A Fast and Generic Method to Identify Parameters in Complex and Embedded Geophysical Models: The Example of Turbulent Mixing in the Ocean

Clement Aldebert, Guillaume Koenig, Melika Baklouti, Philippe Fraunié, Jean-Luc Devenon

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2020MS002245

Special Section:
Dynamical cores of oceanic models across all scales and their evaluation

A Fast and Generic Method to Identify Parameters in Complex and Embedded Geophysical Models: The Example of Turbulent Mixing in the Ocean

Clement Aldebert[1], Guillaume Koenig[1], Melika Baklouti[1], Philippe Fraunié[2], and Jean-Luc Devenon[1]

[1]Aix Marseille University, Université de Toulon, CNRS, IRD, MIO UM, Marseille, France, [2]University of Toulon, Aix Marseille University, CNRS/INSU, IRD, MIO UM, Mediterranean Institute of Oceanography, Toulon, France

Abstract Geophysical models make predictions relying on parameter values to be estimated from data. However, existing methods are costly because they require either many runs of the complex geophysical model or to implement an adjoint model. Here, we propose an alternative approach based on optimal control theory which is the simultaneous perturbations stochastic approximation (SPSA). This gradient-descent method is generic and easy to implement, and its computational cost does not increase with the number of parameters to optimize. This study aims at highlighting the potential of SPSA for parameter identification in geophysical models. Through the example of vertical turbulent mixing in the upper ocean, we show with twin experiments that the method could successfully identify parameter values that minimize model-data discrepancy. The efficient and easy-to-get results provided by SPSA in this study should pave the way for a broader use of parameter identification in the complex and embedded models commonly used in geophysical sciences.

Plain Language Summary Predictions on the past and future state of geophysical systems are made through mathematical models, which rely on numerous constant values (parameters) to be calibrated from prior knowledge and available data. Fine-tuning those parameter values is one of the major means of improving the accuracy of model predictions. To achieve that goal for complex geophysical models in which multiple scales and processes are nested, existing methods are limited by either (a) an important computational cost or (b) an important cost in terms of development and implementation of an adjoint model. Here, we highlight a method from optimal control theory, called simultaneous perturbations stochastic approximation (SPSA). This generic method is easy to implement, and its computational cost is comparatively low. To show the potential of SPSA for parameter identification in geophysical science, we apply it to the example of wind-induced turbulent mixing near the ocean surface. Using the approach of twin experiments, we show that the method can successfully tune the parameter values to minimize the discrepancy between model predictions and empirical data. The efficient and easy-to-get results provided by SPSA in this study pave the way for a broader use of parameter identification in the complex models commonly used in geophysical sciences.

Clement Aldebert is now working as a (New-Zealand registered) private contractor, trading as Bluework-Ocean Science Services.

Key Points:

- Parameters identification is a key to reduce model uncertainty in geophysical models
- We present a method from optimal control called simultaneous perturbations stochastic approximation (SPSA) that is easy to implement and fast to run
- We illustrate the potential of SPSA with a simplified example of turbulent mixing in the ocean

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

C. Aldebert,
clement.aldebert.phd@gmail.com

Citation:

Aldebert, C., Koenig, G., Baklouti, M., Fraunié, P., & Devenon, J.-L. (2021). A fast and generic method to identify parameters in complex and embedded geophysical models: The example of turbulent mixing in the ocean. *Journal of Advances in Modeling Earth Systems*, *13*, e2020MS002245. https://doi.org/10.1029/2020MS002245

Received 12 JUL 2020
Accepted 22 JUN 2021

1. Introduction

Geophysical systems are generally described by nonlinear mathematical models. These models often involve partial differential equations like the Navier-Stokes equations, and embed sub-models describing various processes, such as sub-grid turbulent mixing or biological processes (e.g., growth and interactions of living organisms). The resulting models are costly to develop and to run, bringing together scientists from different disciplines. Moreover, the resulting model predictions remain sensitive to various forms of uncertainty (Arhonditsis & Brett, 2004; Foley, 2010; Lermusiaux et al., 2006; Palmer et al., 2005). One of them lies in the uncertainty generated by the model parameters.

Some parameters are well-quantified physical or physiological constants (e.g., earth acceleration), others are empirical constants (e.g., macroscopic description of unresolved smaller scales) that have to be tuned based on empirical observations. Observations provide constraints to identify parameter values leading to predictions that best fit data. Identifying those optimal parameter values by optimal control theory (Lions, 1968) is of interest for many geophysical applications (Plessix, 2006).

Table 1
Overview of Parameter Identification Methods, and Their Pros and Cons if Applied to Complex Geophysical Models That Are Costly to Program and Run

Type	Method	Pros	Cons
Trials-errors		Easy, only model runs.	Needs many runs, unlikely to succeed without a methodology.
Stochastic, global	Simulated-annealing	Generic, easy to implement, noisy data.	Needs many runs, even for a few parameters.
	Genetic algorithms	Generic, many parameters, noisy data.	Needs many runs, hard to tune.
	Hamiltonian Monte-Carlo	Generic, Bayesian framework.	Needs many runs.
Local gradient descent	Adjoint	Cheap to run, explicit gradient, many parameters.	Linked to the model (i.e., model-dependent), sensitive to non-linearities.
	Finite difference	Generic, easy to implement, almost exact gradient.	Number of runs proportional to the number of parameters.
	Simultaneous Perturbation Stochastic Approximation	Generic, easy to implement, cheap to run, many parameters, less noise-sensitive.	Approximated gradient.

In geophysical models, the high numerical cost of a single model run is problematic for parameter identification. Therefore, an optimization by trials and errors, or with stochastic optimization methods like simulated-annealing (Zhigljavsky & Zilinskas, 2008), genetic algorithms (Mitchell, 1998), and Monte-Carlo algorithms in Bayesian statistics (McElreath, 2015), cannot be considered as appropriate solutions as they require many model runs to be efficient (Table 1).

Less model runs are required by the descent algorithm. The local steepest descent strategy relies on the gradient of the cost function—function quantifying the model-data discrepancy—with respect to parameters. This gradient can be estimated by finite difference approximation, which is easy to implement by perturbating systematically the model parameters one by one, and start it again each step of the descent algorithm. Unfortunately, this requires an increasing number of model runs as the number of parameters to optimize increases, and becomes rapidly prohibitive for a large number of dimensions of parameter space.

To overcome those limitations, algorithms based on an adjoint model have been introduced since the pioneering works of Lions (1968). An adjoint is a code of complexity similar to that of (but closely linked to) the model, which gives the gradient of a cost function with respect to the control parameters (model parameters, or initial or boundary conditions) (Plessix, 2006).

Table 2
Inventory of the Main General Circulation Models With Mention of the Existence or Absence of an Adjoint Code

Model	Boundary conditions	Initial conditions	Other parameters
ADCIRC[a]	No	No	No
CROCO[b]	No	No	No
Delft3D[c]	No	No	No
FVCOM[d]	No	No	No
MARS[e]	No	No	No
MITgcm[f]	Yes	Yes	Yes
NEMO[g]	Yes	Yes	Yes
POM[h]	Yes	Yes	Yes
ROMS[i]	Yes	Yes	No

[a]http://adcirc.org/. [b]https://www.croco-ocean.org/documentation/. [c]https://oss.deltares.nl/web/delft3d/manuals. [d]Alleged from the description on http://fvcom.smast.umassd.edu/fvcom/. [e]https://wwz.ifremer.fr/mars3d/content/download/77020/file/2009_11_22_DocMARS_GB.pdf. [f]https://mitgcm.readthedocs.io/en/latest/related_projects/related_projects.html.[g]https://ljk.imag.fr/membres/Franck.Vigilant/Documents/ReferenceManual-UserGuide-NEMOTangentandAdjointModels.pdf. [h]Peng et al. (2007). [i]Moore et al. (2004).

This adjoint is obtained by calculations by hand (Leredde et al., 1999) or by Automatic Differentiation (Heimbach et al., 2005). As for the finite difference method, an assumption of local linearity is made as the adjoint is the adjoint of the local tangential linear approximation of the model. This can induce some limitations if the mapping between model results and parameters becomes too nonlinear and by this way generating inaccuracies in the gradient of the cost function (Talagrand & Courtier, 1987). Presently, a certain number of circulation models used by oceanographers are indeed developed with these optional adjoint modules. Nevertheless, this is far from being a general rule (see Table 2 for few examples).

In this context, another promising approach seems to be the Simultaneous Perturbations Stochastic Approximation (SPSA; Spall, 1998, 2012), which is as generic and easy to implement as the finite difference approximation, but uses a cheap approximation of the cost function gradient regardless of the number of parameters to be optimized. At this stage, it is important to emphasize that our goal is to identify optimal parameter values that are intended to be held constant during the entire

model simulation. This method is different from the one achieved by the popular data assimilation methods based on (Ensemble) Kalman Filter (KF or EnKF), which are of considerable interest to increase the predictive accuracy of hindcasts and forecasts. Indeed, these techniques assimilate sequentially observational information during the course of model simulation as new data become available, to improve the predicted system state or evolving estimates of model parameters (Evensen, 2009; Sun et al., 2016). Conversely, our inverse approach is based on the whole data set available from the observation period to optimize parameter values at once.

The SPSA method was proposed by Spall (1998, 2012) and has been used in the fields of engineering and optimal control, but this promising approach has not yet received the attention it deserves in geophysical sciences. The only recent exceptions we are aware of are an application to estimate bottom friction (Boutet, 2015) and another one to optimize the parameters of the gain matrix of a reduced-rank Kalman filter (Hoang & Baraille, 2011). Therefore, our purpose in this study is to present the potential of the SPSA method to estimate empirical parameter values in an embedded geophysical model. To achieve this goal, we use as an example a model of turbulent closure in the ocean, the Turbulent Kinetic Energy (TKE) model (Gaspar et al., 1990). Indeed, parameters identification for turbulent closure schemes in ocean circulation models is an open issue in modern oceanography (Dekeyser et al., 2004; Leredde et al., 1999, 2002). The TKE model, when used, is classically embedded into three-dimensional (3D) ocean circulation models that solve hydrodynamic and hydrological fields such as NEMO (Madec et al., 2017). For the sake of simplicity of our proof-of-concept, we embed it into a one-dimension vertical (1DV) hydrodynamical model forced by real momentum and radiative fluxes at the sea surface and applied at the DYFAMED station in the Mediterranean Sea. This study is part of a logical sequence where we apply SPSA to various types of geophysical problems: the parametrization of a biological model of Ordinary Differential Equations advected by Lagrangian methods (Messié et al., 2020), the optimization of boundary conditions for coastal hydrodynamics modeling (Koenig et al., 2020), and here the parametrization of a model of turbulent closure embedded inside a 1DV hydrodynamical model.

