widehat{\eta}^* \\ R_v^* &= -(f_0 + \beta y)(\widehat{S_x u}^* + \widehat{u}^* S_y^*) + g\widehat{\eta}^* \\ R_\eta^* &= -H_0(\widehat{u}^* + \widehat{v}^*) \end{aligned} \qquad (14)$$

The result of the adjoint model run $\phi^*(0)$, that is used in (12), we shall denote as

$$\phi^*(0) = \mathcal{A}^*(T, 0)\phi^*(T) \qquad (15)$$

in order to explicitly show that it was obtained by the adjoint model (13) that starts at time $t = T$ from the state $\phi^*(T)$ and is integrated back in time to $t = 0$.

2.3. Cost function

One of principal purposes of variational data assimilation consists in the variation of control parameters in order to bring the model's solution closer to the observational data. This implies the necessity to measure the distance between the trajectory of the model and data. Introducing a cost function, we define this measure. Generally speaking, the cost function is represented by some norm of the difference between model's solutions and observations, eventually accompanied by some regularization term.

In this paper we use the Euclidean norm because in this preliminary study we are interested essentially in the principal possibility to control the boundaries by data assimilation, rather than to the difficulties that arise due to imperfect data. Consequently, all the artificial observations to be assimilated into the model are considered as perfect, known at any time and at any point for all variables u, v and η, and no regularization is added to the cost function. To characterize the difference between the model's solution and the observational data we use

$$\xi^2 = w_u^2 \sum_{i,j}(u_{i,j} - u_{i,j}^{obs})^2 + w_v^2 \sum_{i,j}(v_{i,j} - v_{i,j}^{obs})^2 + w_\eta^2 \sum_{i,j}(\eta_{i,j} - \eta_{i,j}^{obs})^2 \qquad (16)$$

The norm ξ is, in fact, associated with the scalar product (10) with the fourth component α equal to zero. Expressing ξ in terms of the scalar product, we emphasize its dependence on time and control coefficients α:

$$\begin{aligned} \xi^2 &= \xi^2(\alpha, t) = \ll \phi(\alpha, t) - \phi^{obs}(t), \phi(\alpha, t) - \phi^{obs}(t) \gg = \\ &= \ll \begin{pmatrix} u(\alpha, t) - u^{obs}(t) \\ v(\alpha, t) - v^{obs}(t) \\ \eta(\alpha, t) - \eta^{obs}(t) \\ 0 \end{pmatrix}, \begin{pmatrix} u(\alpha, t) - u^{obs}(t) \\ v(\alpha, t) - v^{obs}(t) \\ \eta(\alpha, t) - \eta^{obs}(t) \\ 0 \end{pmatrix} \gg \end{aligned} \qquad (17)$$

The choice of the cost function for variational data assimilation performed to identify discretizations of model's operators near boundaries is not obvious. If the purpose is to identify the initial conditions of the model, it is reasonable to choose a classical 4D-VAR cost function

$$\mathcal{I}_4(\alpha) = \int\limits_0^T \xi^2(\alpha,t)dt \qquad (18)$$

that gives the same importance to the difference ξ^2 at any time. However, assimilating data with the purpose to identify an internal model's parameter, it may be more reasonable to emphasize the end of the assimilation window $[0,T]$ with respect to it's beginning because in the beginning of the model's integration the value of $\xi(t)$ is low and less important. Indeed, supposing observational data to be perfect, we start from the exact initial state obtained from artificial observations and get $\xi(0) = 0$.

Consequently, we shall try to use two other cost functions

$$\mathcal{I}_{fp}(\alpha) = \xi^2(\alpha,T), \quad \mathcal{I}_T(\alpha) = \int\limits_0^T t\xi^2(\alpha,t)dt \qquad (19)$$

requiring that the end of the interval is weighted more heavily.

To calculate the gradient of the cost function, we calculate first its variation:

$$\begin{aligned}
\delta\mathcal{I}_{fp} &= \mathcal{I}_{fp}(\alpha+\delta\alpha) - \mathcal{I}_{fp}(\alpha) = \xi^2(\alpha+\delta\alpha,T) - \xi^2(\alpha,T) = \\
&= \ll\phi(\alpha+\delta\alpha,T) - \phi^{obs}(T), \phi(\alpha+\delta\alpha,T) - \phi^{obs}(T)\gg - \\
&\quad - \ll\phi(\alpha,T) - \phi^{obs}(T), \phi(\alpha,T) - \phi^{obs}(T)\gg\sim \\
&\sim 2\ll\delta\phi(T), \phi(\alpha,T) - \phi^{obs}(T)\gg
\end{aligned} \qquad (20)$$

This scalar product is similar to the scalar product in the left-hand-side of (12) with $\phi^*(T)$ replaced by $\phi(\alpha,T) - \phi^{obs}(T)$. Consequently, if we run the adjoint model (13) from the state $\phi^*(T) = \phi(\alpha,T) - \phi^{obs}(T)$ back in time from $t = T$ to $t = 0$ we obtain the state $\phi^*(0)$. This state, being scalarly multiplied by $\delta\phi(0)$ provides the variation of the cost function

$$\delta\mathcal{I}_{fp} = 2\ll\delta\phi(0), \phi^*(0)\gg \qquad (21)$$

As it has been noted, first three components of $\delta\phi(0)$ are equal to zero. Consequently, first three components of $\phi^*(0)$ are multiplied by zero and we are interested only in the fourth component of the result of the adjoint model α^*.

Thus, the gradient of the cost function is related to the fourth component of $\phi^*(0)$

$$\nabla\mathcal{I}_{fp} = 2[\phi^*(0)]_4 = 2[\mathcal{A}^*(T,0)(\phi(\alpha,T) - \phi^{obs}(T))]_4 \qquad (22)$$

and namely to the $\alpha^*(0)$ is obtained as the final point of the adjoint model integrations from $t = T$ to $t = 0$ beginning with the state $\phi^*(T) = \phi(\alpha,T) - \phi^{obs}$.

Gradients of two other cost functions can be obtained in a similar way

$$\begin{aligned}
\nabla\mathcal{I}_4 &= 2\int\limits_0^T \mathcal{A}^*(T,t)(\phi(\alpha,t) - \phi^{obs}(t))dt \\
\nabla\mathcal{I}_T &= 2\int\limits_0^T t\mathcal{A}^*(T,t)(\phi(\alpha,t) - \phi^{obs}(t))dt
\end{aligned} \qquad (23)$$

These gradients are used in the minimization procedure that is implemented in order to find the minimum

$$\mathcal{I}(\bar{\alpha}) = \min_{\alpha} \mathcal{I}(\alpha) \tag{24}$$

Coefficients $\bar{\alpha}$ are considered as coefficients realizing optimal discretization of the model's operators in the boundary regions.

The minimization procedure used here was developed by Jean Charles Gilbert and Claude Lemarechal, INRIA [28]. The procedure uses the limited memory quasi-Newton method.

3. Stationary solution of the model

In this section we perform several experiments with the stationary model's solution that contains a strong boundary layer. The model is considered in a square box of side length $L = 4000$ km driven by a steady, zonal wind forcing with now classical sinusoidal profile

$$\tau_x = \tau_0 \cos \frac{2\pi(y - L/2)}{L}$$

that leads to the formation of a double gyre circulation [29]. The maximal wind tension on the surface is taken to be $\tau_0 = 1.5 \frac{dyne}{cm^2}$.

As it has been already noted, the Coriolis parameter is linear function in y with $f_0 = 10^{-4} s^{-1}$ and $\beta = 2 \times 10^{-11} (ms)^{-1}$. The reduced gravity and the depth are respectively equal to $g = 0.02 \frac{m}{s^2}$, $H_0 = 1000m$. The coefficient of Eckman dissipation is chosen as $\sigma = 5 \times 10^{-8} s^{-1}$, that corresponds to the damping time-scale $T_\sigma = 2 \times 10^7 s \sim 200$ days. The lateral friction coefficient μ has been taken to be $\mu = 800 \frac{m^2}{s}$, that corresponds to the damping time scale $T_\mu = 2.5$ days for a wave of 100 km length.

All operators in the model are approximated with the second order accuracy both in the interior of the domain and near its boundary. That means all the coefficients a have been chosen to provide second order discretization as it has been discussed in the section 2.1. The expression (5), that is used to interpolate functions and to calculate their derivatives near boundary, is written with $K = 2$, $M = 1$, $\alpha_0 = 0$. Coefficients α are defined as $\alpha_{1,1} = -1/h$, $\alpha_{2,1} = 1/h$ for all derivative operators and $\alpha_{1,1} = \alpha_{2,1} = 1/2$ for all interpolations. That gives, for example, the value of the derivative of u at the point $i = 1/2$ as $(D_x u)_{1/2, j-1/2} = \frac{u_{1,j-1/2} - u_{0,j-1/2}}{h}$.

The stationary solution of the model in this configuration possesses a boundary layer near the Western boundary. This is a well known Munk layer [30] that is characterized by the local equilibrium between the β-effect and the lateral dissipation. It's width can be easily calculated by solving the Munk equation

$$\frac{\partial^3 v}{\partial x^3} - \frac{1}{\delta^3} v = 0, \quad v(0) = 0, \quad v(\infty) = 0, \quad \delta = \left(\frac{\mu}{\beta}\right)^{1/3} \tag{25}$$

The solution has a form

$$v(x) = C e^{\frac{-x}{2\delta}} \sin(\frac{\sqrt{3}x}{2\delta})$$

The width of the Munk layer for given parameters is equal to $\frac{2\pi}{\sqrt{3}}(\mu/\beta)^{1/3} = 124$ km.

