



HAL
open science

A Nonlinear Statistical Model of Turbulent air-sea Fluxes

Denis Bourras, Gilles Reverdin, Guy Caniaux, Sophie B elamari

► **To cite this version:**

Denis Bourras, Gilles Reverdin, Guy Caniaux, Sophie B elamari. A Nonlinear Statistical Model of Turbulent air-sea Fluxes. *Monthly Weather Review*, 2007, 135 (3), pp.1077-1089. 10.1175/MWR3335.1 . hal-00149474

HAL Id: hal-00149474

<https://hal.science/hal-00149474>

Submitted on 22 Dec 2020

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

L'archive ouverte pluridisciplinaire **HAL**, est destin ee au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche franais ou  trangers, des laboratoires publics ou priv es.

A Nonlinear Statistical Model of Turbulent Air–Sea Fluxes

DENIS BOURRAS

Centre d'Etude des Environnements Terrestre et Planétaires, Vélizy-Villacoublay, France

GILLES REVERDIN

Laboratoire d'Océanographie Dynamique et de Climatologie, Paris, France

GUY CANIAUX AND SOPHIE BELAMARI

Centre National de Recherches Météorologiques, Toulouse, France

(Manuscript received 23 November 2005, in final form 4 July 2006)

ABSTRACT

Most of the bulk algorithms used to calculate turbulent air–sea fluxes of momentum and heat are iterative algorithms whose convergence is slow and not always achieved. To avoid these drawbacks that are critical when large datasets must be processed, a statistical model of bulk air–sea fluxes based on artificial neural networks was developed. It was found that classical bulk algorithms were slower than the statistical model, by a factor of 1.75–7 depending on the bulk algorithm selected for the comparison. A set of 12 global analyses of an operational meteorological model as well as in situ data corresponding to equatorial and midlatitude conditions were used to assess the accuracy of the proposed model. The wind stress, latent, and sensible heat fluxes calculated with neural networks have acceptable biases with respect to bulk fluxes, between 0.4% and 1% depending on the flux magnitudes. Moreover, the rms deviation between bulk fluxes and neural network flux estimates is only 0.003 N m^{-2} for the momentum flux, 0.5 W m^{-2} for the sensible heat flux, and 1.8 W m^{-2} for the latent heat flux, at global scale, which is small compared with the natural variability of these quantities or the expected error.

1. Introduction

The turbulent fluxes of momentum τ , sensible heat H_S , and latent heat L_E at the air–sea interface characterize the exchanges of mechanical energy, temperature, and humidity between the sea and the atmosphere. The calculation of these fluxes is a key issue in climatology, meteorology, and oceanography because air–sea fluxes are a boundary condition for models of the atmosphere and the ocean.

In most applications, turbulent fluxes are derived from in situ meteorological observations, output fields of atmospheric circulation models, or even satellite sensor data. Except in a very few cases for which direct measurements of the fluxes are available, the only possible approach for computing fluxes is to use a bulk

algorithm (e.g., Liu et al. 1979; Fairall et al. 1996). The inputs of bulk algorithms are so-called bulk variables, which are ensemble averages of sea surface temperature (SST), air temperature θ_A at a known altitude z_A above the ocean surface, specific humidity q_A , wind speed u_A , and sea level pressure (SLP). Most of the existing bulk algorithms are iterative, which means that the algorithm stops and returns the values of the fluxes if convergence is achieved. If not, no flux value is returned. This is a major disadvantage because the algorithms may or may not converge toward a solution, depending on the values of the bulk variables used as inputs. Moreover, those algorithms converge slowly, which is a critical issue when large datasets must be processed. There is one exception to this: Fairall et al. (2003) recently produced a bulk algorithm that is several times faster than previous algorithms, and which does not have any convergence problem. This method will be analyzed and its performances compared to those of the method proposed hereafter.

This article presents an accurate and computationally

Corresponding author address: Denis Bourras, CETP-IPSL-CNRS 10-12, Avenue de l'Europe, 78140 Vélizy-Villacoublay, France.

E-mail: denis.bourras@cetp.ipsl.fr

fast method for calculating turbulent air–sea fluxes. The method is a nonlinear statistical model of a modified version of the bulk algorithm developed for the Coupled Ocean–Atmosphere Response Experiment (COARE) program, version 2.5 (the COARE2.5 algorithm; Fairall et al. 1996), which uses the Smith (1980) parameterization of the bulk coefficients. The bulk algorithm is called the modified COARE (MC) algorithm hereafter. The proposed method is based on artificial neural networks (ANNs; e.g., Bishop 1995). ANNs were never used before in this field of research. The proposed approach is particularly adapted to heavy computational tasks such as initializing ocean global circulation models, estimating long time series of the heat budget of the upper ocean, or computing fluxes in ocean–atmosphere coupled models. A useful feature of the method is that it produces fluxes regardless of the combination of bulk variables used as inputs. In contrast, flux fields calculated with the MC algorithm have gaps at the locations where the algorithms do not converge. For evaluating the accuracy of the proposed method, data collected during the Programme Océan Multidisciplinaire Méso Echelle (POMME) experiment (Mémery et al. 2005) were used together with observations of the Pilot Research Moored Array in the Tropical Atlantic (PIRATA) buoys (Servain et al. 1998) and analyses of the European Centre for Medium-Range Weather Forecasts (ECMWF) model. These datasets represent a total of ~250 000 kinds of environmental conditions, which should ensure that the results presented in this article cover most of the possible situations to which the method could be applied.

The MC and ANN algorithms are presented in the next two sections. The datasets used are described in section 4. In section 5, the accuracy of the fluxes calculated with ANNs is analyzed with observations of the POMME and PIRATA datasets. Section 6 presents an evaluation of the accuracy of the method at global scale with ECMWF analyses. It also includes a spatial analysis of the accuracy of the method, with an emphasis on the cases for which the MC algorithm did not converge, followed by the conclusions.

2. MC algorithm and COARE3.0

The MC algorithm is a bulk algorithm, which is a calculation method associated with the bulk parameterization of the turbulent fluxes. The bulk parameterization is based on the Monin and Obukhov (1954) theory of the surface atmospheric boundary layer (SABL; Businger et al. 1971), in which τ , H_S , and L_E are related to the differences between u_A , θ_A , and q_A , and their

values at the sea surface, u_S , SST, and q_S , respectively [Eqs. (1)–(3)]:

$$\tau = \rho C_D (u_A - u_S)^2, \quad (1)$$

$$H_S = \rho C_p C_H (u_A - u_S) (SST - \theta_A), \quad \text{and} \quad (2)$$

$$L_E = \rho L_v C_E (u_A - u_S) (q_S - q_A). \quad (3)$$

In Eqs. (1)–(3), ρ is the density of air, C_p is its specific heat, and L_v is the latent heat of vaporization of water. The τ is positive downward when the atmosphere transmits momentum to the ocean, and H_S and L_E are positive upward when the ocean is cooling. Here C_D , C_H , and C_E are the exchange coefficients for momentum, sensible, and latent heat, respectively, which are usually parameterized as a function of the stability of the SABL, and wind speed. Stability is represented by the ratio $\zeta = z_A/L$, where L is the Monin–Obukhov length. The parameterization of the exchange coefficients used throughout this article follows Smith (1980). It was preferred to the Fairall et al. (1996) parameterization, which was originally designed for tropical regions and may not be adapted to some midlatitude conditions. The MC algorithm is iterative because the relationship between bulk variables and fluxes depends on ζ , which is itself a function of the fluxes. Technically, the MC algorithm consists of two parts, a first calculation of the fluxes for which ζ is assumed to be zero (neutral SABL), and an iterative sequence that includes the calculation of ζ and the fluxes. In most bulk algorithms, the sequence is interrupted when the differences in ζ and fluxes between two successive iterations are smaller than prescribed thresholds, which are defined according to the desired accuracy of the flux calculation. If the convergence is not achieved in a limited number of iterations (20–50 depending on the authors), the algorithm returns no flux value. The MC algorithm was arbitrarily selected for the present study. However, the proposed method may also be applied to other bulk algorithms (Brunke et al. 2003).