The study is organized as follows. Next section introduces the optimization method and presents its computational advantages for optimizing a large number of parameters through a pedagogical example. Moving to our example of geophysical application, Section 3 presents the 1DV model with TKE closure scheme and formulates the optimization problem to solve a two-parameters example. Pushing forward, Section 4 presents our statistical results on the method's ability to optimize eight empirical parameters of the TKE model. These results and the method are discussed in Section 5 through the lens of its practical use in modeling of complex geophysical systems, before concluding with ongoing perspectives of application of SPSA in geophysical sciences.

2. Optimization Method

2.1. Gradient Descent With Nesterov Momentum

The optimization consists in identifying the set of control parameters $\boldsymbol{\theta}$ that minimizes a cost function $J\left(\boldsymbol{\theta}\right)$, which quantifies model-data discrepancy. The lower $J\left(\boldsymbol{\theta}\right)$ is, the closer model predictions are to data.

Optimal parameter values $\boldsymbol{\theta}^{\text{opt}}$ minimize the cost function, with $J^{\text{opt}} := J\left(\boldsymbol{\theta}^{\text{opt}}\right)$ being positive due to data noise. To find the optimal parameter values, we use a gradient descent algorithm. Gradient descent is an iterative procedure starting from an arbitrary initial guess of the parameter values $\boldsymbol{\theta}^{(0)}$:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - a^{(k)}\boldsymbol{z}^{(k)}. \tag{1}$$

The gain parameter $a^{(k)}$, also called learning rate in machine learning, scales the step made at each iteration. This step is made in the direction indicated by the Nesterov momentum $\boldsymbol{z}^{(k)}$, which has the same dimension as the gradient of the cost function $\nabla J\left(\boldsymbol{\theta}\right)$. Nesterov momentum is a variant of momentum based on (a) the direction of the last step made (i.e., its previous value $\boldsymbol{z}^{(k-1)}$), and (b) the gradient of the cost function at the point where we would arrive if we repeat a step in that direction ($\nabla J(\boldsymbol{\theta}^{(k)} - a^{(k)}\boldsymbol{z}^{(k-1)})$):

$$z^{(k)} = \beta z^{(k-1)} + \nabla J\left(\theta^{(k)} - a^{(k)} z^{(k-1)}\right). \tag{2}$$

Momentum is initialized with $z^{(-1)} = \mathbf{0}$, meaning that the algorithm starts with an empty memory. The momentum coefficient $\beta \in [0,1]$ tunes the level of memory of the algorithm. A properly tuned momentum speeds up the convergence, like a ball rolling down a hill: It accumulates momentum while heading in the same direction, slows down when reaching the other side of a valley, and again accelerates while following the bottom of the valley to reach its lowest point. The algorithm anticipates by estimating the gradient at the point where we would arrive if we keep going in the direction of the last step, rather than at the current location in the parameter space. This anticipation allows the algorithm to deal more smoothly with abrupt changes in the gradient, like when the bottom of a valley is reached (Sutskever et al., 2013). The algorithm continues until the desired level of convergence ($J(\theta) < J^{\text{tol}}$) or the arbitrary maximum number of iterations ($k_{\max}$) is reached. Note that many other alternative stopping criteria can also be considered. The estimated optimal parameters $\hat{\theta}$ are the ones giving the lowest $J(\theta)$ among all iterations, namely the ones such as $J(\theta) < J^{\text{tol}}$ if this stopping criterion is used.

Spall (1998) argued that the gain parameter $a^{(k)}$ should decrease at each optimization step following a decreasing sequence that has to be parameterized with some constraints, to ensure convergence. However, preliminary tests showed that a constant value $a^{(k)} = a$ performs equally well in our case, and we kept this simpler solution for our study. The only constraints are that a should be large enough to avoid making too many unnecessary steps, but still small enough to prevent the algorithm to diverge.

2.2. Gradient Estimation

The cost function gradient $\nabla J(\theta)$ can be estimated by model runs giving cost function evaluations for different parameter values. A classical approach is the centered finite difference algorithm:

$$\nabla J(\theta^{(k)}) = \begin{pmatrix} \frac{J(\theta^{(k)} + c^{(k)}\delta\theta_1^{(k)}) - J(\theta^{(k)} - c^{(k)}\delta\theta_1^{(k)})}{2c^{(k)}} \\ \vdots \\ \frac{J(\theta^{(k)} + c^{(k)}\delta\theta_i^{(k)}) - J(\theta^{(k)} - c^{(k)}\delta\theta_i^{(k)})}{2c^{(k)}} \\ \vdots \\ \frac{J(\theta^{(k)} + c^{(k)}\delta\theta_p^{(k)}) - J(\theta^{(k)} - c^{(k)}\delta\theta_p^{(k)})}{2c^{(k)}} \end{pmatrix}, \delta\theta_i^{(k)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3}$$

where $c^{(k)} > 0$ is a stepsize and $\delta\theta_i$ a vector of zeros, except for its i^{th} element which equals 1. For each parameter, the model is run two times to estimate the effect of a change in this parameter. Therefore, evaluating the gradient with respect to p parameters requires $2p$ model runs.

The number of model runs is a limiting resource in optimization, especially for geophysical models. Thus, the approach of SPSA proposed by Spall (1998, 2012) in optimal control is interesting as it approximate the gradient $\nabla J(\theta^{(k)})$ by $\tilde{\nabla} J(\theta^{(k)})$, which is calculated with only 2 model runs by altering all the parameters together:

$$\tilde{\nabla} J(\theta^{(k)}) = \frac{J(\theta^{(k)} + c^{(k)}\Delta^{(\mathbf{k})}) - J(\theta^{(k)} - c^{(k)}\Delta^{(\mathbf{k})})}{2c^{(k)}} \Delta^{(\mathbf{k})}, \Delta^{(\mathbf{k})} = \begin{pmatrix} \Delta_1^{(k)} \\ \vdots \\ \Delta_i^{(k)} \\ \vdots \\ \Delta_p^{(k)} \end{pmatrix} \tag{4}$$

where elements $\Delta_i^{(k)} = \pm 1$ are randomly drawn with equal probability (Bernoulli law). Spall (1998) argued that the stepsize $c^{(k)}$ should follow a decreasing sequence to ensure convergence. However, our preliminary tests showed that a small constant value $c^{(k)} = c$ performs equally well and we kept this simpler solution.

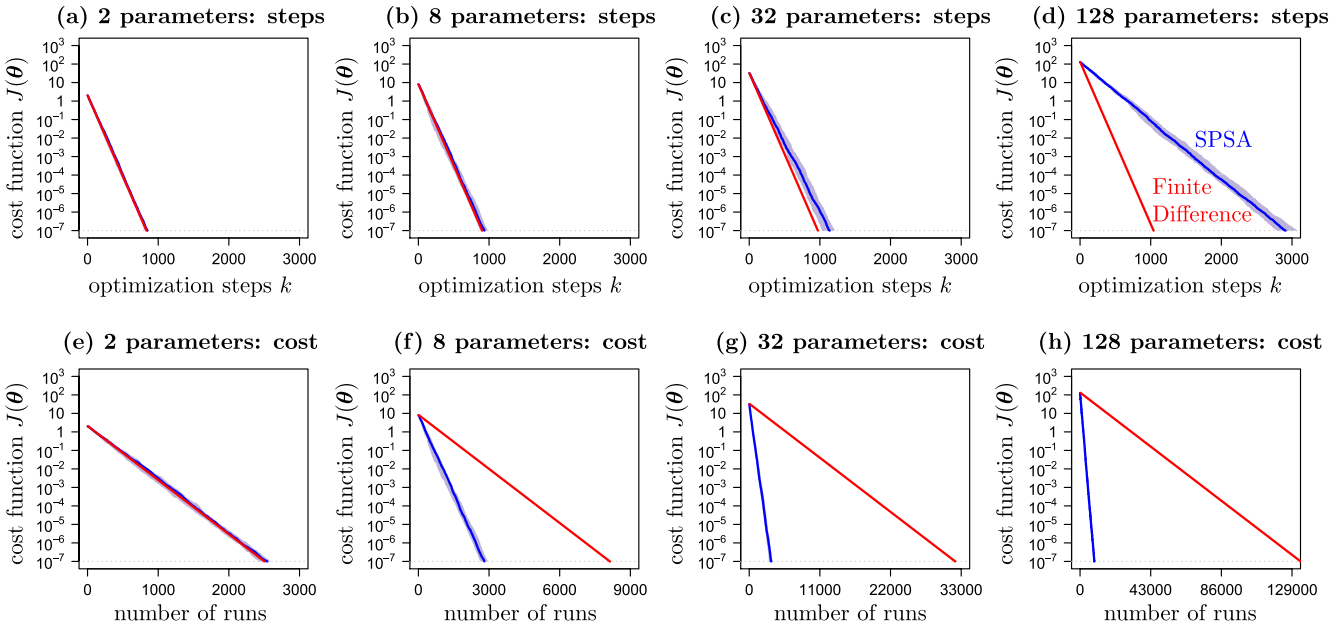

Figure 1. Example comparing 10 stochastic minimizations performed by simultaneous perturbations stochastic approximation (blue, line: median, shaded area: range from minimum to maximum) with a deterministic minimization by finite difference (red), with different numbers of parameters to optimize (2, 8, 32, and 128). For each number of parameters, the decrease of the cost function is presented as a function of the number of optimization steps (a–d) and as a function of the number calculations of the cost function $J(\boldsymbol{\theta})$ (e–h).