As it has been discussed in [31], the model must resolve this layer with at least one grid point (optimally more than one grid point) in order to maintain numerical stability. The work of [32] emphasized the importance of having at least two grid points in the Munk layer in order to minimize the level of spurious oscillations visible in the velocity fields as well in the field of the sea surface elevation.

The resolution of the model in this section is intentionally chosen to be too coarse to resolve the Munk layer. The model's grid is composed of 30 nodes in each direction, that means the grid-step is equal to 133 km, that is more than the Munk layer's width. As it can be seen in fig.1, there is only one grid node in the layer for variables v and η and no nodes for u variable.

Artificial "observational" data are generated by the same model with all the same parameters but with 9 times finer resolution (15 km grid step). The model with fine resolution, having 8 nodes in the Munk layer, resolves explicitly the layer and must have no spurious oscillations. All nodes of the coarse grid belong to the fine grid, consequently, we do not need to interpolate "observational" data to the coarse grid. We just take values in nodes of the high resolution grid that correspond to nodes on the coarse grid.

Along with the models on the fine grid that is used to generate "observations" and on the coarse grid that is used to perform data assimilation experiments, we also use the medium resolution model on the grid 3 times finer than the coarse grid. The grid step of the medium grid is equal to 44 km, there are 3 nodes in the Munk layer, which is, hence, relatively well resolved.

In order to compare these three models, we use the v velocity field because under-resolution of the Munk layer in the most visible in the field of this variable. If we compare profiles of the velocity $v(x, y)$ plotted in fig.2 at $y = 1000$ km (one quarter of the side length), we see that the profiles of the models with fine and medium resolutions coincide almost everywhere. The only visible difference is seen between the second and the third nodes of the medium-resolution grid. The maximum of velocity is located just between these nodes and is not reproduced on the medium grid. However, values of the velocity in nodes adjacent to the maximum are close to corresponding values of the model on the fine grid.

The profile obtained with the coarse resolution model (solid line in fig.2) shows strong spurious oscillations due to the unresolved Munk layer. As expected, one point in the layer is not sufficient to suppress the numerical mode in the velocity field. Similar oscillations are also present in the fields of the velocity u and SSH η.

Performing the data assimilation we hope that the optimal discretization of operators near the boundary may help us to suppress these spurious oscillations even working on the coarse grid.

3.1. Parameters of the assimilation procedure

In this section we shall discuss such parameters of the assimilation procedure as the length of the assimilation window, the choice of the cost function among $\mathcal{I}_4$, $\mathcal{I}_{fp}$ and $\mathcal{I}_T$ and the physical meaning of the control parameters and their eventual dependence on the coordinate. As the first experiment, we identify only three coefficients for each operator and each boundary. That means, we take $K = 2$, $M = 1$ in (5).

We can note that this choice is almost equivalent to the control of the boundary conditions prescribed for u and v variables. Indeed, supposing, for example, the boundary conditions for $u(x, y)$ for theoperator $\frac{\partial u}{\partial x}$ at $x = 0$ are expressed according to (6), the derivative
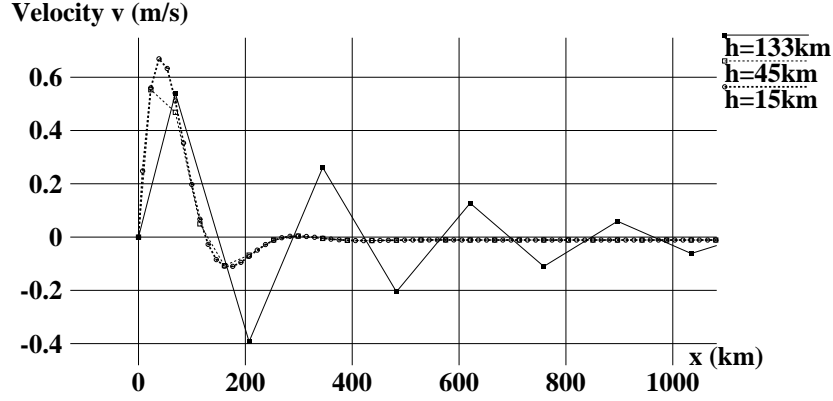
Figure 2. Profiles of the velocity $v(x, y)$ at $y = 1000$ km obtained on the coarse grid with $h = 133km$ (solid line), medium grid with $h = 45$ km (dashed line) and fine grid with $h = 15$ km (bold dashed line)

of u at $x = h/2$ that we need in the third equation of the model (1) can be found as $\left.\frac{\partial u}{\partial x}\right|_{1/2} = \frac{u_{1,j}}{h - r_1} - \frac{r_2}{h - r_1}$. Therefore, if we want to control boundary conditions, we can control either r_1 and r_2 in the expression of boundary conditions or $\alpha_{0,1}^{D_x u} = -\frac{r_2}{h - r_1}$ and $\alpha_{2,1}^{D_x u} = \frac{1}{h - r_1}$ in discretization of the operator (5). The result will be the same in both cases: we control two parameters looking for minimizing the same cost function (the coefficient $\alpha_{1,1}^{D_x u}$ is not controlled due to keeping formal Dirichlet condition $u(0, y) = 0$).

However, even in this case there is a difference between controlling the boundary conditions and controlling the discretizations of operators. Controlling the boundary conditions and finding optimal r_1 and r_2 suppose using the same r_1 and r_2 for the interpolation operator near boundary. But finding the optimal α, we suppose that $\alpha^{D_x u}$ and $\alpha^{S_x u}$ are independent on each other.

Moreover, we admit that coefficients in the gradient are also controlled by data assimilation, despite no boundary condition is prescribed for η. This provide additional control that may help to improve the solution near the boundary.

From the physical point of view, assuming variable coefficients (either r_1 and r_2 or $\alpha_{0,1}^{D_x u}$ and $\alpha_{2,1}^{D_x u}$ we accept the flux can cross the boundary. Impermeability condition $u(0, y) = 0$ is no longer maintained. This fact must be addressed in the first turn, because it represents a major violation of the model's physics. Allowing the fluid to leave the domain in some place, we must allow it to come back in some other place in order to be able to keep the mass constant. In the present formulation, the coefficient $\alpha_{0,1}^{D_x u}$, that determines non null flux crossing the boundary, is constant. That means, the flow may cross the boundary in the same direction all the boundary's long. The flux can not change direction and reenter the domain in some place after leaving it in some other place. Consequently, in order to allow the flux to come back, we must at least allow control coefficients α_0 to depend on the coordinates: latitude y
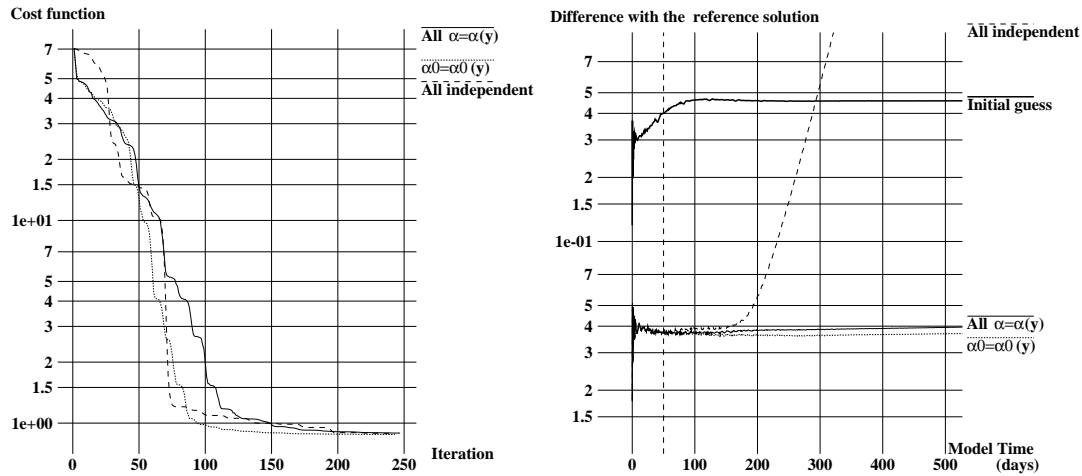
Figure 3. Convergence of the cost function in the minimization procedure (left) and evolution of the norm of difference with the reference solution (right) in the experiments with all coefficients α particular for each point (solid line), with just α_0 depending on the coordinate (small dashes) and all uniform α (long dashes).

corresponding to the index j for operators S_x, D_x and G_x, and longitude x or index i for operators S_y, D_y and G_y. Thus, we obtain a particular discretization of the operators at each point. Parameter $\alpha_{0,m}^{D_x u}$ in the expression (5) will have a form $(\alpha_{0,m}^{D_x u})_j$. We can also allow all parameters $\alpha_{k,m}$ to depend on the coordinates.

The initial guess for the minimization procedure is taken as the classical second order approximation, the same as the approximation that has been used to run the fine resolution model to generate artificial "observations". Three experiments have been performed. In the first experiment we allow all α to depend on the coordinate on the boundary; in the second experiment it is α_0 only that depends on coordinate, and in the third experiment all α are independent. Assimilations was performed minimizing the 4D-VAR cost function $\mathcal{I}_4$ (18) with assimilation window $T = 50$ days and the results are shown in fig.3. The assimilation window is delimited by the vertical dashed line in the figure.