A bulk algorithm named COARE3.0 was recently released by Fairall et al. (2003). According to these authors, the COARE3.0 algorithm is 7 times faster than COARE2.5, and does not have any convergence problem because its main loop performs only 3 iterations instead of 20–50 for more classical algorithms. According to Fairall et al. (2003), three iterations are sufficient because they use an improved first guess for stability (Grachev and Fairall 1997). To check this, we performed the following test. One million fluxes were calculated with the COARE3.0 algorithm, with inputs randomly selected within realistic limits (in the same fashion as the dataset described in section 3a). Next, the

same input data were used for calculating new series of fluxes with modified versions of the COARE3.0 algorithm, in which the number of iterations (initially 3) was increased to an arbitrary number, ranging from 4 to 20. After several attempts, we found that the discrepancy between the fluxes calculated with three and five iterations was not negligible, namely 4 W m^{-2} in rms for the latent and 4.5 W m^{-2} for the sensible heat flux, which converts to 5% of error for a 100 W m^{-2} flux. This result shows that convergence is not always obtained with three iterations only, in spite of the improved guess for ζ . Eventually, we found that 5–10 iterations were necessary to obtain stable flux values (discrepancy smaller than 1 W m^{-2} for the sensible and latent heat fluxes) with the COARE3.0 algorithm, which affects the speed of execution of COARE3.0 by a factor of 1.3–2. In addition, several threshold values are used for the roughness lengths and z/L in COARE3.0. Even if the threshold values are reached, the algorithm continues until it returns a flux value, which can lead to unrealistic flux values. As the COARE3.0 algorithm is faster than other bulk algorithms, its speed of execution will be used as a point of reference in efficiency of algorithm convergence.

3. ANNs

The method proposed is a statistical model of the MC algorithm. It is a straightforward mathematical relationship between bulk variables and turbulent fluxes. The mathematical relationship has coefficients called weights and biases that are adjusted, so that given a set of bulk variables, the outputs of the network are as close as possible to the corresponding outputs of the MC algorithm. The process by which the coefficients of the network are adjusted is called learning. Hereafter, the concepts of neuron, ANN, and learning are described in more detail.

a. Definition

In an analogy to the brain, an entity made up of interconnected neurons, ANNs are made up of interconnected processing elements called artificial neurons, which respond in parallel to a set of input signals given to each. A biological neuron may have as many as 10 000 different inputs, and may send its output (the presence or absence of a short-duration spike) to many other neurons. In an ANN, a neuron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function. Mathematically, this can be written as

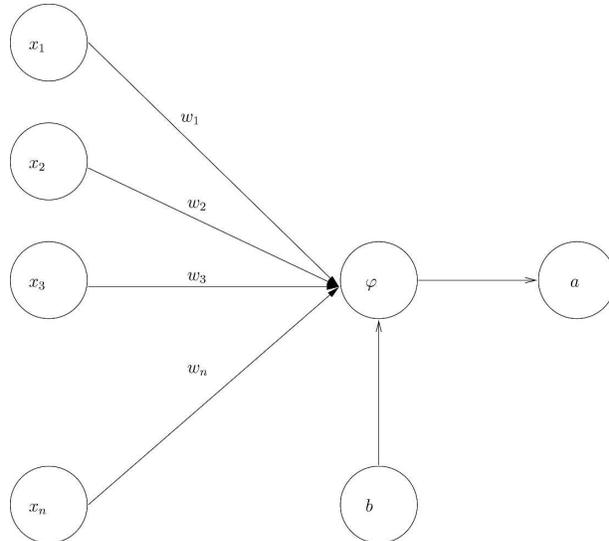


FIG. 1. Representation of an artificial neuron. It computes a single output a . Its inputs x_i are linearly combined (with coefficients w_i and b) and put through a transfer function φ .

$$a = \varphi \left(\sum_{i=1}^n w_i x_i + b \right) = \varphi(\mathbf{w}^T \mathbf{x} + b), \quad (4)$$

where a is the neuron output (i.e., the so-called activation). Here \mathbf{w} denotes the vector of weights, \mathbf{x} is the vector of inputs, b is the bias, and φ is the activation function. A graph of this operation is shown in Fig. 1. The activation function is often chosen to be a log-sigmoid [Eq. (5)] or a linear function [Eq. (6)]:

$$\varphi(x) = \frac{1}{1 + \exp(-x)}, \quad (5)$$

$$\varphi(x) = x. \quad (6)$$

The neurons are organized into layers, as represented in Fig. 2. Mathematically it reads as follows:

$$in_j = a_j^0 \xrightarrow{w_{jk}^1, b_k^1} a_k^1 \xrightarrow{w_k^2, \dots, b^2} \dots \xrightarrow{w^L, \dots, b^L} a_l^L = out_l^{ANN}, \quad (7)$$

where $j \in [1, m_0]$, $k \in [1, m_1]$, and $l \in [1, m_L]$. There are $L + 1$ layers of neurons, and L layers of weights w and biases b . Here m_0, m_1, \dots, m_L are the number of neurons in layers $0, 1, \dots, L$, respectively; in_j are the inputs and out_l^{ANN} are the outputs of the network; and a_k^i is the output of neuron k in layer i . The interconnections within the ANN are such that every neuron in each layer is connected to every neuron in the adjacent layers. For the present study, a typical configuration for an ANN would include the same inputs ($u_A, \theta_A, q_A, SLP,$

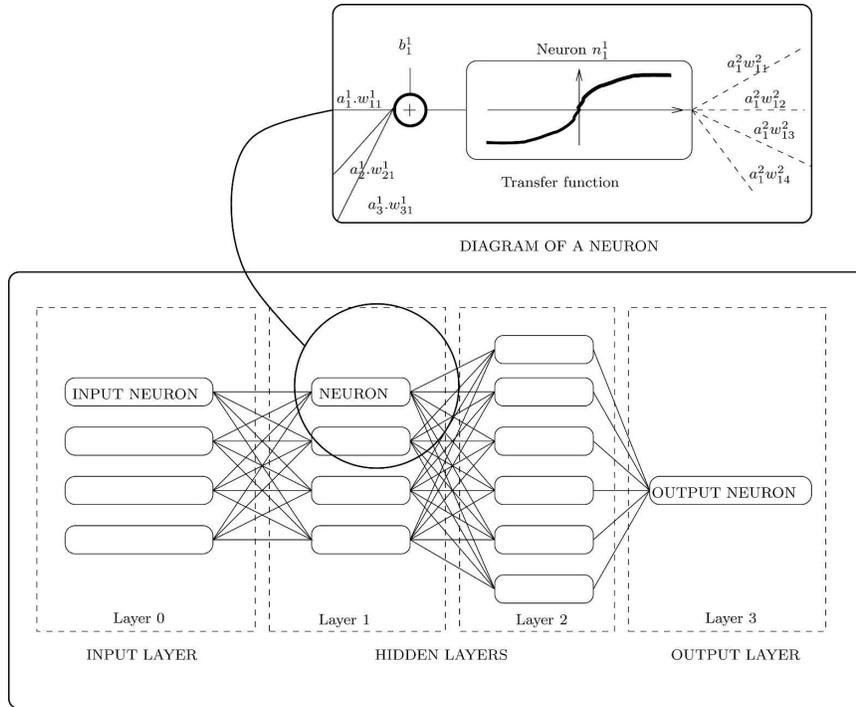


FIG. 2. Description of an ANN. The inputs of each neuron n are the output signals a of the neurons of the previous layer, multiplied by a weight w , and added to a bias b . In the present study, the inputs of the ANN are bulk variables, while the output is either τ , H_S , or L_E .