The SPSA approximates the gradient by a directional derivative. Spall (1998) argued that "one properly chosen simultaneous random change in all the variables in a problem provides as much information *for optimization* as a full set of one-at-a-time changes of each variable" (p. 448). Figure 1 compares the speed of convergence of SPSA and finite difference algorithms to find the minimum $\boldsymbol{\theta}^{\text{opt}} = \mathbf{0}$ of a p-dimensional quadratic cost function $J(\boldsymbol{\theta}) = \sum_{i=1}^{p} \theta_i^2$ for different numbers of parameters p ($a = 0.002, c = 0.01, \beta = 0.6$, $J^{\text{tol}} = 10^{-7}, \boldsymbol{\theta}^{(0)} = \mathbf{1}$). For small numbers of parameters (2 and 8), the loss of accuracy of the SPSA method does not change the speed of convergence in terms of number of optimization steps (Figures 1a and 1b). The SPSA can even be a bit faster than finite difference "by-chance" due to randomness in the algorithm, but it can also be a bit slower for the same reason. For larger numbers of parameters (32 and 128), the loss of accuracy of the SPSA method implies that more optimization steps are required to achieve the same level of convergence (Figures 1c and 1d). Indeed, as the gradient estimation by SPSA is made alongside a random subspace of dimension 1, more optimization steps are required to make the successive averaging of those one-dimensional estimates by Nesterov momentum converging toward a satisfying approximation of the p -dimensional gradient. Despite this need of more optimization steps to counterbalance the gradient approximation when the number of parameters increases, SPSA method performs increasingly better than finite difference in terms of number of cost function calculations required by the method to reach the minimum (Figures 1e–1h). Indeed, for p parameters, the relative gain of using SPSA to estimate a gradient is roughly a factor p. This gain is more important *for optimization* than the small loss in accuracy of the gradient approximation.

The informed reader may notice some resemblance between SPSA and Stochastic Gradient Descent (SGD) methods used in deep learning (Bottou, 2010, 2012). Indeed, both methods rely on stochastic sampling to deal with the respective dimensionality challenge in their field of application. For SGD, the dimensionality challenge is the huge amount of data available to train a neural network, the numerical cost of estimating the cost function (prediction-observation discrepancy) being proportional to the size of the training data set as each input data corresponds to a run of the neural network. Therefore, SGD methods rely on estimates of the gradient of the cost function using only a random subset of available data. Conversely, for SPSA, the dimensionality challenge is the number of parameters, which increases the numerical cost of estimating

the gradient of the cost function by finite differences. Therefore, SPSA method relies on estimates of the gradient alongside a subset of the parameter space to require less model runs.

Moving forward from this pedagogical example, we now ask whether the previous conclusion still holds for a complex geophysical model or not. An optimization problem for parameter identification is usually more complicated than the convex quadratic function used above. On the one hand, the non-linearity of the model can increase the effect of the loss of accuracy of the gradient approximation. On the other hand, each calculation of the cost function implies to run the model with new parameters, making the benefit of requiring less model runs with SPSA even more valuable.

3. An Example of Complex Geophysical Model: A 1DV Model at DYFAMED Station

3.1. The Model

The one-dimension hydrodynamical model used in this study is extensively described in Gaspar et al. (1990). This model solves the conservation equations for heat, salinity, and momentum:

$$\frac{\partial \overline{T}}{\partial t} = \frac{F_{\text{sol}}}{\rho_0 c_p}\frac{\partial I}{\partial z} - \frac{\partial \overline{T'w'}}{\partial z}, \tag{5}$$

$$\frac{\partial \overline{S}}{\partial t} = -\frac{\partial \overline{S'w'}}{\partial z}, \tag{6}$$

$$\frac{\partial \overline{\mathbf{U}}}{\partial t} = -f\mathbf{k}\times\overline{\mathbf{U}} - \frac{\partial \overline{\mathbf{U}'w'}}{\partial z}, \tag{7}$$

where T and S are respectively the temperature and the salinity of water, and $\mathbf{U} = (u, v)$ and w the horizontal and vertical velocities of water, respectively. For each of those quantities, we use Reynold's decomposition $T = \overline{T} + T'$ (same for S, $\mathbf{U}$ and w), where $\overline{T}$ is the average value that can be modeled (we assume no convection with $\overline{w} = 0$) and T' are the turbulent fluctuations that are not resolved by the model. ρ_0 is a reference value of density ρ and c_p is the specific heat of seawater; F_{sol} is the downward flux of sunlight absorbed at the sea surface and $I(z)$ the fraction of F_{sol} reaching depth z in the water column; f is the Coriolis parameter; and $\mathbf{k}$ the vertical unit vector. Heat, salt, and momentum turbulent fluxes at the sea surface are detailed in Gaspar et al. (1990).

The parameterization of turbulent vertical fluxes relies on the concept of eddy diffusivity:

$$\overline{T'w'} = -K_h\frac{\partial \overline{T}}{\partial z}, \overline{S'w'} = -K_s\frac{\partial \overline{S}}{\partial z}, \overline{\mathbf{U}'w'} = -K_m\frac{\partial \overline{\mathbf{U}}}{\partial z}, \tag{8}$$

where K_m, K_s and K_h are the eddy diffusivities, respectively for momentum, salt, and heat. These diffusivities are related to the turbulent kinetic energy (hereafter abridged TKE) $\overline{e} = 0.5(\overline{u'^2 + v'^2 + w'^2})$ through:

$$K_m = \max\left(K_{m_{\min}}, c_k\, l_k\, \sqrt{\overline{e}}\right), \tag{9}$$

$$K_s = K_h = \frac{K_m}{Pr_t}, \tag{10}$$

where c_k is a constant to be determined, $K_{m_{\min}}$ is a minimal value for K_m, l_k a mixing length and Pr_t the turbulent Prandtl number. The conservation equation for the TKE writes:

$$\frac{\partial \overline{e}}{\partial t} = \frac{\partial}{\partial z}\left(K_e\frac{\partial \overline{e}}{\partial z}\right) - \overline{\mathbf{U}'w'}\cdot\frac{\partial \overline{\mathbf{U}}}{\partial z} + \overline{b'w'} - \varepsilon \tag{11}$$

Table 3
Model Parameters and Reference Values, From Gaspar et al. (1990)

Parameter	Value	Unit	Meaning
c_k	0.1	–	Eddies diffusivity constant
c_ε	0.7	–	Turbulent dissipation constant
$K_{m_{\min}}$	310^{-5}	$m^2.s^{-1}$	Minimal value for moment diffusivity
$\overline{e}_{\min}$	210^{-6}	$m^2.s^{-2}$	Minimal value for TKE
$\overline{e}_{\min_0}$	10^{-4}	$m^2.s^{-2}$	Minimal value for TKE at surface
Pr_t	1	–	Prandtl number
K_{ratio}	1	–	Ratio between TKE and momentum diffusivities
c_τ	3.75	–	Constant to compute surface TKE from wind stress

Abbreviation: TKE, turbulent kinetic energy.

where $K_e = K_{\text{ratio}} K_m$ stands for the eddy diffusivity of TKE (with K_{ratio} the dimensionless ratio between TKE and momentum diffusivities), $b = g(\rho_0 - \rho) / \rho_0$ for buoyancy, g for gravity and ε for TKE dissipation. The latter is parameterized as follows:

$$\varepsilon = c_\varepsilon \frac{\overline{e}^{3/2}}{l_\varepsilon}, \tag{12}$$

where c_ε is a constant and l_ε is a characteristic dissipation length computed (together with l_k) from the TKE and the Brunt-Väisälä frequency $N^2 = -\partial b / \partial z$. At the sea surface, $\overline{e}$ is a function of wind stress $\boldsymbol{\tau} = (\tau_x, \tau_y)$ and has a minimum value of $\overline{e}_{\min_0}$ ($\overline{e}_{\min}$ elsewhere in the water column, with $\overline{e}_{\min} \ll \overline{e}_{\min_0}$). This writes:

$$\overline{e}_{(z=0)} = \max\left(\overline{e}_{\min_0}, c_\tau \frac{\sqrt{\tau_x^2 + \tau_y^2}}{\rho}\right) \tag{13}$$

where c_τ is a constant to implicitly consider non-modeled processes (e.g., waves, Langmuir cells) that may affect TKE creation near the ocean surface. More details on the turbulent closure scheme and parameter values are given in Gaspar et al. (1990).

The mesh grid includes 42 vertical levels of variable size (from 1 m near the sea surface to 300 m near the sea bottom). Atmospheric forcings (i.e., sensible and latent heat fluxes, short- and long-wave radiations, and wind stress) used in this study for the 1DV model are the same as those used by Hamon et al. (2016) in the 3D simulation for the Mediterranean basin. They correspond to the atmospheric forcings calculated at DYFAMED station (43°25′ North, 7°52′ East) in the Ligurian Sea, and are available every 3 h.

3.2. The Optimization Problem

We selected eight parameters that may be tuned in practical applications of the TKE model, listed in Table 3. Three of them (c_k, c_ε, c_τ) are empirical descriptions of lower scale processes that are not explicitly resolved by the model, two others (Pr_t, K_{ratio}) are ratios between the different diffusivities, and the three last parameters ($K_{m_{\min}}$, $\overline{e}_{\min}$, $\overline{e}_{\min_0}$) are minimal values for quantities associated with TKE. In the original settings of the model, these parameters are set to reference values based on common assumptions in the literature ($Pr_t = 1$, $K_{\text{ratio}} = 1$ meaning $K_m = K_e$), lab experiments that are simplistic in comparison to the complex dynamics of the ocean surface, or even—especially for minimal values—based on an empirical tuning so that the model reproduces "reasonably well" observed oceanographic data (Gaspar et al., 1990). Therefore, these reference values form a good starting point but they may be tuned in model applications to better fit data, without loss of interest for the underlying theory behind the model.

To test the optimization method, we conduct twin experiments. The first experiment is a 1-year model simulation (year 1980 with corresponding atmospheric forcings and solar irradiance) performed using parameter values from Table 3 as a reference (Figure 2). Model outputs are used to generate pseudo-data that are used in the second experiment to attempt to recover the reference parameter values. This classical procedure of twin experiments allows to assess the efficiency of the optimization method as we know a priori the optimal parameter values that should be found with the pseudo-data set.