As one can see in fig.3, 150 iterations are sufficient for the minimization process to converge in general. In all subsequent iterations the value of the cost function is modified just a little and can be considered as already converged to the same value in all three experiments. We can notice only a slightly different convergence rate. When the coefficients α are particular for each grid point, the convergence is a little slower. The procedure takes about 150 iterations to achieve the value obtained after 90 iteration when the control coefficients are uniform or partly uniform (just α_0 depends on the coordinate). The final value of the cost function is 80 times lower than the starting value, that means the solution is much improved in the assimilation process.

However, the particularity of the data assimilation applied to control the discretization of operators consists in the fact that obtained numerical approximation of derivatives and

interpolations can be instable. In the assimilation window this numerical instability may be dumped by the minimization, but it may bring the solution to infinity beyond the window.

One can observe this phenomenon in the fig.3 on the right where the evolution of the norm of the difference $\xi(t)$ (16) is plotted during 500 days (the assimilation window, as it has been already mentioned, is 50 days long). The solution obtained by the model with uniform coefficients α is indeed instable. In the assimilation window the line that corresponds to this solution is indistinguishable from two other lines, but, after the assimilation's end, the difference grows up and leave the picture's frame.

Hence, as it has been shown above, using all uniform coefficients is not the best choice. Mass and momentum fluxes can not be balanced on the boundary and the assimilation procedure have to look for the optimum in the domain where the scheme is instable. Consequently, at least α_0 should be allowed to be variable and specific in each point near the boundary in order to allow the integral flux across the boundary to be balanced. This requirement substantially increases the dimension of the control space. If all control coefficients do not depend on the coordinate, the number of coefficients for the linear shallow water model may be as few as 48. Requiring the dependence of α_0 on the position of the boundary point, we increase the dimension of the control space up to $32 + 16 \times N$, where N is the number of grid nodes on the boundary part where the control is performed. In the experiment with the coarse resolution ($h = 133$ km, $N = 30$), for example, the dimension increases from 48 to 512. This may be really penalizing when we try to avoid the development of the adjoint model and calculate the gradient of the cost function by a finite difference method. However, even in this case the number of coefficients to control is much less than the dimension of the model's state (900 points for each of 3 variables). Moreover, if we refine the grid, the number of coefficients will always be less than number of model's variables because the number of α is proportional to the length of the boundary, while the dimension of the model state is proportional to the area of the domain.

So far, the difference between experiments with all variable coefficients and with variable α_0 only is low, we shall use this configuration in all experiments below. That means, all the approximations near boundary will be calculated by the formula

$$(D_x u)_{m-1/2,j-1/2} = \frac{1}{h}\left((\alpha_{0,m}^{D_x u})_j + \sum_{k=1}^{K} \alpha_{k,m}^{D_x u} u_{k-1/2,j-1/2}\right) \text{ for } m = 1, 2, \ldots, M \qquad (26)$$

The set of optimal coefficients α obtained in this assimilation is the following. All coefficients α other than α_0 have moved very little from their initial guesses (less than 0.01), except $\alpha^{S_x \eta}$ on the left boundary. The approximation of the interpolation $S_x \eta$ become

$$(S_x \eta)_{1,j-1/2} = (\alpha_0^{S_x \eta})_j + 0.46 \eta_{1/2,j-1/2} + 0.52 \eta_{3/2,j-1/2} \qquad (27)$$

On the other hand, coefficients α_0 have been really used as controls in this experiment. Their deviation from the initial guess (zero) is non negligible and, indeed, they depend on the coordinate. Three of them, that have been obtained for the operators in x ($(\alpha_{0,1}^{D_x u})_j$, $(\alpha_{0,1}^{S_x u})_j$ and $(\alpha_0^{S_x \eta})_j$) are shown in fig.4 on the left, and three others, for the operators in y ($(\alpha_{0,1}^{D_y v})_i$, $(\alpha_{0,1}^{S_y v})_i$ and $(\alpha_0^{S_y \eta})_i$) on the right.

On the left part of this figure we can see two important phenomena. The first one consists in the fact that the flux across the boundary is balanced. Despite no balance requirement has been imposed in the data assimilation procedure, the integral of the flux over the boundary is
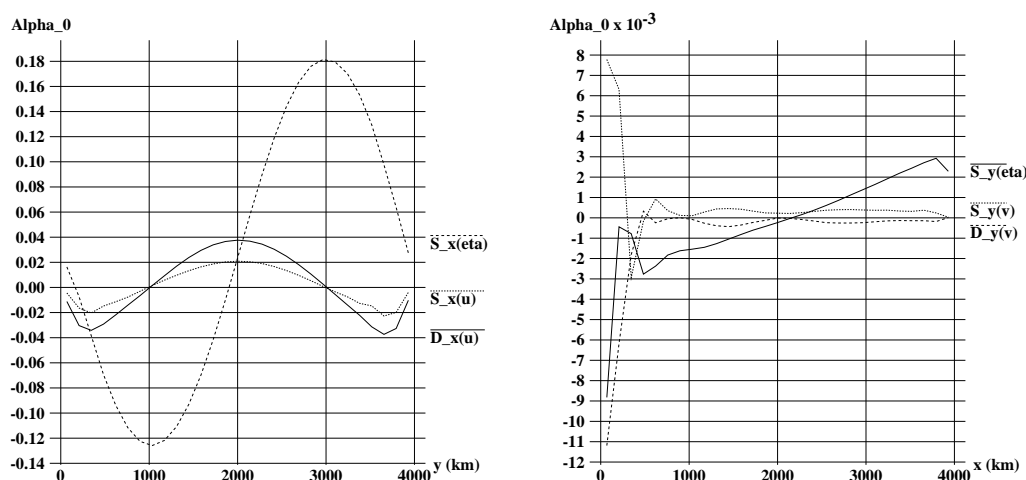
Figure 4. Optimal coefficients α_0 for operators in x (left) and operators in y (right) as functions of $y = jh$ and $x = ih$.

close to zero. The normal component of the velocity on the left boundary (i.e. u component) shows that the flux leaves the domain on the North and on the South of the boundary and returns back in the middle.

The second phenomenon consists in the fact that coefficients $(\alpha_{0,1}^{D_x u})_j$ and $(\alpha_{0,1}^{S_x u})_j$ have a certain similarity. As it has been already mentioned, they have been considered as completely independent on each other, but the result shows the coefficients $(\alpha_{0,1}^{S_x u})_j$ are very close to one half of $(\alpha_{0,1}^{D_x u})_j$.

Analyzing coefficients α_0 for operators in y shown in fig.4 on the right, we see their values are much smaller than for operators in x. It is reasonable because there is no boundary layers near the North and near the South boundaries. These coefficients have maxima only in the North-Western and the South-Western corners of the domain because of influence of the western boundary layer. Out of this layer, only coefficients $\alpha_0^{S_y \eta}$ (solid line) show significant difference from zero. The operator $S_y \eta$ is applied to the result of the interpolation $S_x u$. That means the interpolation of the velocity u to points v is performed by formula

$$u_{i+1/2,1} = (\alpha_0^{S_y \eta})_i + \frac{u_{i+1,3/2} + u_{i,3/2} + u_{i,1/2} + u_{i+1,1/2}}{4}$$

which differs from the standard second order interpolation only by adding $(\alpha_0^{S_y \eta})_i$ that represents a flux to the West in the western half of the domain and the flux to the East in the eastern half as it is shown in fig.4 on the right.

The next question we address in this section concerns the length of the assimilation window and the form of the cost function. A priori, we know the procedure is computationally less efficient when T is long. Computing time per iteration increases with T because each iteration is composed by the integration of the direct model from 0 to T and corresponding backward integration of the adjoint model. Consequently, it would be more computationally efficient to

Table I. Values of ξ obtained by assimilation with 4 different assimilation windows and 3 different cost functions.

T (days)	Number of Iterations	$\xi(T)$			$\xi(1000$ days$)$		
		using $\mathcal{I}_{fp}$	using $\mathcal{I}_4$	using $\mathcal{I}_T$	using $\mathcal{I}_{fp}$	using $\mathcal{I}_4$	using $\mathcal{I}_T$
0.2	10	0.042	0.042	0.042	∞	∞	∞
0.2	50	0.041	0.041	0.041	∞	∞	∞
0.2	200	0.040	0.040	0.040	∞	∞	∞
2	10	0.12	0.041	0.09	1.79	0.74	1.88
2	50	0.031	0.032	0.032	0.09	0.05	0.04
2	200	0.029	0.030	0.030	1.22	∞	∞
20	10	0.27	0.26	0.27	0.56	0.54	0.64
20	50	0.11	0.064	0.08	0.24	0.17	0.14
20	200	0.04	0.037	0.04	0.050	0.040	0.036
200	10	0.25	0.25	0.25	0.25	0.25	0.25
200	50	0.21	0.21	0.21	0.21	0.20	0.19
200	200	0.05	0.04	0.03	0.049	0.039	0.035

choose a small T. However, it is the precision of the assimilation that we want to improve and the precision must be used as criterium in the choice of the assimilation window.

The choice of the cost function among $\mathcal{I}_4$, $\mathcal{I}_{fp}$ and $\mathcal{I}_T$ is not obvious also. As it has been already noted, the cost function $\mathcal{I}_4$ gives the same importance to all part of the assimilation window, while $\mathcal{I}_T$ and $\mathcal{I}_{fp}$ consider the end of the interval to be more important than the beginning.

In order to see the influence of the form of the cost function and the assimilation window, we perform the set of experiments with different cost functions and different assimilation windows. In all experiments we stop the minimization process at different stages of convergence: in the very beginning, just after 10 iterations; in the middle after 50 iterations and after 200 iteration allowing almost complete convergence. So far, final values of different cost functions are difficult to compare among them, we shall compare final values of ξ (16) that is independent on the cost function in use. Moreover, in each experiment we shall compare not only $\xi(T)$ calculated at the end of the assimilation window, but also $\xi(1000$ days$)$ in order to distinguish those assimilations that result in an instable scheme.