SST, and z_A) and outputs (τ , H_S , and L_E) as in the MC algorithm. The intermediate layers are called “hidden” layers. With one or two hidden layers, ANNs can approximate virtually any input–output map, provided that the hidden layer contains enough neurons.

b. Learning

Learning is the process by which a neuron is capable of changing its input–output behavior as a result of changes in the environment. During the learning stage of an ANN, weights and biases are modified in response to the inputs (in^{learn}) and outputs ($\text{out}^{\text{learn}}$) of a learning dataset. The learning dataset can be mentally represented as an array of n lines and $m_0 + m_L$ columns, where n is the length of the learning dataset. Line k contains the following elements, $\{\text{in}_{1k}^{\text{learn}}, \dots, \text{in}_{m_0k}^{\text{learn}}, \text{out}_{1k}^{\text{learn}}, \dots, \text{out}_{m_Lk}^{\text{learn}}\}$. Practically, for the present study it could be $\{u_{A_k}^{\text{learn}}, \theta_{A_k}^{\text{learn}}, q_{A_k}^{\text{learn}}, \text{SLP}_k^{\text{learn}}, z_{A_k}^{\text{learn}}, \tau_k^{\text{learn}}, H_{S_k}^{\text{learn}}, L_{E_k}^{\text{learn}}\}$, where $\{u_{A_k}^{\text{learn}}, \theta_{A_k}^{\text{learn}}, q_{A_k}^{\text{learn}}, \text{SLP}_k^{\text{learn}}, z_{A_k}^{\text{learn}}\}$ could be randomly selected, and the corresponding values $\{\tau_k^{\text{learn}}, H_{S_k}^{\text{learn}}, L_{E_k}^{\text{learn}}\}$ would be calculated with the MC algorithm.

Many learning algorithms are available in the literature. The most popular is the back-propagation method, which was used in the present study (Mehrotra

et al. 1996). It is an iterative distributed gradient descent technique that changes the weights and biases so that the actual outputs $\text{out}_{lk}^{\text{ANN}}$ become closer to the desired output $\text{out}_{lk}^{\text{learn}}$. A measure of this is the summed square error (SSE), defined as the quadratic sum of the differences between $\text{out}_{lk}^{\text{learn}}$ and $\text{out}_{lk}^{\text{ANN}}$, for all k .

The learning must be stopped at the right time. If it continues for too long, it results in overlearning. Overlearning means that the ANN extracts too much information from the individual cases of the learning dataset, forgetting the relevant information of the general case. Overlearning can be identified by checking the accuracy of the ANN outputs not only with the learning dataset but also with additional independent datasets that are so-called testing datasets. At each iteration of the learning process, the SSE should decrease for the testing and the learning datasets. But if overlearning happens, the SSE will continue to decrease for the learning dataset, whereas it will start to increase for the testing dataset.

c. Optimization of the ANN

The performances of the ANNs strongly depend on the number of layers and neurons, and the choice of the

TABLE 1. Characteristics of the ANNs.

Name	Variable calculated	Application conditions	Neurons of layers 1, 2, 3, 4	Input variables
ANN1	τ (N m^{-2})	$\tau \in [0, 0.5]$	8, 14, 7, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}$
ANN2	τ (N m^{-2})	$\tau > 0.5$	8, 14, 7, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}$
ANN3	H_S (W m^{-2})	$\text{SST} \leq \theta_A$	9, 14, 7, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}, u_A(\text{SST} - \theta_A)$
ANN4	H_S (W m^{-2})	$\text{SST} > \theta_A$	9, 14, 7, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}, u_A(\text{SST} - \theta_A)$
ANN5	L_E (W m^{-2})	$q_S < q_A$	9, 14, 7, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}, u_A(q_S - q_A)$
ANN6	L_E (W m^{-2})	$q_S > q_A$ $L_E \in [0, 150]$	9, 13, 6, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}, u_A(q_S - q_A)$
ANN7	L_E (W m^{-2})	$q_S > q_A$ $L_E > 150$	9, 14, 7, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}, u_A(q_S - q_A)$
ANN8	L_E (W m^{-2})	$q_S > q_A$ $L_E > 150$ $\text{SST} > \theta_A$	10, 13, 6, 1	$z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)], \theta_A u_A^2, u_A, q_S - q_A, \text{SST} - \theta_A, \theta_A, q_A, z_A, \text{SLP}, u_A(q_S - q_A)$

transfer functions. In practice, the optimization of the ANNs is done by modifying the configuration of the ANNs as long as the accuracy of the ANN fluxes is not satisfactory for the learning and the testing datasets. Note that the choice of the configuration of the ANNs is always a matter of trial and error.

Although it is theoretically possible to create a single ANN that would give τ , H_S , and L_E at the same time, several ANNs were used because of the complexity of the relationships between bulk variables and fluxes. Two ANN were dedicated to the calculation of τ , two others were used for H_S (according to the sign of $\text{SST} - \theta_A$), and four ANNs were necessary for L_E . One ANN was designed for the calculation of negative values of L_E (negative $q_S - q_A$), and another for the positive L_E smaller than $+150 \text{ W m}^{-2}$. Beyond $+150 \text{ W m}^{-2}$, two ANNs were used for the calculation of L_E . These two ANNs correspond to positive and negative values of $\text{SST} - \theta_A$, respectively. Note that the combination of a negative H_S value and a value of L_E larger than 150 W m^{-2} is not common over the World Ocean, but it is not unphysical. The eight ANNs, which are called ANN1–ANN8 in the following, are independent from each other. They use 8–10 input variables and have 4 layers, as reported in Table 1. The transfer functions are log-sigmoid for all the neurons, except for those of the output layers, which are linear. The input variables of the ANNs include not only the bulk variables, but also combinations of them that were helpful to increase the accuracy of the calculated fluxes (Table 1). In particular, the reader may notice the presence of the following quantities $z_A[(\theta_A - \text{SST}) + 0.608\theta_A(q_A - q_S)]$ and $\theta_A u_A^2$, in Table 1. They correspond to the numerator and the denominator of the Richardson number, which may be considered as an estimator of ζ .