Pseudo-data are generated by adding a white Gaussian noise to temperature (T), salinity (S), and horizontal current (u and v) predictions. The added noise has a zero mean and its standard deviation is set to mimic the typical values of measurement uncertainty in actual in-situ instruments: $0.01°C$ for temperature, 0.01 for salinity and $0.01 m.s^{-1}$ for each direction of horizontal current (Thomson & Emery, 2014). An example of resulting pseudo data is shown in Figures 3a and 3b. Pseudo data are constructed from a low-frequency sampling of the numerical solution provided by the model. Therefore, the pseudo-data set shall be regarded

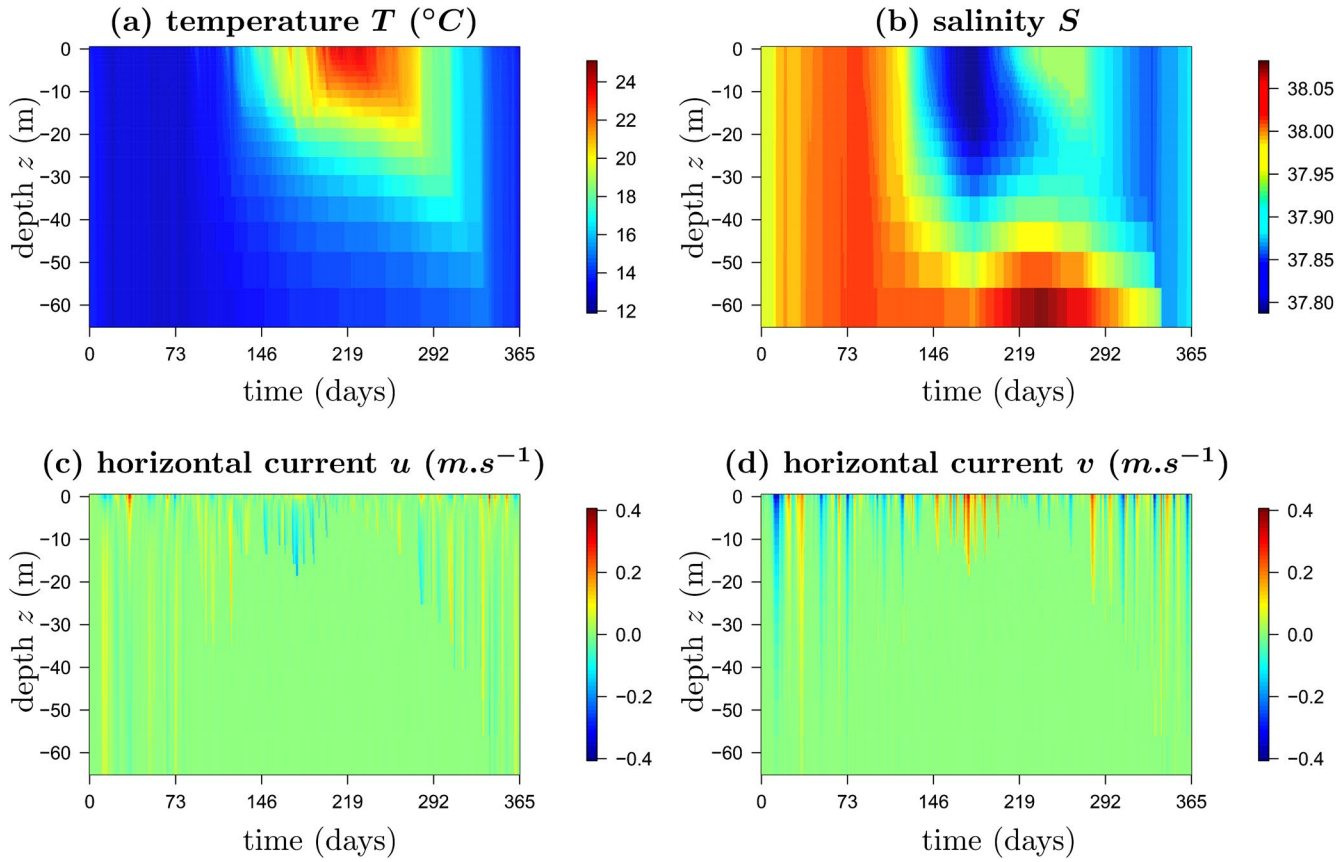

Figure 2. Reference simulation used as example in our study. For readability, only the upper water column is shown (deeper part not affected by atmospheric forcing).

as a data set collected every 3 h by in-situ instruments on a mooring line. This sampling frequency is far below the one of actual instruments (data every minute or few seconds), making our virtual setup more pessimistic than what could be realistically achieved in the field.

So, we have access to n observations (here pseudo-observations) of horizontal current, temperature and salinity at each time and depth predicted by model outputs. We denote u_i^{obs} the i-est observation of u (same notation for v, T and S). Each observation can be compared with model prediction $u_i^{\text{pred}}(\boldsymbol{\theta})$ at the same time and depth (Figures 3c and 3d), which depends on $\boldsymbol{\theta}$ the set of p model parameters that we intend to optimize. Parameters are expressed in different units and are of different orders of magnitude. To ensure a similar treatment for all parameters, we scale them by a scaling value, which can be a commonly accepted value from the literature or an order of magnitude. For our twin experiments, we use the reference parameter values (Table 3) as scaling values, resulting in the scaled parameter vector at the optimal parameter values being equal to $\mathbf{1}$. In addition, to avoid a distortion between scales and ease our graphical representations, we take $\boldsymbol{\theta}$ equals to the $\log_{10}$ of the scaled parameters. As a consequence, the optimal set of parameters is given by $\boldsymbol{\theta}^{\text{opt}} = \log_{10}(\mathbf{1}) = \mathbf{0}$, this choice having no effect on the results.

Model-data discrepancy is quantified by the cost function:

$$
\begin{aligned}
J(\boldsymbol{\theta}) \;= \alpha_{uv} &\left(\sum_{i=1}^{n} \left(u_i^{\text{pred}}(\boldsymbol{\theta}) - u_i^{\text{obs}} \right)^2 + \sum_{i=1}^{n} \left(v_i^{\text{pred}}(\boldsymbol{\theta}) - v_i^{\text{obs}} \right)^2 \right)^{1/2} \\
&+ \alpha_T \left(\sum_{i=1}^{n} \left(T_i^{\text{pred}}(\boldsymbol{\theta}) - T_i^{\text{obs}} \right)^2 \right)^{1/2} + \alpha_S \left(\sum_{i=1}^{n} \left(S_i^{\text{pred}}(\boldsymbol{\theta}) - S_i^{\text{obs}} \right)^2 \right)^{1/2},
\end{aligned}
\tag{14}
$$

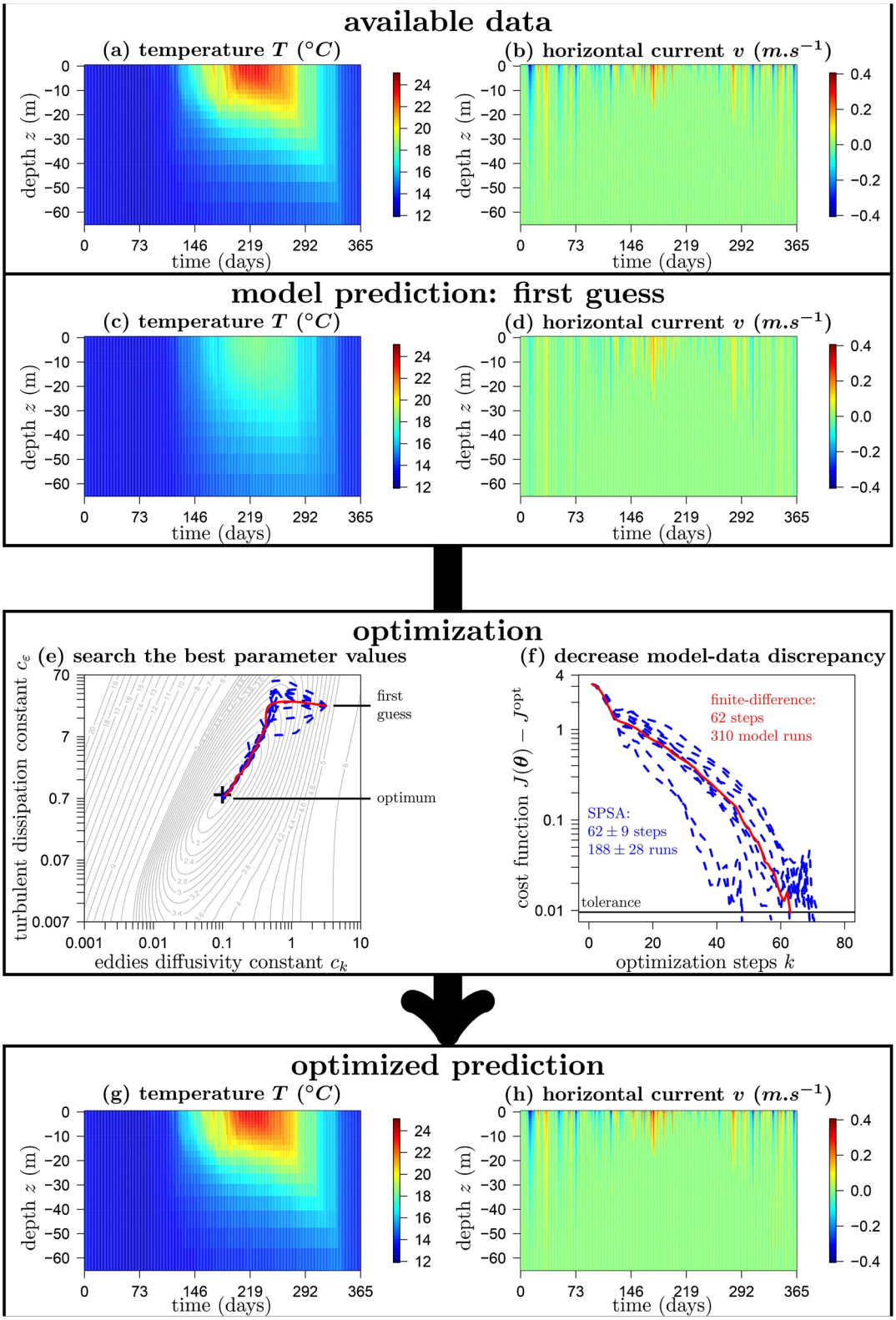
available data
(a) temperature T (°C)
(b) horizontal current v ($m.s^{-1}$)
depth z (m)
time (days)
model prediction: first guess
(c) temperature T (°C)
(d) horizontal current v ($m.s^{-1}$)
optimization
(e) search the best parameter values
turbulent dissipation constant c_ϵ
eddies diffusivity constant c_k
first guess
optimum
(f) decrease model-data discrepancy
cost function $J(\theta) - J^{opt}$
optimization steps k
finite-difference:
62 steps
310 model runs
SPSA:
62 ± 9 steps
188 ± 28 runs
tolerance
optimized prediction
(g) temperature T (°C)
(h) horizontal current v ($m.s^{-1}$)