Analyzing the Table I, we can note several interesting particularities. First of all, when assimilation window is short, the convergence is very rapid. In fact, 10 iterations are sufficient for the minimization to converge when $T = 0.2$ days (just two time steps of the model). Taking into account that iterations with a so short T are extremely cheap in computer time, it seems to be advantageous to perform assimilations with short windows. However, these assimilations result in an unstable scheme. Two time steps is too short interval to develop the instability. The cost function contain no information about stability of the scheme, consequently, the minimization chooses a set of coefficients α with no requirements about stability of the obtained scheme.

If we consider assimilation window $T = 2$ days, we see another interesting particularity. Ten iterations are no longer sufficient for the minimization to converge, but after 50 iterations the minimization has almost reached the minimum. Next 150 iterations allow just to reduce the cost function by 0.002, i.e. less than 10%. But these 150 supplementary iterations transform a

stable scheme to an instable one. After 50 iterations the scheme was perfectly stable allowing the model to be integrated as long as necessary. At the end of 1000 days model's run we get rather low difference with the reference solution of the model on a fine grid. But continuing minimization we win a little in the cost function loosing the stability. Consequently, we should assume the optimal discretization of operators near the boundary is situated not far from the stability limit. Performing more iterations than some maximal allowed quantity, we can cross the stability limit and obtain an instable scheme. This maximal number of iterations depends on the assimilation window. The scheme becomes instable just after several iterations when $T = 0.2$ days because there is very few information in the cost function about stability. When $T = 2$ days, the cost function knows more about stability (the solution must not diverge during at least 2 days), and the minimization supports several dozens of iterations before become instable. The stability is not guaranteed even with $T = 20$ days. This experience is not shown in the Table I, but after 1000 iterations we obtain an instable scheme also.

The conclusion, hence, is simple: short T ensures a rapid convergence, but long T is necessary to ensure the stability. Reasonable strategy of data assimilation in this circumstances may consist in using variable assimilation window. Performing few iterations with short T we obtain the set of coefficients α that ensures low cost function value. This set is corrected by assimilation with longer T in order to ensure the stability of the scheme. Of course, it is important to restrict the number of iterations with short windows. Otherwise, assimilation with long windows will not be able to correct instabilities of the scheme.

Looking at the results obtained using different cost functions in the assimilation procedure, we see just a little difference in final values of $\xi(1000$ days$)$. However, the final ξ obtained with $\mathcal{I}_T$ is a little lower than with any other cost function. Consequently, the hypothesis that the end of the assimilation window should be more heavily weighted is reasonable. Excessive weighting of the end of the window by $\mathcal{I}_{fp}$, however, provides worse results. When we use just the final point in the cost function, the value of $\xi(1000$ days$)$ is bigger than in all other experiments.

Taking into account the results obtained in this section, we shall use the cost function $\mathcal{I}_T$ and perform the data assimilation with variable assimilation window T in all experiments below. A comparison of the convergence of the cost function in experiments with fixed assimilation windows of different length and with variable windows is shown in fig.5. So far, the value of the cost functions depends explicitly on T, we normalize it and plot the value $\frac{2\mathcal{I}_T}{T^2}$ which depends on T only implicitly.

In this figure one can see the same phenomenon as in the Table I. When the assimilation window is short, the convergence is rapid. Just several iterations are sufficient for minimization to converge when $T = 0.2$ days or $T = 2$ days. Unfortunately, as we have seen in the Table I, obtained numerical schemes are instable. When the assimilation window is long, more iterations are necessary to converge (about 100 when $T = 20$ days and about 200 with $T = 200$ days). Moreover, each of these iterations is more expensive in computer time.

One can see in fig.5 also, the convergence of $\frac{2\mathcal{I}_T}{T^2}$ with the variable window is rapid, the final value of the cost function is the same and obtained set of coefficients α provide a stable scheme. In the experiment with variable window we have performed 4 iterations with $T = 0.2$ days, 8 iterations with $T = 2$ days and 20 iterations with each of $T = 20$ and 200 days. In fact the computing time in the experiment with variable window is equivalent to 23 iterations with $T = 200$ days, that means we obtain the same result almost 10 times more rapidly in terms of
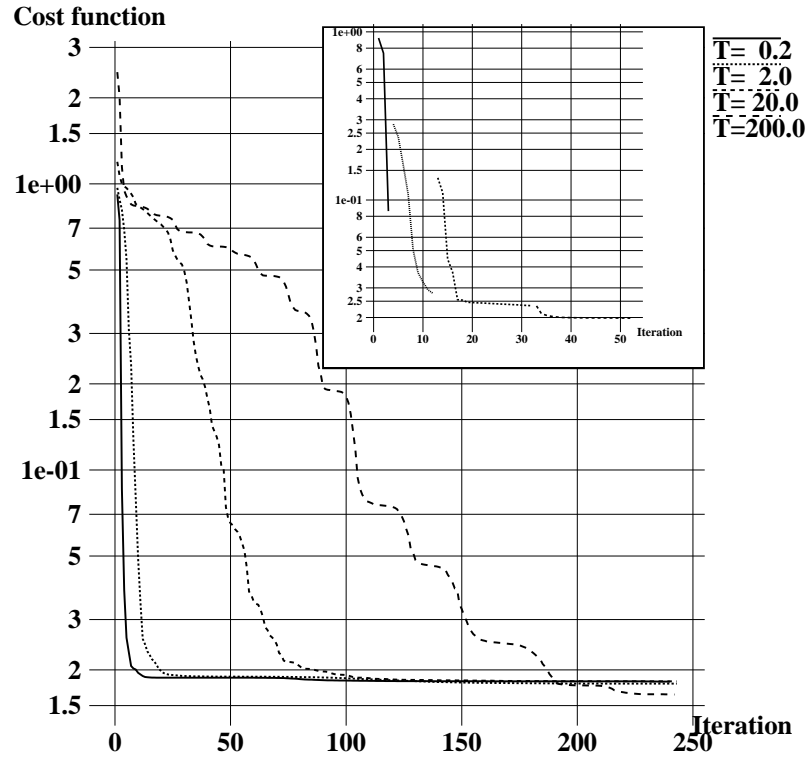
Figure 5. Convergence of the normalized cost function in the assimilation procedure with different assimilation windows and with sequentially increasing T.

CPU time.

3.2. Assimilation window and the number of control parameters.

As it has been already noted, the difference with controlling boundary conditions consists in the liberty to increase the number of controlled parameters α increasing either K or M (or both) in (26). In this section we perform a set of experiments with different K or M looking for the solution of the model on the coarse grid that is close to the reference model's solution on a fine grid. To measure the distance between two model's solutions we use the norm of the difference ξ (16) between them taken at 1000th day of the model time. The time 1000 days is chosen as sufficiently long time at which all the intermittent processes are over and the model has reached its equilibrium. (the model is linear and dissipative, so the equilibrium exists and represents the global attractor of dimension 0). From the Table I we see that with $K = 2$ and $M = 1$ we can obtain $\xi(1000 \text{ days})=0.035$ while the value of $\xi(1000 \text{ days})$ calculated for the original coarse resolution model with classical discretizations of all operators is equal to 0.46.

   That means, performing data assimilation, we have obtained the solution that is more than

10 times closer to the reference one than the initial guess. In fact, optimal discretization of operators near boundary has allowed us to obtain the solution on the coarse grid (grid step $h = 133$ km) that is similar to the solution on the medium grid ($h = 44$ km). Indeed, the difference ξ between equilibria on the medium and on the fine grids is equal to 0.037 that is almost the same as for the coarse grid's solution with optimal α. The profile of the velocity $v(x, y)$ with $x = 1000$ km of the solution on the medium grid is presented in fig.2 together with coarse grid's and fine grid's solutions. We see that there is no spurious numerical oscillations in the solution on the medium grid. Hence, the data assimilation helped us to remove these oscillations on the coarse grid also.

However, increasing K and M does not help to diminish the difference with the reference model. Several experiment have been performed with $M = 2$ and $M = 3$ and with K varying from 2 to 7, but no significant improvement has achieved. The lowest value of ξ obtained in these experiments is equal to 0.012. This value is slightly lower than 0.035 obtained with $K = 2$ and $M = 1$, but these values are of the same order.

That means controlling the discretization of operators near boundary we can avoid spurious numerical oscillation due to under-resolved boundary layer and obtain the solution similar to the solution on the 3 times finer grid. But we can not go further and to get a solution closer to the reference one. This phenomenon can be explained by the analysis of the error field, i.e. the differences $(u_{i,j} - u_{i,j}^{obs})$, $(v_{i,j} - v_{i,j}^{obs})$ and $(\eta_{i,j} - \eta_{i,j}^{obs})$ that are used to calculate ξ. The error in the initial guess is concentrated in the boundary layer while the model with optimal discretization near the boundary provides a solution with more uniform error's distribution in the whole domain. Thus, the error in the sea surface elevation in the initial guess reaches 30 meters in the boundary layer and 2-3 meters in the eastern part of the square. But after data assimilation we get an error of 1 meter near the western boundary and 0.3 meter in the eastern part. More uniform distribution of the difference with the reference solution indicates that the error produced by numerical scheme in the interior of the domain become relatively more important. And so far, we do not control the interior scheme we can not reduce this error. No matter how many α we use to control the approximation near the boundary, they can not improve the approximation in regions far from the boundary.