4. Datasets

Five datasets are presented in this section; namely, four datasets that were used to create the ANNs (one learning dataset and three testing datasets) and one validation dataset intended to assess the accuracy of the ANN fluxes for global-scale applications. Note that the term dataset is used for simplicity and does not only mean “set of observations.” For instance, the validation dataset consists of model analyses, not data. Each of the datasets described hereafter is an ensemble of values of bulk variables (and/or combinations of them) used as inputs of the ANNs, and the corresponding fluxes calculated with the MC algorithm (MC fluxes).

a. Learning dataset

The learning dataset consists of 60 000 uniformly distributed values of bulk variables and z_A that were randomly selected between the minima and maxima reported in Table 2 (rows 1–6). The extrema of u_A , θ_A , q_A , SLP, and SST were chosen according to ECMWF analyses. The range of z_A corresponds to most of the possible range of application of the model, from 3.5 to 35 m. As they were selected in a random manner, many combinations of the bulk variables were unrealistic. To avoid at least part of these combinations, the values of the bulk variables that corresponded to MC fluxes and values of ζ outside the extrema specified in Table 2, lines 7–10 were rejected.

b. Testing datasets

The first testing dataset is identical to the learning dataset, except that its size is larger (100 000 situations). This dataset—named the “all-conditions dataset” in the following—covers every possible type of environmental

TABLE 2. Minimum and maximum values selected for the bulk variables and fluxes of the learning datasets.

Variable	Units	Min value	Max value
u_A	m s^{-1}	0.1	27
θ_A	$^{\circ}\text{C}$	-20	32
SST	$^{\circ}\text{C}$	0.1	36
q_A	g kg^{-1}	0.1	24
SLP	hPa	900	1040
z_A	m	3.5	35
τ	N m^{-2}	0	1.5
H_S	W m^{-2}	-150	600
L_E	W m^{-2}	-100	800
ζ	—	-50	0.25

conditions within the limits reported in Table 2. To specifically assess the accuracy of the ANNs in midlatitude and equatorial regions, two other testing datasets based on in situ observations were created. Data collected during the POMME experiment were gathered for the first dataset. POMME was conducted in the North Atlantic, from February to April 2001, in a midlatitude region from 35°N , 38°W to 49°N , 33°W . We used data that gather thermosalinograph SST measurements and meteorological observations made at 17.5 m above the sea surface, which were performed onboard the R/V *Atalante*, a French research vessel (Caniaux et al. 2005). During POMME, various stability conditions were encountered, from very unstable ($\zeta = -47$) to stable ($\zeta = 0.24$), with winds from 0 to 20 m s^{-1} and moderate humidity ($4\text{--}10 \text{ g kg}^{-1}$). In contrast, the second testing dataset represents equatorial conditions, with free-convection episodes, low winds (5.9 m s^{-1} on average), and large humidity (19.6 g kg^{-1} for the whole dataset). Data are 10-min-averaged observations from 22 January 1998 to 2 May 2001, transmitted by one moored buoy of the PIRATA network, located at 0° , 35°W . The height of measurement of the meteorological variables was 3.5 m. Note that the PIRATA buoys did not measure SLP, which was thus assumed to be constant (1013.15 hPa) for this study.

c. Validation dataset

The validation dataset consists of 12 ECMWF operational analyses whose dates and hours were selected between 1997 and 1998. Global analyses were chosen because they correspond to the intended context of application of the ANNs, which is initializing numerical models of the global ocean. The choice of 12 fields is a compromise between a good representation of all the possible environmental conditions at global scale and the maximum number of data that could reasonably be processed. Two useful features of this dataset are its large size (268 227 flux values) and its spatial consistency,

which means that it is possible to analyze the performance of the ANNs as a function of latitude and longitude. The spatial resolution of the analyses is $1.125^{\circ} \times 1.125^{\circ}$. Bulk variables were extracted at the lowest level of the ECMWF model, and fluxes were calculated with the MC algorithm.

5. Learning of the ANNs

In this section, we present the results of the learning phase of the eight ANNs described in section 3, that is the comparison between ANN fluxes and MC fluxes for the learning and testing datasets. In the following, the estimates of τ , H_S , and L_E obtained after application of the ANNs to bulk variables are called τ_{ANN} , H_{SANN} , and L_{EANN} , respectively. Similarly, the values of τ , H_S , and L_E calculated with the MC algorithm are called τ_{MC} , H_{SMC} , and L_{EMC} , respectively. The rms deviations and biases between ANN fluxes and MC fluxes are given in flux units (N m^{-2} and W m^{-2}) as well as in percentage, with respect to the following reference fluxes: 0.1 N m^{-2} for the momentum flux, 30 W m^{-2} for the sensible heat flux, and 70 W m^{-2} for the latent heat flux. These values correspond to an average over the World Oceans, calculated with the 12 analyses of the validation dataset.

The comparisons between the ANN fluxes of the learning and testing datasets and the corresponding MC fluxes are reported in Table 3 and suggest a very good performance. Specifically, the rms deviation between τ_{ANN} and τ_{MC} for the learning dataset is 0.003 N m^{-2} or 3%. The rms deviation between τ_{ANN} and τ_{MC} is even smaller for the all-conditions dataset and POMME (0.002 N m^{-2}), and is negligible for PIRATA (0.001 N m^{-2}). One may notice a positive bias between τ_{ANN} and τ_{MC} for this dataset ($+0.004 \text{ N m}^{-2}$, or 4%), and a small negative bias for POMME (-0.001 N m^{-2}). It was found that the bias slightly depended on the sign of ζ , revealing a nonperfect learning.

The rms deviations between H_{SANN} and H_{SMC} vary between 0.33 and 1.1 W m^{-2} , or 1.1% and 3.7%, depending on the dataset. The performance of the ANNs is the poorest for PIRATA in terms of bias (1.1 W m^{-2}), whereas it is negligible for POMME (0.2 W m^{-2}) and the all-conditions dataset (-0.1 W m^{-2}).

The comparison between L_{EANN} and L_{EMC} indicates that the rms error of the ANN estimates is 1.6 W m^{-2} for the learning dataset (2.3%), which is adequate. The rms is also small for POMME and PIRATA (i.e., $0.6\text{--}1.1 \text{ W m}^{-2}$). One may notice that the rms deviation is larger for the all-conditions dataset (2.1 W m^{-2} , or 3%) than for the learning dataset. This means that parts of the situations of the learning dataset were not correctly

TABLE 3. Comparison between T_{ANN} , H_{SANN} , L_{EANN} , and T_{MC} , H_{SMC} , L_{EMC} for the learning dataset, the all-conditions dataset, the POMME, and the PIRATA datasets.

Variables compared (units)	Statistics	Learning dataset	All-conditions dataset	POMME	PIRATA
T_{ANN} , T_{MC} ($N\ m^{-2}$)	Correlation	1	1	1	0.999
	Rms diff	0.003	0.002	0.002	0.001
	Bias	0	0	-0.001	0.004
	Linear fit	x	x	$0.996x$	$0.984x + 0.004$
H_{SANN} , H_{SMC} ($W\ m^{-2}$)	Correlation	1	1	1	0.992
	Rms diff	1.031	1.106	0.33	0.53
	Bias	0	-0.05	0.232	1.056
	Linear fit	x	$x - 0.04$	$0.985x + 0.368$	$1.002x + 1.05$
L_{EANN} , L_{EMC} ($W\ m^{-2}$)	Correlation	1	1	1	0.999
	Rms diff	1.638	2.137	1.082	0.606
	Bias	0	0.038	-0.144	0.007
	Linear fit	$x - 0.007$	$x - 0.045$	$0.985x + 0.886$	$1.02x - 0.015$

accounted for during the learning phase, which could possibly be solved by using a larger learning dataset. The bias of L_{EANN} with respect to L_{EMC} is negligible for all the testing datasets (0%–0.2%).