Figure 3.

where $\alpha_{uv} = 100$ and $\alpha_T = \alpha_S = 1$ are chosen to have variations of similar magnitude for both u, v, S and T. Evaluating the cost function $J(\boldsymbol{\theta})$ requires to run the model once. If $\boldsymbol{\theta}$ contains few parameters, a map of the cost function as a function of parameter values can be computed, an example with two parameters is shown in Figure 3e. Note that the map of the cost function in Figure 3e is costly to estimate—here 10000 model runs—and is not required for optimization, we show it here just to ease the presentation of the method applied to the TKE model with two parameters to optimize.

Starting from the same parameter values, both finite difference and SPSA algorithms converge to the optimal parameter values (Figure 3f) that make the closest model predictions to observed data (Figures 3g and 3h). On average, the stochastic optimization performed by SPSA converges to the global minimum (J^{tol} has been set to 1.44, corresponding to a tolerance of 0.01, since in this case $J^{\text{opt}} := J(\boldsymbol{\theta}^{opt}) \approx 1.43$ is the minimum of the cost function, obtained with the reference parameter values) in the same number of steps as the deterministic finite difference, with the same tuning for both algorithms ($a = 0.01$, $c = 0.01$, and $\beta = 0.8$). It is worth noting here the role of Nesterov momentum, which allows the finite difference algorithm to move forward like a ball rolling down a hill rather than by small incremental steps, making it converging along a faster and smoother trajectory. The same effect is also observed with the SPSA, with Nesterov momentum also playing an important role in averaging the successive gradient approximations, thus stabilizing the algorithm.

Finally, the strength of SPSA appears when looking at the computational cost. Even with two parameters to optimize, the SPSA uses 3 model runs per optimization step (an additional run computes the cost function at the new parameter values) whereas the finite difference uses 5 model runs, which saves 40% of the computational cost. For p parameters, the relative gain of using SPSA is roughly p. With eight parameters to optimize in the next section, SPSA can be expected to be eight times faster to run.

4. Optimizing All Parameters Together at Low Numerical Cost

4.1. Strategy to Evaluate SPSA Efficiency

For this exercise of optimizing the eight parameters together, we doubled the previously used standard deviation of our added noise (now $0.02°C$ for temperature, 0.02 for salinity and $0.02 m.s^{-1}$ for currents), to include in a simplistic way the uncertainty in the physical process itself, like the non-homogeneity of currents inside the mesh cells of an Acoustic Doppler Current Profiler, or geophysical fluctuations that are not resolved by the model.

Based on preliminary tests, we adjusted the SPSA parameters ($a = 0.003$, $c = 0.01$, and $\beta = 0.8$) to ensure the algorithm convergence. To test the method ability to optimize our example model, we conducted 100 repetitions of the optimization procedure, each one starting from a random initial guess of the parameter values. All the initial parameter sets for each optimization procedure are taken as equally distant (Euclidean norm) to the optimum parameter set $\boldsymbol{\theta}^{\text{opt}} = \mathbf{0}$. Therefore, the initial parameter sets correspond to points in the parameter space that all lie on a sphere centered around $\boldsymbol{\theta}^{\text{opt}}$ and that are randomly drawn following a uniform probability distribution (Marsaglia, 1972, method 2). The sphere radius is set to $d_{\text{IC}} = 1$, meaning that the initial parameter values will be up to 10 times smaller or larger than the optimal value.

For each of the 100 repetitions of the optimization procedure, a new set of pseudo-data is generated by adding a different random white Gaussian noise (see Section 3.2) to ensure the genericity of our results obtained by twin experiments. Also, we did not impose any tolerance J^{tol} and specified only $k_{\max} = 1000$ as stopping criterion to ease comparisons between our 100 repetitions. Only the SPSA method is applied, as both preliminary tests and intuition based on results from Figure 1 indicated that the finite difference algorithm takes too much computational time to converge when applied to the geophysical model.

Figure 3. Sketch of the optimization problem. Data (a, b) (here pseudo-data to test the optimization method) and a model to attempt to describe data (c, d). Model-data discrepancy is minimized by an optimization procedure that tunes model parameters: evolution of parameters and map of the cost function (e), corresponding decrease of the cost function $J(\boldsymbol{\theta})$ relatively to its value at optimum J^{opt} (f). Here, one finite difference gradient descent (red) and 10 simultaneous perturbations stochastic approximation runs (blue) starting from the same first guess are shown. The obtained parameter values lead to model predictions that are closer to observed data (g, h).

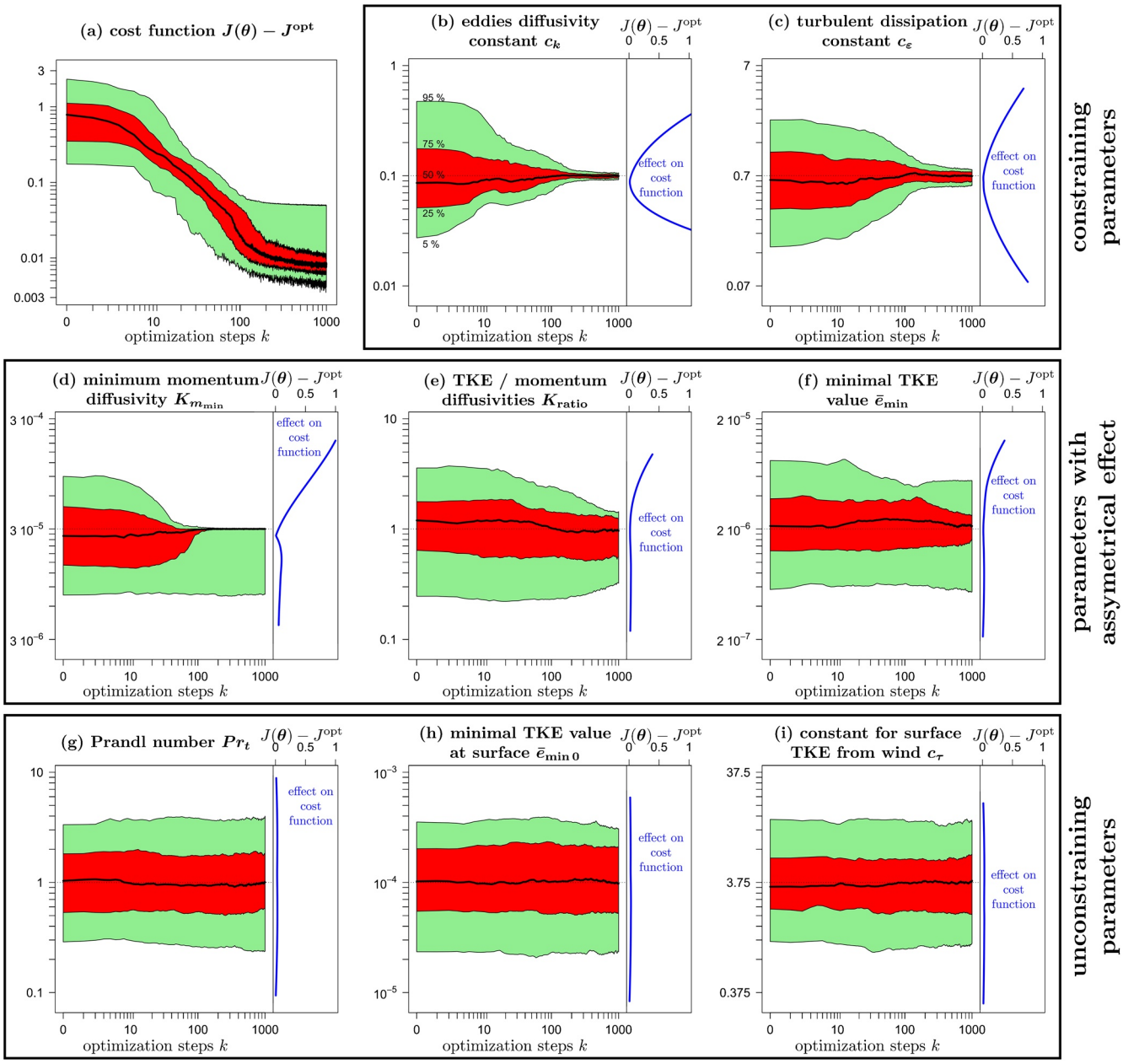

Figure 4. Statistics over 100 optimizations by simultaneous perturbations stochastic approximation of eight parameters of the Turbulent Kinetic Energy model: decrease of the cost function $J(\boldsymbol{\theta})$ relatively to its value at optimum J^{opt} (a), and evolution of the eight parameter values (b–i) classified in three qualitative evolutions. All axes are in log-scale. Results are summarized by quantile intervals (5% – 95% in green, 25% – 75% in red) and the median (solid curve), and the optimal value to find for the parameters (b–i) is indicated by the dotted line. (b)–(i) The right-hand-side inset with a blue curve shows the sensitivity of the cost function to the parameter, estimated by a non-parametric local regression predicting the cost function value as a function of the parameter value (span parameter set to 0.75), based on all the cost function evaluations performed during the 100 optimizations. A convex (flat) curve corresponds to a high (low) sensitivity of the cost function—that is, model predictions—to the parameter, and explains why the optimal parameter value can be estimated with a high (low) accuracy, independently of the optimization method we use.