In order to verify this hypothesis, we perform a similar set of experiments with the fourth order approximation's schemes in all internal points of the domain. As well as before, all operators have been approximated by (3), but with coefficients $a_k^D = \frac{1}{24h}(1, -27, 27, -1)$ for $k = -2, -1, 0, 1$ for derivatives and $a_k^S = \frac{1}{8}(-1, 9, 9, -1)$ for interpolations. The same initial guess α as in experiments with the second order approximation near the boundary has been used.

Data assimilation in all these experiments has been performed with variable windows from $T = 2$ to $T = 200$ days. As we have seen before, obtained numerical scheme may be instable and generate a rapidly divergent solution after the end of assimilation. For this reason, together with the cost function over the assimilation window T we look at the value of the norm of the difference between the solution on the current iteration and the reference solution ξ (16) taken at the end of the test interval, i.e. on the 1000th day (i.e. 5 times the longest assimilation window). The minimization process is interrupted when the scheme begins to loose its stability. That means the value of ξ taken at the end of the test interval $T = 1000$ days starts to grow despite decreasing of ξ at the end of the assimilation window. The lowest value $\xi(1000$ days) obtained in the assimilation is shown in the Table II.

Table II. Values of $\xi(1000$ days) obtained by assimilation with different number of control coefficients by second and fourth order schemes.

K and M	Second order	Fourth order
Initial Guess	0.46	1.3
Medium grid	0.039	0.028
$K = 2,\ M = 1$	0.035	0.15
$K = 3,\ M = 1$	0.035	0.15
$K = 5,\ M = 1$	0.035	0.15
$K = 5,\ M = 2$	0.018	0.010
$K = 7,\ M = 3$	0.013	0.001
$K = 7,\ M = 4$	0.012	0.0004

Considering the initial guess, we can see, that the value of ξ at the end of test interval is bigger for the model with the fourth order scheme than with the second order one. The reason of this is clear: high order scheme work worse when principal physical scales are not resolved explicitly. Indeed, the coarse grid of the model does not resolve the Munk boundary layer (25). The grid step h is approximately twice the Munk parameter $2\delta = 2(\mu/\beta)^{1/3}$. But it is the ratio $\left(\dfrac{h}{2\delta}\right)^n$ where n is the order of approximation that determines the approximation error (4). Higher is the order of approximation, bigger is this ratio and bigger is the error in the approximation of the boundary layer.

Another interesting phenomenon that can be seen in the Table II consists in relative independence of the final value of ξ on the number of control coefficients K. No matter how many terms participate in the linear combination that approximates the derivative or interpolates the function, the minimization procedure converges always to the same value. Obviously, this fact is in agreement with the feature discussed above. Using more terms in linear combination allows us to increase the approximation's order. One can construct a second order scheme with two terms only, but with 4 terms any approximation up to fourth order is possible. However, as we have seen, increasing the order of approximation does not help to improve the model on this coarse grid. Moreover, the interpolation obtained by data assimilation (27) approximates the function with zero order $O(1)$. Consequently, it is useless to increase the number of terms K in the linear combination because the data assimilation does not look for increasing the order.

On the other hand, increasing M with the fourth order scheme really improves the result. Controlling numerical schemes in more than one point near the boundary allows us to obtain the solution which is very close to the reference one, obtained on the nine times finer grid. In fact, controlling α in one additional point divides the final value of ξ by 10. When M is equal to 4, $\xi(1000$ days) is as small as 4×10^{-4}, that means the solution with optimal α is almost indistinguishable from the reference solution.

Let us look at the set of control coefficients α obtained in the last experiment: $K = 7,\ M = 4$ with the fourth order approximation in the interior part of the domain. It is this experiment that provides the closest trajectory to the reference one. This set of control parameters contains more elements that the set described in the previous section. We control discretization of derivatives and interpolation operators in four points near boundary ($m = 1, 2, 3, 4$) and each
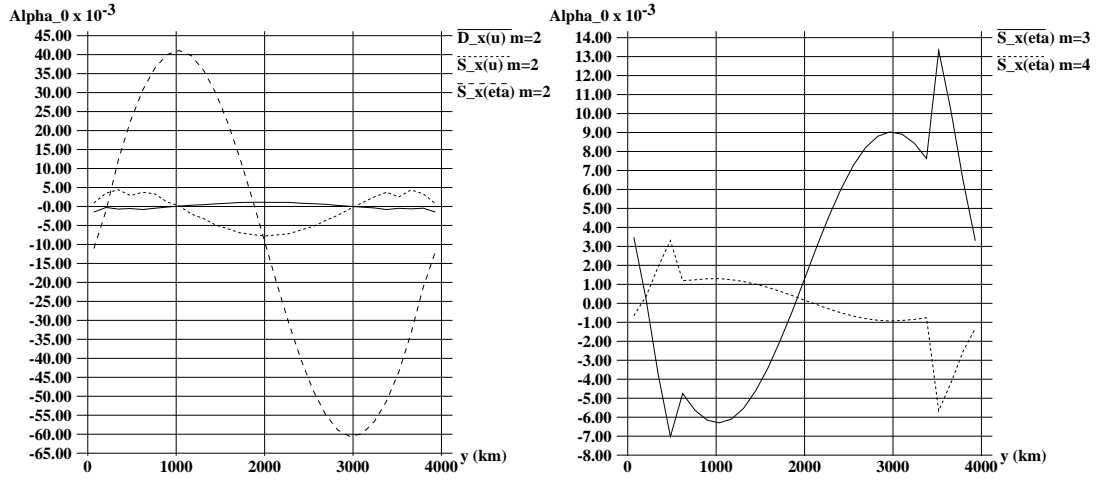
Figure 6. Optimal coefficients α_0 for $m = 2$ (left) and $m = 3$ and 4 (right) as functions of $y = jh$.

derivative and interpolation is approximated by linear combination of the function's values in seven adjacent points ($k = 1, \ldots, 7$).

However, we get a very similar discretizations in the closest to the boundary points ($m = 1$). All coefficients α other than α_0 have also moved very little from their initial guess (less than 0.01), except $\alpha^{S_x\eta}$ on the left boundary. The approximation of the interpolation $S_x\eta$ become

$$(S_x\eta)_1 = (\alpha_0^{S_x\eta}) + 0.46\eta_{1/2} + 0.50\eta_{3/2} - 0.02\eta_{5/2} - 0.006\eta_{7/2} - 0.006\eta_{9/2} - 0.001\eta_{11/2}$$

Comparing this expression with (27), we can see that the difference consists just in replacement of the term $0.52\eta_{3/2}$ by the linear combination of η in next nodes.

Coefficients α_0 have also been really used as controls in this experiment and they have almost the same values as in the experiment with $K = 2$, $M = 1$. Their dependence on coordinate is very similar to the dependence shown in fig.4.

The coefficients $\alpha_{k,2}$ that are used in approximations in nodes next to the first has been modified more significantly than coefficients $\alpha_{k,1}$ in nodes adjacent to the boundary. The deviation from the initial guess exceeds 0.01 for all operators in the Munk Layer and for $D_y v$, $S_y v$, $S_y \eta$ near the North and the South boundaries. Coefficients $\alpha_{0,2}$ and $\alpha_{0,3}$ $\alpha_{0,4}$ in the Munk layer are shown in fig.6.

Comparing this figure with the fig.4 on the left, we can see that all $\alpha_{0,1}^{S_x u}$, $\alpha_{0,1}^{D_x u}$, $\alpha_{0,2}^{S_x u}$ and $\alpha_{0,2}^{D_x u}$ are proportional to each other as well as $\alpha_{0,1}^{S_x \eta}$ and $\alpha_{0,2}^{S_x \eta}$. The data used to plot these figures allows us to show that $\alpha_{0,2}^{S_x u} = -\dfrac{\alpha_{0,1}^{S_x u}}{2.7}$ and $\alpha_{0,2}^{D_x u} = \dfrac{\alpha_{0,1}^{D_x u}}{34}$.

Concerning operator $S_x\eta$, we see that coefficients $\alpha_{0,1}^{S_x \eta}$ and $\alpha_{0,2}^{S_x \eta}$ may reach values 0.18 and 0.04 respectively. Taking into account the fact that these are dimensional variables, that means additional elevation up to 18 and 4 centimeters is added to the interpolation of the sea surface hight near boundary.

Table III. Values of $\xi(1000$ days$)$ obtained by assimilation of all 3 variables and of the altimetry data only.

K and M	All variables			Only η		
	2nd order	4h order	Iterations	2nd order	4th order	Iterations
Initial Guess	0.46	1.3		0.46	1.3	
$K = 2,\; M = 1$	0.035	0.15	200	0.042	0.18	1 200
$K = 5,\; M = 2$	0.016	0.010	1 000	0.016	0.015	5 000
$K = 8,\; M = 4$	0.012	0.0004	3 500	0.016	0.0006	15 000

3.3. Using restricted observational data.

In the previous section we have supposed that observational data are available for all variables. However, it is the altimetry that is the most probable candidate to be an observable variable in real data assimilations. In this section we suppose to have the sea surface elevation η as the only observable data. As well as before, these data are produced by a high resolution model on the 9 times finer grid, but we put $w_u = w_v = 0$ in (10) in order to neglect all the information about u and v in the cost function and in its gradient.

However, assimilating the sea surface elevation only, we hope to be capable to identify numerical schemes for all variables. Consequently, comparing the quality of assimilation we keep former w_u and w_v in (16). Thus, the norm of the difference ξ between the solution of the reference model and the solution with optimal numerical scheme on the boundary takes into account the information about all variables. The difference ξ is, consequently, directly comparable with corresponding ξ obtained in previous experiments.