An extended version of the all-conditions dataset whose size was 1 000 000 000 fluxes was used for comparing the computational speed of the MC and ANN algorithms. The algorithms were implemented in

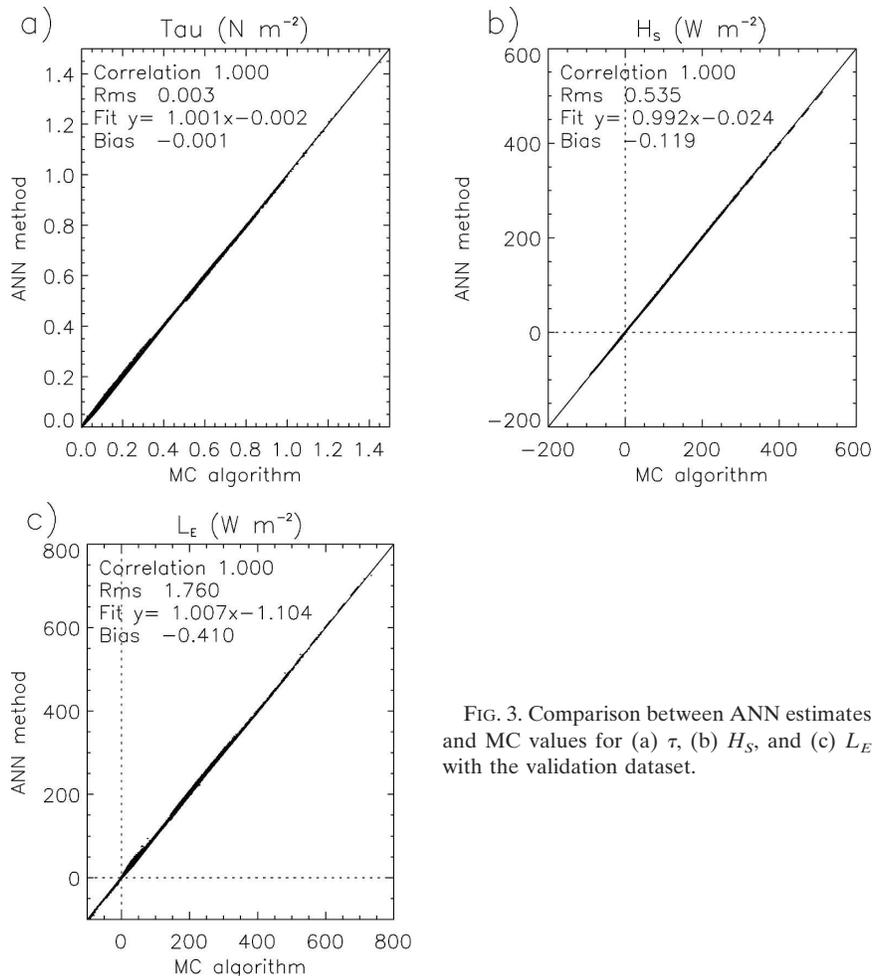


FIG. 3. Comparison between ANN estimates and MC values for (a) τ , (b) H_s , and (c) L_E with the validation dataset.

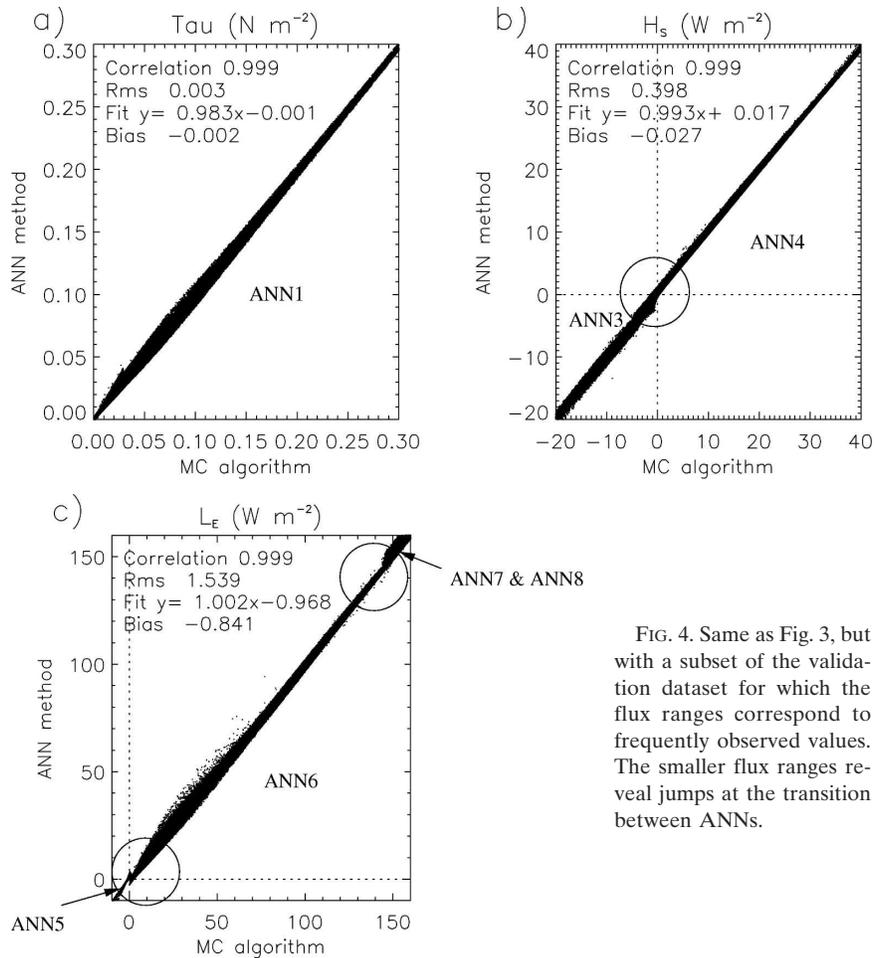


FIG. 4. Same as Fig. 3, but with a subset of the validation dataset for which the flux ranges correspond to frequently observed values. The smaller flux ranges reveal jumps at the transition between ANNs.

FORTTRAN, on a workstation with a 2-GHz processor. The computation time found was 1 min 12 s for the MC algorithm against 9.6 s for its ANN version, which means that the ANN fluxes were calculated more than 7 times faster than the MC fluxes. With the same data as input, it took 16.8 s for the COARE3.0 (three-iteration version) to compute the fluxes. This means that the COARE3.0 algorithm is very fast compared with the MC algorithm. However, the ANN is still faster than the COARE3.0 algorithm by a factor of 1.75 ($\sim 43\%$). Furthermore, the execution time increased to 21.6 s when the COARE3.0 algorithm was used with five iterations (which corresponds to $+55\%$ speed gain for the ANN, or a factor of 2.2), and 33.7 s with 10 iterations (71%, or $\times 3.5$).

6. Accuracy of the method for large-scale applications

In this section, the ANN flux estimates are compared with the MC fluxes of the validation dataset. Next, the

deviation between MC and ANN fluxes is analyzed in terms of spatial variations. Last, the accuracy of the ANN flux estimates is assessed for the cases where the MC algorithm did not converge.

a. Comparison between ANN and MC fluxes

The accuracy of the ANN fluxes is noticeably close for both the learning and the validation dataset (Table 3; Fig. 3). It confirms that the learning of the ANNs was overall correct. The rms error between τ_{ANN} and τ_{MC} is 0.003 N m^{-2} for the validation dataset, against 0.003 N m^{-2} for the learning dataset. For H_s , the rms deviation is even smaller for the validation dataset (0.5 W m^{-2}) than for the learning dataset (1.0 W m^{-2}). The rms deviation between $L_{E\text{ANN}}$ and $L_{E\text{MC}}$ is 1.76 W m^{-2} for the validation dataset, which is 0.1 W m^{-2} more than the rms deviation on the learning dataset. The biases between ANN fluxes and MC fluxes are all negative, though small, namely -0.001 N m^{-2} (1%) in τ , -0.12 W m^{-2} ($<1\%$) for H_s , and -0.41 W m^{-2} ($<1\%$) for L_E .