4.2. Results

Results for 100 repetitions of the optimization of the eight parameters are summarized in Figure 4 and Table 4 (last column). First of all, the method decreases the cost function value close to the noise level $J^{opt} := J(\boldsymbol{\theta}^{opt})$, defined as the cost function at the known optimal parameter values and which is positive due to the noise in pseudo-data. A decrease of 1–2 orders of magnitude of $J(\boldsymbol{\theta})$ is obtained in a 100

Table 4
Optimal Values and Optimization Results for the Eight Parameters Simultaneously Optimized, Depending on the Noise Level in Pseudo-Data

	Optim.	Noise = 0.005		Noise = 0.01		Noise = 0.02	
Param.	Value	Average	Confidence	Average	Confidence	Average	Confidence
c_k	$1.00\,10^{-1}$	0.9910^{-1}	$[0.9410^{-1}, 1.0410^{-1}]$	0.9810^{-1}	$[0.9210^{-1}, 1.0510^{-1}]$	0.9910^{-1}	$[0.9210^{-1}, 1.0610^{-1}]$
c_ε	7.0010^{-1}	6.8910^{-1}	$[6.0910^{-1}, 7.7110^{-1}]$	6.7510^{-1}	$[5.8510^{-1}, 7.7210^{-1}]$	6.8710^{-1}	$[5.7010^{-1}, 7.9310^{-1}]$
$K_{m_{\min}}$	3.0010^{-5}	2.6910^{-5}	$[0.7810^{-5}, 3.0410^{-5}]$	2.7010^{-5}	$[0.8710^{-5}, 3.0410^{-5}]$	2.7010^{-5}	$[0.7810^{-5}, 3.0310^{-5}]$
$\overline{e}_{\min_0}$	$1.00\,10^{-4}$	1.0510^{-4}	$[0.1910^{-4}, 2.8710^{-4}]$	1.1810^{-4}	$[0.2410^{-4}, 2.7710^{-4}]$	1.3710^{-4}	$[0.2210^{-4}, 3.1410^{-4}]$
$\overline{e}_{\min}$	2.0010^{-6}	2.3010^{-6}	$[0.6010^{-6}, 5.2410^{-6}]$	2.4310^{-6}	$[0.5210^{-6}, 5.3010^{-6}]$	2.3710^{-6}	$[0.5710^{-6}, 5.4610^{-6}]$
Pr_t	1.00	1.81	[0.21, 5.15]	1.45	[0.20, 4.78]	1.42	[0.24, 3.82]
K_{ratio}	1.00	0.93	[0.48, 1.28]	0.85	[0.37, 1.27]	0.91	[0.31, 1.41]
c_τ	3.75	5.17	[0.77, 12.88]	5.12	[0.80, 15.19]	4.80	[0.99, 13.19]

Note. Results presented in Figure 4 are the ones obtained with the highest tested noise level (0.02). Meaning and unit for each parameter can be found in Table 3. Confidence interval are defined by the 5% and 95% quantiles of the results obtained in our 100 repetitions of the optimization procedure.

optimization steps, costing 300 model runs (Figure 4a). Pushing forward the optimization to 1,000 optimization steps leads to a small additional decrease of $J(\boldsymbol{\theta})$ lower than an order of magnitude, as the remaining potential gain after 100 optimization steps is small. In this part of the optimization, the effect of noise in pseudo-data combined with the stochastic component of SPSA appears in the "random-like" oscillations of cost function evolution. Last, the 95% quantile (over the 100 repetitions) of cost functions stays merely constant after a 100 optimization steps. This result is due to the 15% of our optimizations that ended up being stuck into a local minimum or a plateau, a phenomenon on which we will come back later.

Each parameter has a different effect on model predictions and therefore on the cost function value. The cost function will significantly increase if a parameter that governs model predictions deviates from its optimal value. Therefore, the accuracy in parameter estimation is related to the near-optimum curvature of the cost function in the direction defined by this parameter. Based on this curvature, the eight parameters in our example can be grouped into three qualitative types.

First, the two constants of the TKE model c_k and c_ε (Figures 4b and 4c) govern the model predictions considered in the cost function. Indeed, these parameters are recovered with a high accuracy, 90% of estimations falling in $\hat{c}_k \in [0.092, 0.106]$ and $\hat{c}_\varepsilon \in [0.58, 0.77]$ (5% and 95% quantiles). Based on the optimization trajectories, we can estimate the effect of each of these parameters on the cost function. This effect is shown in the right-hand-side inset in Figures 4b and 4c. It indicates that the cost function is strongly convex and rises rapidly in both directions for each parameter, making the corresponding parameter easy to optimize with high accuracy.

Second, some parameters have on the opposite a little effect on model predictions and are estimated with a large uncertainty, usually spanning one order of magnitude (Figures 4g–4i). Such parameters are the Prandtl number ($\widehat{Pr}_t \in [0.24, 3.82]$), the minimal value for TKE at the surface ($\hat{\overline{e}}_{\min_0} \in [0.22, 3.14] \times 10^{-4}\,\mathrm{m}^2.\mathrm{s}^{-2}$) and the constant linking TKE at the surface with wind stress ($\hat{c}_\tau \in [1.0, 13.2]$). For each parameter, the cost function looks flat in both directions, making the search of its minimum value a difficult task and explaining the low accuracy in parameter estimation.

Third, some parameters are a mix of the previous ones, they have little effect on model predictions in one direction and strongly affect model predictions in the other one (Figures 4d–4f). As a consequence, the confidence interval in the estimation of those parameters is asymmetrical, presenting a large uncertainty

toward lower values (half an order of magnitude). Two of these parameters are minimal thresholds, for the momentum diffusivity ($\hat{K}_{m_{\min}} \in [0.76, 3.03] \times 10^{-5}\,\mathrm{m}^2.\mathrm{s}^{-1}$) and TKE ($\hat{\overline{e}}_{\min} \in [0.57, 5.47] \times 10^{-6}\,\mathrm{m}^2.\mathrm{s}^{-2}$) respectively, the last parameter being the ratio between TKE and momentum diffusivities ($\hat{K}_{\mathrm{ratio}} \in [0.31, 1.41]$). For each parameter, the cost function looks flat on one side and steep on the other side. This makes the search of the lowest point a difficult (easy) task if the initial guess is on the flat (steep) side.

Now, we would like to elaborate a bit further on the minimum momentum diffusivity $\hat{K}_{m_{\min}}$. In this parameter direction, the cost function presents a small hump (Figure 4d) that traps some of the optimization trajectories starting at lower values (15% of our repetitions of the optimization procedure). These trajectories end near a local minimum (also associated with an overestimated value for $\overline{e}_{\min}$) where the cost function value is larger than its optimal value of about 0.1.

All those previous results, namely on how accurately each parameter value is estimated, do not seem to be affected by the level of noise in the pseudo-data (Table 4, Figures S1 and S2). Lower noise levels than the one we used previously lead to the same results in terms of model sensitivity and accuracy of parameter estimation. Somewhat counter-intuitively, the increase of noise level did not significantly increase the scatter in parameter estimation. This scatter in a parameter estimation seems mostly driven by the constraints imposed by the parameter on model predictions rather than by the level of noise in data.

As the level of noise in the pseudo-data has little effect on the optimization results, the level of noise has also little effect on the resulting predictive error and predictive uncertainty. The predictive error (relatively to the reference simulation) after optimization has a null median value (Figure 5). The predictive error varies between model simulations based on the 100 optimal parameter sets. The previously discussed parametric uncertainty results on a predictive uncertainty (defined here as the 95% confidence interval of the predictive error) that is, of the same magnitude as the noise introduced in pseudo data to mimic measurement errors in real data, except for the temperature. The temperature in the 0–10 m layer during summer is on average slightly underestimated by less than 1°C, with a 95% confidence interval spanning up to 6°C near the surface (Figures S3–S8). At this time of the year, the radiative warming is maximal which magnifies the uncertainty in the temperature diffusivity. One possible explanation is that it results from the strong uncertainty on the optimization of the Prantl number. Therefore, the resulting predictive uncertainty would be lower in any optimization attempt that will not optimize the Prantl number and set it to its commonly accepted value of 1, or narrow the authorized range within which the Prantl number can be optimized. A related explanation would also be that the weighting of the state variables used in the cost function in our particular example strongly favors the optimization of the current prediction in comparison to the temperature.

5. Discussion

The main point to be highlighted is that the level of difficulty in determining a given parameter is more likely dictated by the shape of the cost function rather than by the optimization method itself (see the previous section). Any optimization method is likely to present similar scatters in the estimation of parameters that poorly constrain model predictions. The strength of SPSA is its low numerical cost, allowing to optimize many parameters together even for models that are costly to run. For “small models” like the one used here, this low numerical cost allows Monte-Carlo repetitions, like repeating the optimization starting from different initial guesses of the parameter values. These repetitions allow not only (a) to have a confidence interval in our estimation of parameter values, but (b) to have greater odds of finding a global minimum instead of a (sub-optimal) local one, and (c) to conduct a global sensitivity analysis of the model to its parameters at the same time. This sensitivity analysis provides an overview of the predictive uncertainty that arises from the uncertainty in the value of each parameter. Therefore, we learn on which crucial parameters we should focus our measurement and calibration efforts, whereas those of lower importance can be set to a commonly accepted value in the literature.