Several experiments with different K and M have been performed in the same way as in the previous section. We are also using $\xi(1000$ days$)$ as the final measure of data assimilation's quality because this value, being taken well beyond the assimilation window, ensures the stability of the scheme. Results of the assimilation of altimetry data only are shown in the Table III together with several results of assimilation of all variables already shown in the Table II.

Analyzing the Table II, we have concentrated our attention on the final results of the assimilation and namely on the value of the difference with the reference solution. Now, when we restrict the number of observable variables, we should also pay attention to the cost of assimilation. As it has been already mentioned, results in the Table II have been obtained by data assimilation with variable window T. To create the Table III, we have re-obtained the same results assimilating data with a fixed window $T = 200$ days. The minimization procedure in each experiment has been stopped after stabilization of the cost function. The number of iterations in this case is proportional to the computer time spent to minimize the cost function. This value is, consequently, comparable in different assimilation experiments.

Comparing the number of iterations shown in the Table III, we see a valuable growth of computing time necessary to converge the minimization process. The number of iterations increases rapidly with increasing of the number of control parameters. Bigger K and M ensures lower difference with the reference solution, but the assimilation becomes more expensive. In fact, the convergence rate (i.e. average decrease of the cost function per iteration) is independent on the number of control parameters, but more iterations are necessary to reach
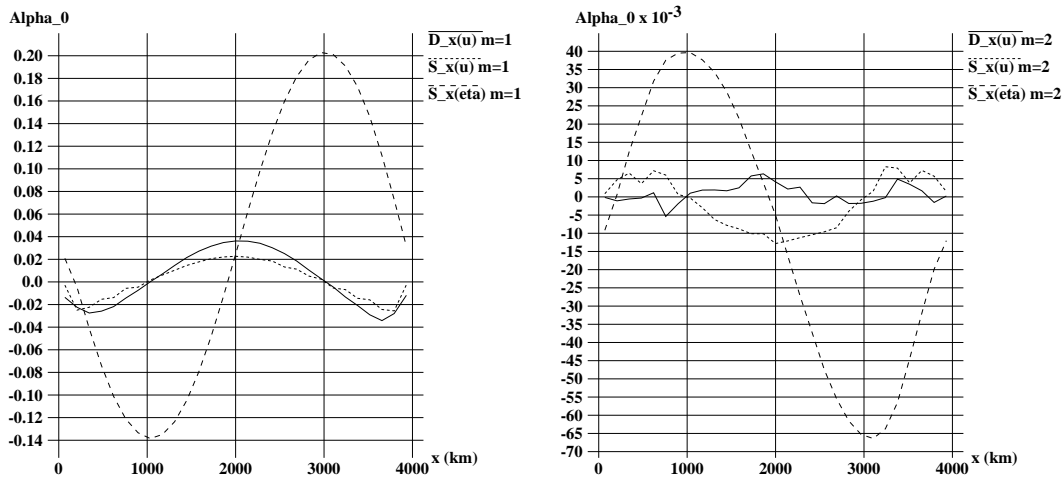
Figure 7. Optimal coefficients α_0 for $m = 1$ (left) and $m = 2$ (right) as functions of $y = ih$ in the experiment with the only observable variable η.

lower final value of the cost function. Thus, intermediate values of $\xi(1000$ days$)$ obtained after 200 and 1000 iterations in the experiment with $K = 8$, $M = 4$ are equal to 0.15 and 0.01 respectively, that corresponds exactly to the final values of ξ in the experiments with $K = 2$, $M = 1$ and $K = 5$, $M = 2$.

Restricted number of observable variables leads also to slower convergence. The number of iterations with only one observable variable η is approximately 5-8 times higher than in experiments assimilating data of all three variables. This is not surprising: spurious oscillations due to under resolved Munk layer are the most visible in the velocity v while the sea surface height data are assimilated. The information, hence, must be transferred to v. This takes more time and requires supplementary iterations.

Concerning assimilation results with only one observable variable, we can see that final values of ξ are close to corresponding values obtained assimilating data collected for all variables. Values in the left part of the Table III are a little lower, but this seems to be due to preliminary stop of the minimization procedure. Convergence becomes so slow that matches the stop condition despite further decrease is still possible.

The hypothesis of the preliminary stop is also confirmed by the analysis of obtained assimilation results. As well as in previous experiments, all α except α_0 have moved very little from their initial positions. Coefficients α_0 have been subjected to more significant modifications. Their values for operators acting in x coordinate in the experiment with $K = 8$, $M = 4$ are shown in fig.7 for $m = 1$ and 2. This figure can be compared with fig.4 and fig.6. We can see that $\alpha_{0,1}$ for all operators $D_x u$, $S_x u$ and $S_x \eta$ have converged to very similar values as in previous experiments. Preliminary stop of the minimization procedure results in irregularity of $\alpha_{0,2}$ for operators $D_x u$ and $S_x u$. Their lines have the same appearance as corresponding lines in fig.6, but they contain w some irregular noise which may represent the consequence of too early stop of the minimizator.

Consequently, the lack of observations leads to a slower convergence. Assimilating the sea surface elevation only, we must take into account that the computational cost of the procedure may be multiplied by 10.

4. Waves

Forced and dissipative linear shallow water model possesses a stationary solution that was considered in the previous section. So far, the numerical scheme under control does not depend on time, the use of a stationary solution represent an obvious advantage. Of course, it is easier to find optimal discretization when the flow is stable.

In this section we shall look for the set of α that is optimal for a non-stationary solution of the shallow water model and namely inertia gravity and Rossby waves. The shallow water model is written without forcing nor dissipation:

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= (f_0 + \beta y)S_x S_y v - g G_x \eta \\
\frac{\partial v}{\partial t} &= -(f_0 + \beta y)S_y S_x u - g G_y \eta \\
\frac{\partial \eta}{\partial t} &= -H_0(D_x u + D_y v)
\end{aligned}
\tag{28}
$$

The system becomes hyperbolic with exact conservation of total energy and momentum. The mass is also conserved as well as in the forced-dissipative model. So far, there is no more harmonic dissipation in the problem, boundary conditions have been reformulated in order to ensure the existence and uniqueness of a solution:

$$
u(0, y) = u(L, y) = v(x, 0) = v(x, L) = 0
\tag{29}
$$

We keep the same model's parameters as above, including interpolation and derivative operators, their discretizations in the interior of the domain (3) and near boundaries (26). The initial state of the model (28) have been chosen as

$$
\begin{aligned}
u(x, y, 0) &= \sin(\pi x/L)\sin(2\pi y/L), \ v(x, y, 0) = \sin(\pi x/L)\sin(\pi y/L), \\
\eta(x, y, 0) &= H_0 + 10\cos(\pi x/L)\cos(\pi y/(2L)).
\end{aligned}
$$

Artificial "observational" data have been produced by the same model on the high resolution grid with 263×263 nodes ($h = 15$ km) with the second order approximation scheme in the interior of the domain. Assimilation experiments are performed on the 31×31 nodes grid ($h = 133$ km) with either second or fourth order approximation.

As well as before, we use the norm of the difference ξ (16) to measure the quality of assimilation. However, the solution now is not stationary and ξ varies in time also. Consequently, instead of using a value of ξ at some sufficiently distant moment of time, we shall examine the evolution of $\xi(t)$. This evolution is compared with the evolution of the difference between the solution of the reference model on a high resolution grid and the solution of the model on the medium resolution (89×89 nodes) grid. This comparison allows us to position the assimilation's result.
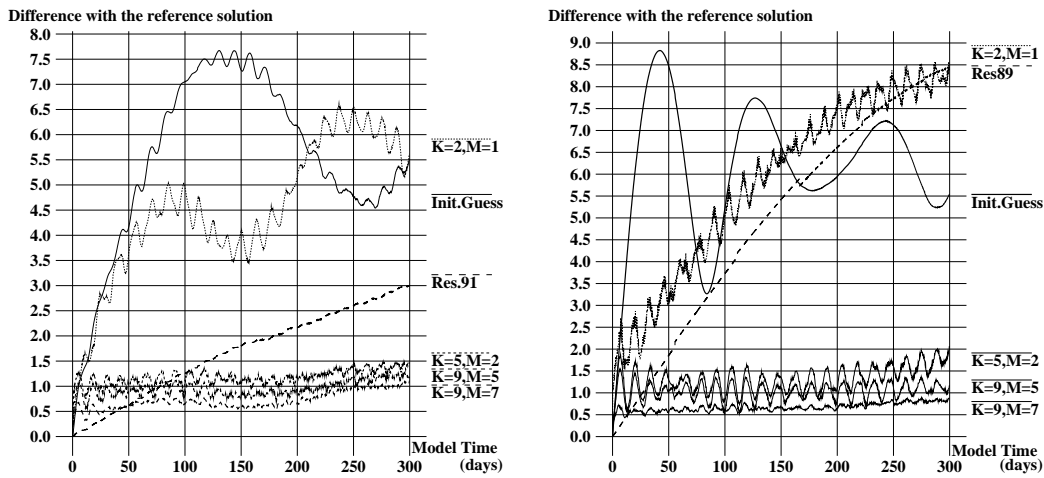
Figure 8. Evolution of the difference $\xi(t)$ in the assimilation experiment with different K and M with the second (left) and fourth (right) order approximation scheme in the interior.

Several assimilation experiments have been carried out with different number of control coefficients K and M. Variable assimilation window have been used in all experiments because the convergence is much more slower in this case. Assimilating data when the solution is represented by waves, requires more iteration with a fixed long assimilation window. Even 5000 iterations with window $T = 100$ days are not sufficient to move significantly from the initial guess point and we have to use variable windows in order accelerate the convergence rate.