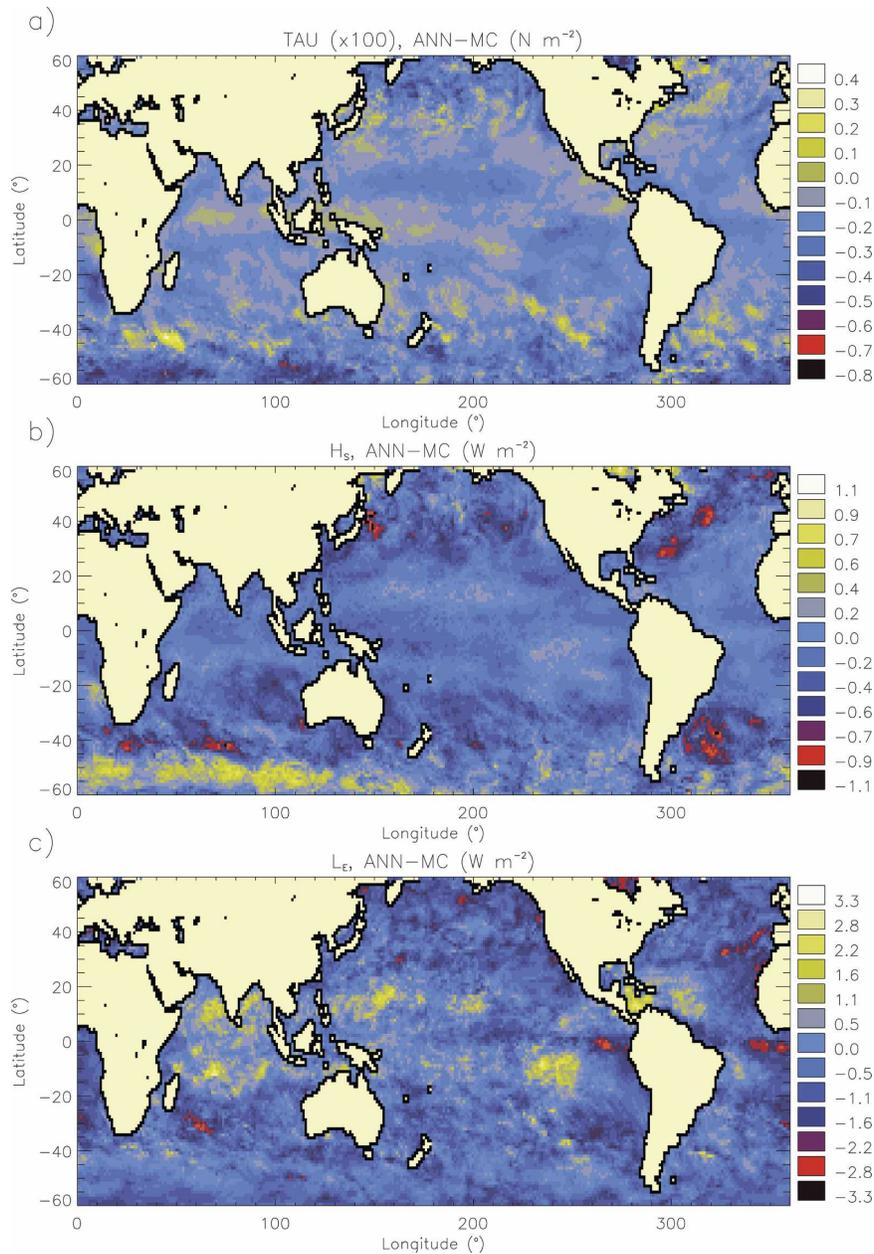


FIG. 5. Spatial representation of the difference between ANN estimates and MC values for (a) τ , (b) H_S , and (c) L_E .

Although encouraging, these results could be misleading because 90% of the fluxes in the validation dataset are within smaller limits: 0 to 0.3 N m⁻² for τ , -20 to 40 W m⁻² for H_S , and -10 to 150 W m⁻² for L_E . The comparisons between ANN fluxes and MC fluxes in these ranges are shown in Fig. 4. The rms deviations between ANN and MC fluxes are 0.003 N m⁻² for τ , 0.4 W m⁻² for H_S , and 1.5 W m⁻² for L_E , which is either smaller or equal to the deviations found with the whole dataset. On the other hand, the biases are larger in Figs.

4a,c than in Fig. 3, namely -0.002 N m⁻² (2%) for τ and -0.8 W m⁻² (1.2%) for L_E . These results show that the linearity of the relationships between MC and ANN fluxes is not perfect, that is, the bias between ANN and MC fluxes depends on the value of the flux. The reason is twofold. First, the biases are related to the use of different ANNs. Furthermore, the clusters of points in Figs. 4b,c clearly show the delimitations between the areas where ANN3-ANN8 were used. In particular, Fig. 4c shows a steplike bias of 4 around 0 W m⁻²,

which corresponds to the transition from ANN5 to ANN6. Next, to a lesser extent, the biases result from a nonperfect learning of the ANNs, that is, the relationships between the ANN fluxes and some of the input variables were not perfectly accounted for by the ANNs during the learning phase. This is noticeable in Fig. 4c, where L_E values calculated with ANN6 have a bias of -0.96 W m^{-2} with respect to L_{EMC} between 0 and 80 W m^{-2} , while their bias is smaller (-0.90 W m^{-2}) between 80 and 150 W m^{-2} .

b. Spatial analysis

The 12 ANN and MC flux fields of the validation dataset were averaged. The rms deviation between the averaged τ_{ANN} and τ_{MC} is 0.001 N m^{-2} , and the bias is -0.0016 N m^{-2} or -1.78% . As shown in Fig. 5, the spatial distribution of the deviation between τ_{ANN} and τ_{MC} is not perfectly homogeneous over the global ocean, although the deviation is smaller than $\pm 0.003 \text{ N m}^{-2}$ for 90% of the ocean. This occurs mostly at latitudes from 0° to $\pm 45^\circ$, with a trend to underestimate τ in the Tropics (10° – 20°) and overestimate it near the equator ($\pm 5^\circ$) and in the subtropics (20° – 30°). The remaining 10% cases correspond to fluxes that are underestimated by 0.008 N m^{-2} (or 8.5%) at maximum. These maxima are located in the higher midlatitudes and high-latitude regions (beyond $\pm 45^\circ$), where the variability of τ_{MC} is large (not shown).

The rms and mean deviations between averaged H_{SANN} and H_{SMC} are 0.22 and -0.12 W m^{-2} , respectively, or $-0.4\% \pm 0.7\%$. An analysis of the deviation between H_{SANN} and H_{SMC} as a function of H_{SMC} revealed that the ANNs tended to overestimate negative flux values by 0 – 0.8 W m^{-2} , while it underestimated up to -0.5 W m^{-2} the positive fluxes smaller than 25 W m^{-2} . Beyond this threshold, the deviation did not depend on H_{SMC} . These biases are clearly visible in Fig. 5b, where the underestimation of H_S is maximum (-1 W m^{-2}) in the neighborhood of the Gulf Stream and the Kuroshio, as well as near 40°S , at locations where H_S is large. Larger biases in H_S are observed at latitudes larger than $\pm 45^\circ$, where the variability of H_S is large. Specifically, regions of strong underestimation and overestimation of H_S are sometimes close to each other, like at 45°S and 50°S . For instance, the maximum deviation between H_{SANN} and H_{SMC} is 2 W m^{-2} along a cross section from 45°S , 80°E to 60°S , 80°E . However, this represents only 2.5% of the error because the flux variation is large, $\sim 80 \text{ W m}^{-2}$, along this cross section.