The geophysical example we used here, though limited to the optimization of eight model parameters to ease the presentation of the method, focuses on its ability to deal with nonlinear embedded models. SPSA method is indeed suitable for embedded and coupled models since it does not require additional effort to take the embedding/coupling into account. Furthermore, the SPSA method can be used to optimize larger numbers of model parameters, but also boundary and initial conditions, and external forcings. Indeed, the

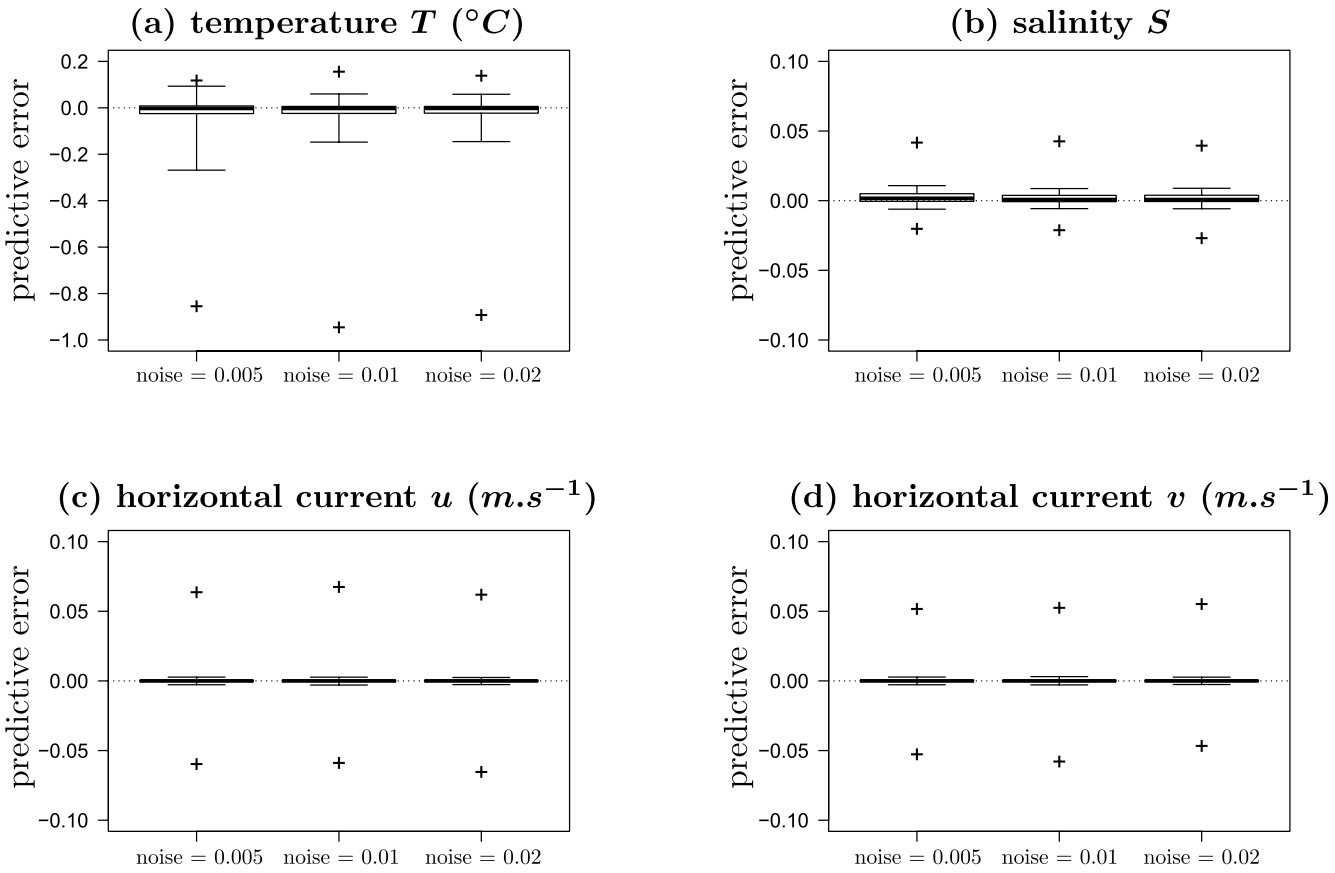

Figure 5. Statistics on the predictive uncertainty resulting from the 100 optimizations by Simultaneous Perturbations Stochastic Approximation of eight parameters of the Turbulent Kinetic Energy model. Each panel presents the predictive error for the model's state variables (T, S, u, v), computed from the optimizations performed for three different levels of noise in the pseudo-data (100 optimizations per noise level). Results from the optimizations are summarized in the form of boxplots computed from model predictions at each vertical cell and time step. Boxplots quantiles: median (horizontal bar), 25% and 75% (box), 2.5% and 97.5% (whiskers), minimum and maximum (points).

existing literature on SPSA in optimal control applications (Spall, 1998, 2012) and our pedagogical example minimizing a convex cost function (128 parameters) both show that the method can handle optimization problems with hundreds of parameters. In a related study involving the same variant of SPSA method, we optimized eight constants describing a mixture of biological parameters, initial conditions, and external forcing, in a nonlinear growth-advection model (coupling Lagrangian simulations with a biological model of ODEs) describing phytoplankton blooms triggered by island inputs (Messié et al., 2020). In another study, the same SPSA method successfully optimized the boundary conditions of a two-dimensional tidal model in a lagoon, discretized into 794 control parameters (tide amplitude and phase in 397 boundary cells) (Koenig et al., 2020). There, we also propose a combination of SPSA with successive spline approximations (resulting in 66 parameters to be optimized) to ensure the desired local continuity constraints for the boundaries. This approach can also be used for the common need of optimizing initial conditions. While initial conditions were assumed to be known in our example, they can therefore be estimated together with the parameter values through the SPSA method.

If the system exhibits deterministic chaos, like with weather prediction, the predicted system state needs to be regularly corrected with data to limit the exponentially growing uncertainty coming from the sensitivity to initial condition. This job is usually performed by Ensemble Kalman Filters or Ensemble Variational methods (3D-EnVar, 4D-EnVar), which include knowledge from data to refine the model prediction during the simulation (Buehner et al., 2013; Evensen, 2009). To perform a parameter identification with SPSA on a chaotic system, a possibility might be to use one of the above-mentioned methods each time a model simulation is run with a given set of parameters (e.g., one model run includes EnKF data assimilation to

counterbalance the sensitivity to initial condition). In that case, the cost function to minimize would be a weighted sum of the model-data discrepancy and of the amount of system state correction performed by the data assimilation method. Indeed, one main issue would be to assess the relative gains in predictive accuracy that can be achieved respectively thanks to parameter optimization and system state correction. Such an idea needs to be tested and its numerical cost to be evaluated.

The low cost of an SPSA optimization (in regards to the number of parameters to optimize) and its genericity (negligible implementation cost) form an interesting combination for the scientific exploratory process, as we experienced here and in Messié et al. (2020). Indeed, any change in the optimization problem to solve only requires small programming efforts as it does not require to develop an adjoint numerical code, and the reasonable computational cost allows to test many of such changes. In addition, if any new knowledge suggests to modify the model, for instance, by adding some new variables and/or processes—especially if the model has biological components—again only little effort is needed to adapt the interface between the model and the optimization method. The same reasoning holds if one wants to change the cost function to focus on improving different model outputs. Finally, the low computational cost of SPSA allows multiple attempts to tune or even improve the optimization method itself.

The optimization method we used here is only one of the multiple variants of SPSA that exist. These variants are based on the same general spirit of estimating a gradient by perturbating all the parameters together. Besides that, all refinements that may be imagined for gradient descent algorithms can be implemented. To bound the parameter space to explore to remain within an acceptable range, both strong and weak constraints can be implemented (Hartl et al., 1995). If one needs to speed up the convergence of the algorithm, there are multiple solutions. First, momentum effect can be used, in its Nesterov variant (like here) or in its classical version without anticipation (Spall & Christion, 1994; Sutskever et al., 2013). Despite its use has been the exemption rather than the rule in the SPSA literature, momentum is of considerable interest to stabilize the algorithm by averaging the successive gradient estimations, as emphasized throughout the present study. Second, there is a second-order SPSA that estimates the Hessian of the cost function, but there is a trade-off to find between the benefit of this new information and the uncertainty in the Hessian estimation with only 4 model runs (Zhu & Spall, 2002). Third, a line-search algorithm can adapt the step-size, but it increases the risk to converge only in a local minimum (Armijo, 1966). If one needs to optimize many parameters, it might help to work alternatively on—carefully defined–subsets of the whole parameter set using the cluster-SPSA (Tympakianaki et al., 2015). In the same spirit, one may favor some model parameters by using different SPSA coefficients for each model parameter, which is the reverse of our scaling that ensured the same treatment for all of them. Finally, if one deals with a cost function presenting many local minima, SPSA can be coupled with a Memetic Gradient Search to perform a global optimization (Li et al., 2008).

Finally, the application of SPSA to our example provides valuable solutions for optimal control of turbulent closure schemes (Dekeyser et al., 2004). First of all, our estimations are robust to the noise level, which only slightly increases the scatter in parameter estimation. This scatter seems mostly driven by the fact that some parameters have little effect on the cost function, whatever the level of noise, as these parameters poorly affect model predictions. Some of those parameters are involved in processes occurring near the sea surface, and have little effect on our cost function which encompasses the whole vertical domain. Using a cost function that focuses on near-surface data would allow to better estimate such parameters. Other parameters that are poorly estimated in an asymmetrical way are the minimal thresholds that only have to be *sufficiently small*. The last two parameters that poorly constrain model outputs are the Prandtl number and the ratio between moment and TKE diffusivities. This is an interesting result since, if the value of the Prandtl number is quite well-known at small scale, namely the scale of turbulence (i.e., a few centimeters), this is far less true at the scales explicitly resolved by circulation models (a few meters or tens of meters in the vertical direction). Hence, the 1DV model is not really sensitive to the value of the Prandtl number in that case. However, any refinement of these simple assumptions might be of little interest as our results indicate that they will have almost no effect on model predictions. Finally, the two constants c_k and c_ε of the TKE model can be estimated with a high accuracy. This is of special interest as these values are estimated from laboratory experiments and comparisons of model predictions with observations (Gaspar et al., 1990). Therefore, the proposed approach of optimization by SPSA provides an easy way to adjust these empirical constants to any oceanographic specific case, or even to find a reference value over a large range of observational in-situ data.