Solutions of the system (28) is composed by inertia-gravity and Rossby waves. Characteristical periods of the inertia-gravity wave for the given parameters and initial conditions is of order of 17 hours while typical period of Rossby waves is about 100 days. This difference in characteristic times makes additional advantages of using variable assimilation windows. When the window is short, of several days length, the numerical scheme is adapted to correspond to inertia gravity waves. And one hundred days window is more adapted to assimilate the information about Rossby waves.

In this chapter we use the sequence $T = 2, 6, 20, 100$ days with 50 iterations made for each assimilation window T. Experiments with different K and M have been carried out for both second and fourth order approximation schemes in the interior of the domain.

This evolution of the difference $\xi(t)$ is shown in fig.8. We plot this difference during 300 days time interval in order to show the evolution beyond the assimilation window and, in particular, to see whether obtained scheme is stable. .

One can see that the principal error produced by the numerical scheme consists in a wrong wave speed of inertia-gravity waves. This error is visible in the fig.8 because there is only one trigonometric mode in the initial conditions of the model. The phase speed error with a single mode lead to the oscillating behavior of lines representing initial guess in both left and right figures. When the second order approximation is used, the initial guess line has a minimum

on the 260th day indicating that the wave on the low-resolution grid is exactly one period shifted with respect to the wave approximated on the high resolution grid. Using fourth order approximation in the interior, we get this minimum earlier, on the 85th day.

Identification of the optimal numerical scheme at just one point near boundary does not help to correct the error in wave velocity. We need to control the approximation at least at two points near boundary. However, when M is equal or bigger than two, data assimilation lead to similar behavior of the solution. As it is shown in fig.8 the error in velocity is corrected by optimal numerical scheme for all $M \geq 2$, but only a little improvement is achieved by further increasing of M. When $M = 7$, the difference with the reference solution ξ is approximately two times lower than this difference obtained with $M = 2$. Taking into account that the data assimilation is much more expensive with $M = 7$, we can state that using $M = 2$ is optimal in the experiments with inertia-gravity waves. We can note, that similar result has been obtained in [23] in experiments with one-dimensional wave equation. It has been shown that if the wave composed by multiple trigonometric modes, data assimilation with $M = 1$ can not correct the wave speed error because the the numerical scheme that could be optimal is unstable. We have to control the scheme at two points near boundary at least.

Comparing assimilation results with the test solution obtained using a classical scheme on the medium resolution grid (89×89), we see the medium resolution also suffers by the error in the wave velocity. The error is, obviously, much smaller than in the initial guess case that uses a 3 times coarser grid 30×30. Smaller error results in a slower increase of the difference ξ. Dashed lines in fig.8, that represent the difference between the solution on the medium grid and on the fine grid, grow slower than solid lines, but they all reach the same maximal values.

So far the error in the velocity has been corrected for the solution on the coarse grid with optimal discretization near boundary, on long time scales this solution is closer to the reference solution than the solution obtained on the medium grid.

Modifications that have been made in the discretizations of interpolation operators and derivatives near boundary are not the same as in the case of the stationary solution. Contrary to results of experiments in the previous section, coefficients α_0 have not moved from their initial guesses $\alpha_0 = 0$ for all operators. Principal modifications have been applied to the discretizations them-self, i.e. to $\alpha_{k,m} : k \neq 0$.

Taking into account that experiments in this section have been performed with sufficiently large numbers of control coefficients K and it is difficult to understand what global effect brings each modification, we consider the coefficients of the Taylor expansion. For each linear combination like (26) that represent an approximation of the interpolation operator or a derivative, we calculate coefficients of the three first terms of the Taylor expansion:

$$(D_x u)_{m/2} = \frac{1}{h} \sum_{k=1}^{K} \alpha_{k,m}^{D_x u} u_{k-1/2} = \frac{\gamma_{1,m}}{h} u \mid_{m/2} + \gamma_{2,m} \frac{\partial u}{\partial x}\bigg|_{m/2} + \gamma_{3,m} h \frac{\partial^2 u}{\partial x^2}\bigg|_{m/2}$$

$$(S_x u)_{m/2} = \sum_{k=1}^{K} \alpha_{k,m}^{S_x u} u_{k-1/2} = \gamma_{1,m} u \mid_{m/2} + \gamma_{2,m} h \frac{\partial u}{\partial x}\bigg|_{m/2} + \gamma_{3,m} h^2 \frac{\partial^2 u}{\partial x^2}\bigg|_{m/2} \tag{30}$$

Coefficients γ are obtained simply as $\gamma_{l,m} = \sum_{k=1}^{K} \frac{k^{l-1}}{l!} \alpha_{k,m}$ These coefficients of the Taylor expansion calculated for α that have been obtained in the experiment with $K = 5, M = 2$ are shown in the Table IV.

Table IV. Coefficients of the Taylor expansion calculated in the experiment with $K = 5, M = 2$.

Operator	m	Second order			Fourth order		
		γ_1	γ_2	γ_3	γ_1	γ_2	γ_3
$(D_x u)$	1	0.0463	1.0490	0.0393	0.0191	1.1128	0.2183
$(D_x u)$	2	-0.0137	0.8763	0.1620	-0.0375	1.0366	0.0678
$(D_x p)$	1	0.0020	1.4587	-0.6313	-0.0048	1.2669	0.3334
$(D_x p)$	2	0.0041	1.2567	0.0930	-0.0065	1.0930	0.2289
$(D_y v)$	1	0.1225	0.6699	0.5234	0.0626	1.0514	0.0667
$(D_y v)$	2	-0.1252	1.3678	-0.3923	-0.0145	0.9344	-0.0462
$(D_y p)$	1	-0.0019	1.3599	0.2992	0.0013	0.9191	-0.2578
$(D_y p)$	2	0.0475	1.6242	0.0138	0.0123	0.1542	-0.4943
$(S_x u)$	1	1.2711	0.4103	0.5558	1.2049	-0.3784	0.5495
$(S_x p)$	1	0.8935	0.0345	0.0006	0.9211	-0.2167	-0.1967
$(S_x p)$	2	0.7497	0.1913	-0.1252	0.7632	0.1147	-0.3810
$(S_y v)$	1	1.1307	-0.1101	0.1195	1.1312	-0.0580	-0.0097
$(S_y v)$	2	1.1259	-0.0702	0.2285	1.0686	0.0113	0.0139
$(S_y p)$	2	1.5921	-1.2996	1.2440	0.8134	0.0573	0.0728

Taylor expansions show that obtained expressions do not approximate corresponding operators. So far, the value of γ_1 is never zero, the derivative near boundary is always approximated with the minus first order due to the presence $\frac{\gamma_1}{h}$ in it's Taylor expansion. Values of γ_1 may be as high as 0.12, that is comparable with the coefficient in front of the first derivative γ_2, which is, in turn, may be as low as 0.15 and as high as 1.6.

The first coefficient of the Taylor expansion of the interpolation near boundary varies also in a wide range between 0.7 and 1.6. That means interpolation multiplies simultaneously the function by some value which is different from 1.

Similar phenomenon have been seen in [23] in experiments with numerically approximated waves propagating with a wrong wave velocity. The data assimilation and control of the discretization of the derivatives near the boundary can not modify numerical wave velocity. The only way for this control to get a better solution consists in modifying the size of the domain by deforming the grid cell adjacent to boundary. The boundary cell of the domain is adapted by data assimilation to ensure the wave with numerical velocity propagates the modified interval in the same time that the exact wave propagates the exact interval. So far, the control can not correct the error in the wave velocity, it commits another error in the interval size in order to compensate the first one.

5. Conclusion

The purpose of this paper is to study the variational data assimilation procedure applied for identification of the optimal parametrization of derivatives and interpolation operators near the boundary on the example of a linear shallow water model.

Comparing this procedure with now well developed data assimilation intended to identify optimal initial data, we can say there are both common points and differences as well.

Tangent (7) and adjoint (13) models are composed of two types of operators: operators

acting in the space of perturbations of $\delta u, \delta v$ and $\delta \eta$, and operators acting on the variations of the control coefficients $\delta \alpha$ directly. The first type, that constitute the 3×3 block in the matrix of (8), governs the evolution of a small perturbation by the model's dynamics. This term is common for any data assimilation no matter what parameter we want to identify. The second type, (9), determines the way how the uncertainty is introduced into the model. So, if we intend to identify an optimal discretization of operators near the boundary for a model with an existing adjoint developed for data assimilation and identification of initial point, we can use this adjoint as 3×3 block in the matrix of (8) because this part is common for any data assimilation. However, the fourth column of this matrix (9) must be developed from the beginning because this column is specific for the particular control parameter. As we have seen, the development of the fourth column is as complex as the development of the 3×3 block even for a simple linear model. For a non-linear shallow-water model, the development of the fourth column will become even more complex because non-linear terms contribute more to operators responsible for introducing of uncertainty into the model.

Using automatic code differentiation for generation of the adjoint model may also become more complex in the case of controlling the model's internal parameters. The adjoint model is not standard, the model must be rewritten in order to satisfy the automatic differentiation requirements.

Another difference consists in the number of control parameters. The dimension of initial point of the model is usually equal to the dimension of the model's state variable. Contrary to this, when we control boundary parametrization, the dimension of control variables is much lower than the dimension of the model's variable because the dimension of the control is proportional to the length of the boundary of the domain, while the dimension of the model's state relates to the area of the domain. That means the quantity of controlled parameters and the dimension of the gradient of cost function may be much lower than the quantity of variables in the models state.