The L_{EANN} is underestimated by 0.42 W m^{-2} in average with respect to the MC fluxes. The negative bias is found over most of the ocean with peak values of

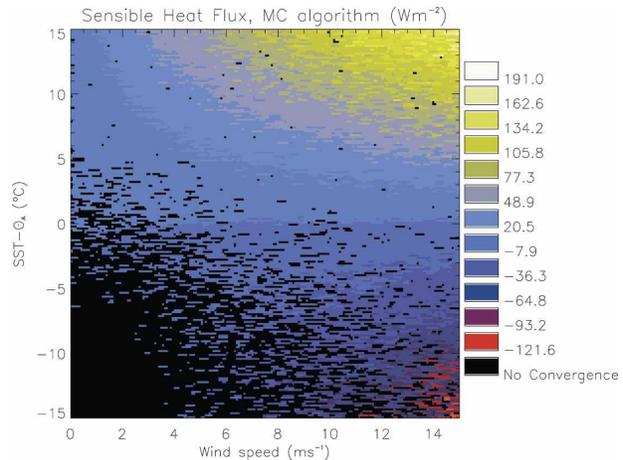


FIG. 6. Representation of the cases for which the MC algorithm did not converge. Only the sensible heat flux is represented, as a function of the wind speed and air–sea temperature difference.

-3 W m^{-2} at the equator and in the subtropics. On the other hand, the ANN fluxes are overestimated by 2.8 W m^{-2} maximum in the Tropics. These results imply that in the Pacific, spatial gradients of H_S along latitudinal cross sections from 0° to $\pm 30^\circ$ are overestimated by 6 W m^{-2} . This is, however, reasonable given that the flux variation exceeds 180 W m^{-2} along such cross sections, that is the maximum error is 3.33%.

c. Cases of no convergence

An analysis of the convergence of the MC algorithm was conducted with data from the all-conditions dataset. It revealed that stable boundary layers were most problematic. Convergence was specifically more difficult to obtain for u_A smaller than 6 m s^{-1} and negative values of $\text{SST} - \theta_A$, as shown in Fig. 6. When the MC algorithm was applied to the validation dataset, convergence was not achieved in 4855 cases out of 268 227, which represents only $\sim 1.8\%$ of the cases. Although this figure is small, this particularity of the MC algorithm may be problematic for certain applications. This is best shown in Fig. 7a, which represents one of the MC flux fields of the validation dataset (0600 UTC 6 April 1997). The numerous black areas in Fig. 7a correspond to the cases for which the MC algorithm did not converge. If this flux field was used to force an ocean model, it would be necessary to use an interpolation scheme to fill the gaps (black areas). Note that although Fig. 7a represents H_{SMC} only, the gaps appear as well in the fields of τ_{MC} and L_{EMC} (not shown). It is difficult to firmly state the accuracy of the ANN fluxes for those cases because no reference flux values are available. However, one may show that the flux values

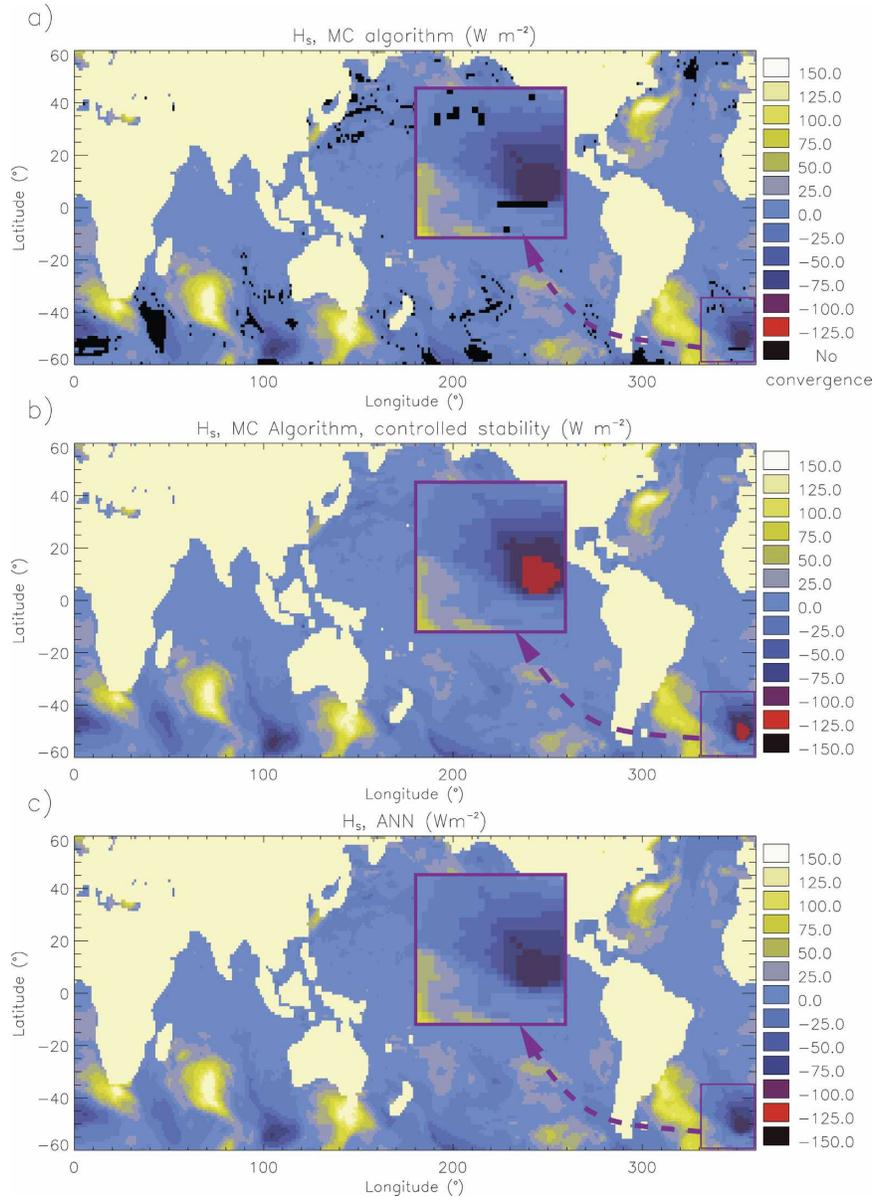


FIG. 7. Spatial distribution of the sensible heat flux H_s issued from (a) MC values, (b) SCMC values, and (c) ANN estimates. The SCMC fluxes correspond to a special version of the MC algorithm, for which ζ is prescribed.

given by the ANNs are spatially consistent, and are consistent with rough flux estimates obtained with a modified version of the MC algorithm that uses a constant ζ . Indeed, if the stability is imposed on the MC algorithm (which of course leads to nonaccurate fluxes), the algorithm converges every time.