6. Conclusion

In this study, we presented the SPSA method coming from optimal control theory (Spall, 2012) to estimate the parameter values in complex embedded models in geophysical sciences. We illustrated the method by conducting twin experiments on the example of a turbulent closure scheme in a 1DV hydrodynamical model. As a result, parameters that mostly constrain model predictions are estimated with a high accuracy. On the opposite, some parameters are well estimated on average, but with poor confidence levels. Such parameters have little influence on the cost function which is flat regarding their variations—and this is independent from the minimization technique which is used—so they do not require calibration efforts and can be set to a commonly accepted value. Given the number of parameters to optimize simultaneously, all those results were obtained at a low numerical cost. This allows to optimize models that are costly to run without constructing an adjoint model. As the SPSA algorithm can be easily applied to any model, it allows the scientist to change the optimization problem to solve by changing either the model, and/or the cost function and/or the available data if required during the course of the scientific exploratory process. Finally, the method is fast enough to allow repetitions and to perform a sensitivity analysis. For all those reasons, we believe that SPSA method has a great potential for identifying parameters in the complex embedded models we use in geophysical sciences. As examples, recent applications are the optimization of (a) boundary conditions for tides modeling in a lagoon (Koenig et al., 2020), and (b) biological parameters, initial conditions, and external forcing in a growth-advection model describing phytoplankton blooms triggered by a delayed island effect in the Pacific Ocean (Messié et al., 2020).

Data Availability Statement

No empirical data were used. All the information regarding the 1DV model with TKE closure scheme and the simulation set-up are both (a) described in Gaspar et al. (1990) and Hamon et al. (2016) and (b) summarized in the present study. The computational effort required for all the simulation and optimization runs can be achieved on a desktop or laptop computer.

Acknowledgments
The authors acknowledge M. Messié, D. Nerini, and J. Fuda for discussions. C. Aldebert, P. Fraunié, and J.-Luc Devenon received funding from the French ANR and DGA under project Turbident (ANR16-ASTR-0019-01). The PhD scholarship of GK was funded by the French Ministry on Higher Education and Research. C. Aldebert, G. Koenig, M. Baklouti, and J. Devenon received funding from European FEDER Fund under project 1166-39417.

References

Arhonditsis, G., & Brett, M. (2004). Evaluation of the current state of mechanistic aquatic biogeochemical modeling. *Marine Ecology Progress Series*, *271*, 13–26. https://doi.org/10.3354/meps271013

Armijo, L. (1966). Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, *16*(1), 1–3. https://doi.org/10.2140/pjm.1966.16.1

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177–186). Springer. https://doi.org/10.1007/978-3-7908-2604-3_16

Bottou, L. (2012). Stochastic gradient descent tricks. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.). *Neural networks: Tricks of the trade* (pp. 421–436). Springer. https://doi.org/10.1007/978-3-642-35289-8_25

Boutet, M. (2015). *Estimation du frottement sur le fond pour la modélisation de la marée barotrope (estimation of bottom friction for barotropic tidal modelling French). (Unpublished doctoral dissertation)* (p. 175). Université de Bretagne Occidentale.

Buehner, M., Morneau, J., & Charette, C. (2013). Four-dimensional ensemble-variational data assimilation for global deterministic weather prediction. *Nonlinear Processes in Geophysics*, *20*(5). https://doi.org/10.5194/npg-20-669-2013

Dekeyser, Y., Leredde, Y., & Devenon, J. (2004). Marine-turbulence modeling using data assimilation. In H. Baumert, J. Simpson, & J. Sundermann (Eds.), *Marine turbulence—Theories, observations and models, results of the CARTUM project*. Cambridge University Press.

Evensen, G. (2009). The ensemble Kalman filter for combined state and parameter estimation. *IEEE Control Systems Magazine*, *29*(3), 83–104. https://doi.org/10.1109/mcs.2009.932223

Foley, A. (2010). Uncertainty in regional climate modelling: A review. *Progress in Physical Geography*, *34*(5), 647–670. https://doi.org/10.1177/0309133310375654

Gaspar, P., Grégoris, Y., & Lefèvre, J.-M. (1990). A simple eddy kinetic energy model for simulations of the oceanic vertical mixing: Tests at station papa and long-term upper ocean study site. *Journal of Geophysical Research: Oceans*, *95*(C9), 16179–16193. https://doi.org/10.1029/jc095ic09p16179

Hamon, M., Beuvier, J., Somot, S., Lellouche, J.-M., Greiner, E., Jordà, G., et al. (2016). Design and validation of MEDRYS, a Mediterranean Sea reanalysis over the period 1992–2013. *Ocean Science*, *12*(2), 577–599. https://doi.org/10.5194/os-12-577-2016

Hartl, R., Sethi, S., & Vickson, R. (1995). A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, *37*(2), 181–218. https://doi.org/10.1137/1037043

Heimbach, P., Hill, C., & Giering, R. (2005). An efficient exact adjoint of the parallel MIT general circulation model, generated via automatic differentiation. *Future Generation Computer Systems*, *21*, 1356–1371. https://doi.org/10.1016/j.future.2004.11.010

Hoang, H., & Baraille, R. (2011). On efficiency of simultaneous perturbation stochastic approximation method for implementation of an adaptive filter. *Computer Techniques and Applications*, *2*, 948–962.

Koenig, G., Aldebert, C., Chevalier, C., & Devenon, J.-L. (2020). Identifying lateral boundary conditions for the M2 tide in a coastal model using a stochastic gradient descent algorithm. *Ocean Modelling*, *156*, 101709. https://doi.org/10.1016/j.ocemod.2020.101709

Leredde, Y., Dekeyser, I., & Devenon, J. (2002). T-s data assimilation to optimise turbulent viscosity. An application to the berre lagoon hydrodynamics. *Journal of Coastal Research*, *18*(3), 555–567.

Leredde, Y., Devenon, J., & Dekeyser, I. (1999). Turbulent viscosity optimized by data assimilation. *Annales Geophysicae*, *17*(11), 1463–1477. https://doi.org/10.1007/s00585-999-1463-9

Lermusiaux, P., Chiu, C., Gawarkiewicz, G., Abbot, P., Robinson, A., Miller, R., & Lekien, F. (2006). *Quantifying uncertainties in ocean predictions*. Harvard University.

Li, B., Ong, Y.-S., Le, M., & Goh, C. (2008). Memetic gradient search. *Paper presented at the 2008 IEEE Congress on Evolutionary Computation*. IEEE World Congress on Computational Intelligence. https://doi.org/10.1109/CEC.2008.4631187

Lions, J. (1968). *Contrôle optimal de systèmes gouvernés par des opérateurs aux dérivées partielles (optimal control of systems governed by operators with partial derivatives in French)* (p. 426). Dunot.

Madec, G., Bourdallé-Badie, R., Bouttier, P.-A., Bricaud, C., Bruciaferri, D., Calvert, D., & Vancoppenolle, M. (2017). *Nemo ocean engine (version v3.6)*. Notes Du Pôle De Modélisation De L'institut Pierre-simon Laplace (IPSL). https://doi.org/10.5281/zenodo.1472492

Marsaglia, G. (1972). Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, *43*(2), 645–646. https://doi.org/10.1214/aoms/1177692644

McElreath, R. (2015). *Statistical rethinking: A Bayesian course with examples in R and Stan* (p. 483). CRC Press.

Messié, M., Petrenko, A., Doglioli, A., Aldebert, C., Martinez, E., Koenig, G., & Moutin, T. (2020). The delayed island mass effect: How islands can remotely trigger blooms in the oligotrophic ocean. *Geophysical Research Letters*, *47*(2), e2019GL085282. https://doi.org/10.1029/2019gl085282

Mitchell, M. (1998). *An introduction to genetic algorithm* (p. 221). MIT Press.

Moore, A. M., Arango, H. G., Di Lorenzo, E., Cornuelle, B. D., Miller, A. J., & Neilson, D. J. (2004). A comprehensive ocean prediction and analysis system based on the tangent linear and adjoint of a regional ocean model. *Ocean Modelling*, *7*(1–2), 227–258. https://doi.org/10.1016/j.ocemod.2003.11.001

Palmer, T., Shutts, G., Hagedorn, R., Doblas-Reyes, F., Jung, T., & Leutbecher, M. (2005). Representing model uncertainty in weather and climate prediction. *Annual Review of Earth and Planetary Sciences*, *33*, 163–193. https://doi.org/10.1146/annurev.earth.33.092203.122552

Peng, S.-Q., Xie, L., & Pietrafesa, L. J. (2007). Correcting the errors in the initial conditions and wind stress in storm surge simulation using an adjoint optimal technique. *Ocean Modelling*, *18*(3–4), 175–193. https://doi.org/10.1016/j.ocemod.2007.04.002

Plessix, R. (2006). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, *167*, 495–503. https://doi.org/10.1111/j.1365-246x.2006.02978.x

Spall, J. (1998). An overview of the simultaneous perturbation method for efficient optimization. *John Hopkins APL Technical Digest*, *19*(5), 482–492.

Spall, J. (2012). Stochastic optimization. In J. Gentle, W. Härdle, & Y. Mori (Eds.), *Handbook of computational statistics: Concepts and methods* (2nd ed., pp. 173–201). Springer-Verlag. https://doi.org/10.1007/978-3-642-21551-3_7

Spall, J., & Christion, J. (1994). Nonlinear adaptive control using neural networks: Estimation with a smoothed form of simultaneous gradient approximation. *Statistica Sinica*, *4*, 1–27.

Sun, L., Seidon, O., Nistor, I., & Liu, K. (2016). Review of the Kalman-type hydrological data assimilation. *Hydrological Sciences Journal*, *61*(13), 2348–2366. https://doi.org/10.1080/02626667.2015.1127376

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *Paper presented at the 30th International Conference on Machine Learning (ICML-13)* (Vol. 28, pp. 1139–1147).

Talagrand, O., & Courtier, P. (1987). Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quaterly Journal of the Royal Meteorological Society*, *113*(478), 1311–1328. https://doi.org/10.1002/qj.49711347812

Thomson, R., & Emery, W. (2014). *Data analysis methods in physical oceanography* (3rd ed., p. 716). Elsevier.

Tympakianaki, A., Koutsopoulos, H., & Jenelius, E. (2015). c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin-destination matrix estimation. *Transportation Research Part C*, *55*, 231–245. https://doi.org/10.1016/j.trc.2015.01.016

Zhigljavsky, A., & Zilinskas, A. (2008). *Stochastic global optimization* (p. 262). Springer.

Zhu, X., & Spall, J. (2002). A modified second-order SPSA optimization algorithm for finite samples. *International Journal of Adaptive Control and Signal Processing*, *16*, 397–409. https://doi.org/10.1002/acs.715