Taking into account mentioned technical difficulties in development of the adjoint, it may be reasonable to try to calculate the gradient by some other method beginning with the simplest finite difference method.

As we have seen in this paper, controlling the discretization of operators in the boundary region allows to obtain better results than controlling boundary conditions of the model. The possibility to control more than two parameters, as it is the case when boundary conditions are controlled, allows us to get lower difference with the reference model. Enlarging the boundary region, where derivatives and interpolations are controlled, allows us to obtain the solution on the coarse grid which is closer to the reference one than the solution on the 3 times finer grid. This is valid as for the stationary solution with under-resolved Munk layer, and for the non-stationary solution representing inertia-gravity and Rossby waves.

In the case of the stationary solution with Munk layer, only one point near the boundary may be controlled allowing to avoid spurious oscillations due to under-resolved boundary layer. To obtain the solution almost indistinguishable from the reference one, we have to enlarge boundary region and to control operators at 4 nodes near boundary.

Uniform control of all coefficients α all the long the boundary is not the best choice. Mass and momentum fluxes can not be balanced on the boundary and the assimilation procedure have to look for the optimum in the domain where the scheme is instable. Consequently, at least α_0 should be allowed to be variable and particular in each point near the boundary in order to allow the flux across the boundary to be compensated. This requirement substantially

increases the dimension of the control variables, but it is necessary to obtain a stable scheme: we must at least allow control coefficients α_0 to depend on coordinate: on the latitude y corresponding to the index j for operators S_x, D_x and G_x, and on the longitude x or index i for operators S_y, D_y and G_y.

Concerning the choice of the assimilation window, we must take into account that assimilation using short windows may provide an unstable scheme while using long windows is computationally more expensive. Moreover, better precision we want, longer windows we must use. We have seen in the experiment with the stationary solution that 10 iterations are sufficient for the minimization to converge when $T = 0.2$ days resulting, however, in an unstable scheme. When assimilation window $T = 2$ days is used, ten iterations are no longer sufficient for the minimization to converge. We need at least 50 iterations to bring the cost funtion close to the minimum and to keep a stable scheme. Performing 150 supplementary iterations we win a little in the cost function but the scheme looses the stability. Assimilation window $T = 20$ days ensures the stability of the scheme during 1000 iterations, but after that we obtain an instable scheme also.

To combine rapid convergence of the assimilation with short T and the stability of obtained scheme with long T we may use variable assimilation windows. Performing few iterations with short T we obtain the set of coefficients α that provides low cost function value. This set is corrected by assimilation with longer T in order to ensure the stability of the scheme.

When we vary the number of control parameters K and M, we see that parameter K influences only a little the assimilation quality. The difference between the solution produced by the model with optimal discretization of operators in the boundary region and the reference solution is very little sensitive to the number of coefficients α used in the approximation of derivatives and interpolations near the boundary. On the other hand, the difference $\xi(t)$ is sensitive to M, that is to the number of grid nodes near boundary in which the discretization is controlled by data assimilation. In other words, to the width of the boundary region. We have seen, despite under-resolution of the Munk layer by the model's grid, optimal discretization allows us to obtain the solution as close to the reference one as we want. It is sufficient to take a sufficient M.

In the experiment with waves, parameter M influences also the assimilation results. One have to take M at least equal to 2 in order to compensate the error in wave velocities.

It has been shown also that the sea surface height may be used as the only observable variable. It is possible to reconstruct discretization of all operators using observations of η only. However, assimilation in this case becomes 5 times more expensive due to slower convergence of the minimization procedure.

Analyzing coefficients α obtained in assimilation, we can note that optimal parametrization of interpolations and derivatives near the boundary may approximate nothing in classical sense, i.e. it may not be valid for an arbitrary function. We have seen here that coefficients γ of the Taylor expansion of obtained expressions may be as high as 1.6 and as low as 0.15 instead of 1. Moreover, the approximation of the derivative has minus first order. Hence, we must take into account that coefficients found by data assimilation do not approximate interpolations or derivatives in general. They are valid for given model's parameters only.

Thus, there exist a possibility to apply data assimilation technique to identification of optimal discretizations of the model's operators in the boundary region. Optimal discretization allows us to correct such errors of the numerical scheme as under-resolved boundary layer and wrong wave velocity.

REFERENCES

1. Lorenz E. Deterministic non periodic flow. *Journal of the Atmospheric Sciences* 1963; **20**:130–141.
2. Lions JL. *Contrôle optimal de systèmes gouvernés pas des équations aux dérivées parielles.* Dunod, 1968.
3. Marchuk G. Formulation of theory of perturbations for complicated models. *Appl. Math. Optimization* 1975; **2**:1–33.
4. Le Dimet FX. A general formalism of variational analysis. *Technical Report OK 73091*, CIMMS report, Normann 1982.
5. Le Dimet FX, Talagrand O. Variational algorithm for analysis and assimilation of meteorological observations. theoretical aspects. *Tellus* 1986; **38A**:97–110.
6. Verron J, Blayo E. The no-slip condition and separation of western boundary currents. *Journal of the Physical Oceanography* 1996; **26**(9):1938–1951.
7. Adcroft A, Marshall D. How slippery are piecewise-constant coastlines in numerical ocean models? *Tellus* 1998; **50A**:95–108.
8. Holland W. Baroclinic and topographic influences on the transport in western boundary currents. *Geophys.Fluid Dyn.* 1973; **4**:187–210.
9. Eby M, Holloway G. Sensitivity of a large-scale ocean model to a parametrisation of topographic stress. *Journal of the Physical Oceanography* 1994; **24**:2577–2587.
10. Losch M, Heimbach P. Adjoint sensitivity of an ocean general circulation model to bottom topography. *Journal of the Physical Oceanography* 2007; **37(2)**:377–393.
11. Bryan F, Wainer I, Holland WR. Sensitivity of the tropical atlantic circulation to specification of wind stress climatology. *JGR* 1995; **100**:24 729–24 744.
12. Milliff R, Large W, Morzel J, Danabasoglu G, Chin T. Ocean general circulation model sensitivity to forcing from scatterometer winds. *JGR-Oceans* 1998; **104**(C5):11 337.
13. Bryan F. Parameter sensitivity of primitive equation ocean general circulation models. *Journal of the Physical Oceanography* 1987; **17**:970–985.
14. Losch M, Adcroft A, Campin JM. How sensitive are coarse general circulation models to fundamental approximations in the equations of motion? *Journal of the Physical Oceanography* 2004; **34**(1):306–319.
15. Losch M, Wunsch C. Bottom topography as a control variable in an ocean model. *J. Atmospheric and Oceanic Technology* 2003; **20**:1685–1696.
16. Kazantsev E. Identification of optimal topography by variational data assimilation. *Ocean Modelling* 2009; Submitted.
17. Shulman I. Local data assimilation in specification of open boundary conditions. *J. of Atmospheric and Oceanic Technology* 1997; **14**:1409–1419.
18. Shulman I, Lewis JK, Blumberg AF, Kim BN. Optimized boundary conditions and data assimilation with application to the m2 tide in the yellow sea. *J. of Atmospheric and Oceanic Technology* 1998; **15**(4):1066–1071.
19. Taillandier V, Echevin V, Mortier L, Devenon JL. Controlling boundary conditions with a four-dimensional variational data-assimilation method in a non-stratified open coastal model. *Ocean Dynamics* 2004; **54**(2):284–298.
20. Chen H, Lin S, Wang H, LC Fang. Estimation of two-sided boundary conditions for twodimensional inverse heat conduction problems. *Int. J. Heat Mass Transfer* 2002; **45**:15–43.
21. Gillijns S, Moor BD. Joint state and boundary condition estimation in linear data assimilation using basis function expansion. *Modelling, Identification, and Control*, Bruzzone L (ed.), 2007.
22. Le Dimet FX, Ouberdous M. Retrieval of balanced fields: an optimal control method. *Tellus A* 1993; **45**(5):449–461.
23. Kazantsev E. Identification of an optimal boundary approximation by variational data assimilation. *J.Comp.Phys* 2009; Submitted.
24. Leredde Y, Lellouche JM, Devenon JL, Dekeyser I. On initial, boundary conditions and viscosity coefficient control for Burgers' equation. *International Journal for Numerical Methods in Fluids* 1998; **28**(1):113–128.
25. Lellouche J, Devenon J, Dekeyser I. Boundary control on Burgers equation. A numerical approach. *Comput. Math. Appl.* 1994; **28**:33–44.
26. Gill A. *Atmosphere-Ocean Dynamics.* Academic Press, 1982.
27. Pedlosky J. *Geophysical Fluid Dynamics.* 2d ed., Springer-Verlag, 1987.
28. Gilbert J, Lemarechal C. Some numerical experiments with variable storage quasi-newton algorithms. *Mathematical programming* 1989; **45**:407–435.
29. Le Provost C, Verron J. Wind-driven ocean circulation transition to barotropic instability. *Dyn.Atmos.Oceans* 1987; **11**:175–201.
30. Munk WH. On the wind-driven ocean circulation. *Journal of Meteorology* 1950; **7**:3–29.
31. Bryan K, Manabe S, Pacanowski R. A global ocean-atmosphere climate model. part ii. the oceanic circulation. *Journal of the Physical Oceanography* 1975; **5**(1):30–46.

32. Griffies SM, Pacanowski RC, Hallberg RW. Spurious diapycnal mixing associated with advection in a z-coordinate ocean model. *Monthly Weather Review* 2000; **128**(3):538–564.