Figure 7b shows the field of sensible heat flux that corresponds to the field of Fig. 7a, except that the MC algorithm was applied with a constant stability, equal to $+0.05$ if SST was smaller than θ_A , and -0.05 in the opposite case. This modified version of the MC algo-

rithm is called the stability controlled MC (SCMC) algorithm. Figures 7a,b exhibit similar spatial structures of sensible heat flux. Moreover, the correlation coefficient and rms deviation between the SCMC and MC fluxes are 0.994 and 4.2 W m^{-2} , respectively. These results show that the fluxes calculated with the SCMC algorithm are a reasonably good estimator of the MC fluxes. This is, however, not as good as ANN fluxes, for which we have found an rms deviation of 0.55 W m^{-2} with respect to MC fluxes. Figures 7b,c reveal similar structures of sensible heat flux in the cases where the

TABLE 4. Comparison between ANN fluxes and SCMC fluxes for the cases where the MC algorithm did not converge. The SCMC fluxes correspond to a special version MC algorithm, for which ζ is prescribed.

Variable (units)	Correlation	Rms	Bias	Linear fit
τ (N m^{-2})	0.989	0.009	-0.004	$0.97x - 0.003$
H_S (W m^{-2})	0.989	3.78	2.35	$0.69x - 0.77$
L_E (W m^{-2})	0.998	3.36	0.38	$1.08x - 2.15$

MC algorithm did not converge, which suggests that the ANN fluxes do not diverge significantly from reasonable values of the heat fluxes. Furthermore, the statistical comparison between SCMC fluxes and ANN fluxes in these nonconvergent areas give acceptable deviations for τ , H_S , and L_E , as reported in Table 4. In Figs. 7a–c, a zoom was made on a region of the South Atlantic that corresponds to strong downward sensible heat fluxes (-75 W m^{-2}). The MC algorithm did not converge systematically everywhere in this region (black areas in Fig. 7a). On the other hand, the SCMC algorithm strongly underestimated the negative fluxes, with respect to the MC fluxes (Fig. 7b). Only the ANNs gave a result that was consistent with MC fluxes, but with no gaps. In this particular case, the ANN field was thus a good compromise with respect to the other algorithms, in terms of near-surface stability effects and convergence.

7. Conclusions

An accurate and computationally fast method was proposed for calculating turbulent fluxes at the air–sea interface. The method is a statistical model of a bulk algorithm that is here supposed to be perfect. It is 7 times faster than the bulk algorithm and 40% faster than the recent COARE3.0 algorithm. The accuracy of the method was checked with four datasets representing global, equatorial, and midlatitude environmental conditions, with heights of bulk variables ranging from 3.5 to 35 m. The results indicated that the rms accuracy of the flux estimates was 0.003 N m^{-2} for the momentum flux, 0.5 W m^{-2} for the sensible heat flux, and on the order of 1.8 W m^{-2} for the latent heat flux, for the World Oceans. The systematic errors of the ANNs are -0.001 N m^{-2} , -0.1 W m^{-2} , and -0.4 W m^{-2} , for the momentum, sensible, and latent heat fluxes, respectively.

The comparisons also revealed some learning faults, which result in rms and systematic deviations that depend on environmental conditions and flux values. The learning faults are related to the choice of the configu-

ration and input variables of the ANNs, as well as the selection of the learning datasets. These faults are very difficult to avoid because a very large range of fluxes must be accounted for during the learning phase, while a small range is eventually used when the ANNs are applied. In this context, the linearity of the estimated fluxes must be very good, which is a challenge because it is by essence difficult to control with ANNs. The solution chosen was to use several ANNs for the calculation of each of the three fluxes. As a result, each ANN had a limited range of application, and the linearity and biases were subsequently better controlled. The limitation of this approach is the presence of jumps at the transitions between the use of two ANNs. However, it seems to be a reasonable choice because the resulting errors are acceptable, compared with the uncertainties associated with the use of bulk parameterizations.

The spatial distribution of the systematic errors in ANN fluxes is not homogeneous over the global ocean, with peak deviations at latitudes where the variability of the fluxes is large. However, these peak deviations are acceptable, namely $\pm 0.008 \text{ N m}^{-2}$ for the momentum flux, $\pm 1 \text{ W m}^{-2}$ for the sensible heat flux, and $\pm 3 \text{ W m}^{-2}$ for the moisture flux. It was also shown that the fluxes produced by the ANN at the locations where the bulk algorithm did not converge were spatially consistent, and were consistent with fluxes calculated with a special version of the bulk algorithm whose stability was controlled (i.e., a version that gave fluxes even if the bulk algorithm did not converge), but that was globally not as efficient as the ANN method. The proposed approach seems thus particularly well suited for calculating fluxes in ocean, atmosphere, and ocean–atmosphere coupled models, an effort that is under way at the Centre National de Recherches Météorologiques.

REFERENCES

- Bishop, C. M., 1995: *Neural Networks for Pattern Recognition*. Oxford University Press, 482 pp.
- Brunke, M. A., C. W. Fairall, X. Zeng, L. Eymard, and J. A. Curry, 2003: Which bulk aerodynamic algorithms are least problematic in computing ocean surface turbulent fluxes? *J. Climate*, **16**, 619–635.
- Businger, J. A., J. C. Wyngaard, and Y. Izumi, 1971: Flux profile relationships in the atmospheric surface layer. *J. Atmos. Sci.*, **28**, 181–189.
- Caniaux, G., A. Brut, D. Bourras, H. Giordani, A. Paci, L. Prieur, and G. Reverdin, 2005: A 1 year sea surface heat budget in the northeastern Atlantic basin during the POMME experiment: 1. Flux estimates. *J. Geophys. Res.*, **110**, C07S02, doi:10.1029/2004JC002596.
- Fairall, C. W., E. F. Bradley, D. P. Rogers, J. B. Edson, and G. S. Young, 1996: Bulk parameterization of air–sea fluxes for TOGA COARE. *J. Geophys. Res.*, **101**, 3747–3767.

- , —, J. E. Hare, A. A. Grachev, and J. B. Edson, 2003: Bulk parameterization of air–sea fluxes: Updates and verification for the COARE algorithm. *J. Climate*, **16**, 571–591.
- Grachev, A. A., and C. W. Fairall, 1997: Dependence of the Monin–Obukhov stability parameter on the bulk Richardson number over the ocean. *J. Appl. Meteor.*, **36**, 406–414.
- Liu, W. T., K. B. Katsaros, and J. A. Businger, 1979: Bulk parameterization of air–sea exchanges of heat and water vapor including the molecular constraints at the interface. *J. Atmos. Sci.*, **36**, 1722–1735.
- Mehrotra, K., K. M. Chilukuri, and S. Ranka, 1996: *Elements of Artificial Neural Networks*. MIT Press, 344 pp.
- Mémery, L., G. Reverdin, J. Paillet, and A. Oschlies, 2005: Introduction to the POMME special section: Thermocline ventilation and biogeochemical tracer distribution in the northeast Atlantic Ocean and impact of mesoscale dynamics. *J. Geophys. Res.*, **110**, C07S01, doi:10.1029/2005JC002976.
- Monin, A. S., and A. M. Obukhov, 1954: The main features of turbulent mixing in the surface atmospheric layer. *Tr. Inst. Geophys. Acad. Sci. USSR*, **24**, 163–187.
- Servain, J., A. J. Busalacchi, M. J. McPhaden, A. D. Moura, G. Reverdin, M. Vianna, and S. E. Zebiak, 1998: A Pilot Research Moored Array in the Tropical Atlantic (PIRATA). *Bull. Amer. Meteor. Soc.*, **79**, 2019–2031.
- Smith, S. D., 1980: Wind stress and heat flux over the ocean in gale force winds. *J. Phys. Oceanogr.*, **10**, 709–726.