Algorithms for Vector Field Generation in Mass Consistent Models

Ciro Flores,[1] Héctor Juárez,[2] Marco A. Núñez,[3] María Luisa Sandoval[2]

[1] *Instituto Tecnológico y de Estudios Superiores de Monterrey en Hidalgo,*
Blvd. Felipe Ángeles 2003, C.P. 42080, Pachuca, Hgo., México.

[2] *Departamento de Matemáticas, Universidad Autónoma Metropolitana Iztapalapa,*
A. P. 55–534, C.P. 09340, D. F., México

[3] *Departamento de Física, Universidad Autónoma Metropolitana Iztapalapa, A. P.*
55–534, C.P. 09340, D. F., México

Received 26 February 2008; accepted 5 January 2009
Published online in Wiley InterScience (www.interscience.wiley.com).
DOI 10.1002/num.20458

Diagnostic models in meteorology are based on the fulfillment of some time independent physical constraints as, for instance, mass conservation. A successful method to generate an adjusted wind field, based on mass conservation equation, was proposed by Sasaki and leads to the solution of an elliptic problem for the multiplier. Here we study the problem of generating an adjusted wind field from given horizontal initial velocity data, by two ways. The first one is based on orthogonal projection in Hilbert spaces and leads to the same elliptic problem but with natural boundary conditions for the multiplier. We derive from this approach the so called E–algorithm. An innovative alternative proposal is obtained from a second approach where we consider the saddle–point formulation of the problem—avoiding boundary conditions for the multiplier—and solving this problem by iterative conjugate gradient methods. This leads to an algorithm that we call the CG–algorithm, which is inspired from Glowinski's approach to solve Stokes–like problems in computational fluid dynamics. Finally, the introduction of new boundary conditions for the multiplier in the elliptic problem generates better adjusted fields than those obtained with the original boundary conditions. © 2009 Wiley Periodicals, Inc. Numer Methods Partial Differential Eq 000: 000–000, 2009

Keywords: mass consistent model; variational formulation; finite element method; elliptic problem; saddle–point problem; conjugate gradient

I. INTRODUCTION

For several meteorological problems and a large number of applications, the knowledge of the 3D wind field over a region is required. Examples include prediction of the transport, diffusion, and dispersion of air pollutants in the atmosphere [1,2], and realization of wind maps [3] for the design

Correspondence to: Héctor Juárez, Departamento de Matemáticas, Universidad Autónoma Metropolitana Iztapalapa, A. P. 55–534, C.P. 09340, D. F., México (e-mail: hect@xanum.uam.mx)

© 2009 Wiley Periodicals, Inc.

of different urban and general projects. Moreover, meteorological wind fields are also required inputs for air quality models. In practice, usually limited horizontal wind field measurements are available, and therefore the calculation of the vertical motion must be predicted or calculated. The intrinsic errors associated to the measurements and the sparse data available complicate the estimation of the vertical velocity. This is particularly troublesome in region of complex topography as well in varying land type and areas where land and sea breeze dominate flow patterns [4].

Several methods and strategies, with various levels of complexity, have been proposed to address this problem. They can be included into two general model types: prognostic models and diagnostic models. Prognostic models are complex time-dependent hydrodynamic models governing air flow, including thermal effects, density variation, and turbulent interaction. Although these models are "realistic", they are expensive to operate, need extensive computer facilities, and require specialized training for their operation. On the other hand, diagnostic wind models do not require the integration of the nonlinear hydrodynamic equations. Instead, available interpolated data is used to generate wind fields which satisfy some physical or dynamical constraints. For instance, to assure mass conservation, a simplified steady state version of the continuity equation is imposed and the resulting model is then called a mass–consistent model. A review of these models is available in [5] and [6]. Another model type used to supply fields to air quality models may include a third approach, utilizing four-dimensional data assimilation, which is intended to combine the best features of diagnostic and prognostic approaches by integrating a numerical model in which data are included throughout the integration period. A general review of meteorological modeling for air-quality assessments is available in [7].

In this work, we focus in a variational mass–consistent model which is based in the original formulation by Sasaki [8]. This approach has been used for a variety of meteorological problems, [1–3], [9–16]. Mass-consistent models are attractive because of their simplicity and because they are easy and economical to operate. In some applications, these models outperform the more sophisticated and expensive dynamical models [5]. However, mass-consistent models have some disadvantages, because they are based on incomplete or idealized models and have difficulty representing flows accurately in data-sparse regions as mountains or oceans. Despite these limitations, mass-consistent models are a valuable tool for air quality applications and consequently several developments have taken place over last decades [5, 6, 12, 16–18].

The variational method proposed by Sasaki, uses the continuity equation in the form

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

where $\mathbf{u}$ is the wind velocity vector field on a given domain Ω. The method is based on the minimization of the functional L defined by

$$L(\mathbf{u}, \lambda) = \frac{1}{2} \int_{\Omega} \{S(\mathbf{u} - \mathbf{u}^0) \cdot (\mathbf{u} - \mathbf{u}^0) + \lambda[\nabla \cdot \mathbf{u}]\} dV, \tag{2}$$

where $\mathbf{u}^0$ is an initial observed wind field, λ is a Lagrange multiplier and S is a diagonal matrix with weighting parameters α_i, $i = 1, 2, 3$, called Gaussian precision moduli, related to the scales of the respective components of the velocity field. The vertical component of the initial wind field $\mathbf{u}^0$ is taken as zero because meteorological stations do not measure this component. The Euler–Lagrange equations of the Lagrangian (2) are [5]:

$$\mathbf{u} = \mathbf{u}^0 + S^{-1} \nabla \lambda, \tag{3}$$

$$\lambda \delta \mathbf{u} \cdot \hat{\mathbf{n}} = 0. \tag{4}$$

From (1) and (3) we get the elliptic equation

$$-\nabla \cdot (S^{-1}\nabla\lambda) = \nabla \cdot \mathbf{u}^0. \tag{5}$$

To complement this equation, two types of boundary conditions are commonly used [5]:

1. Dirichlet boundary conditions (for open or "flow through" boundaries)

$$\lambda = 0. \tag{6}$$

2. Neumann boundary conditions (for closed or "no flow through" boundaries)

$$\frac{\partial\lambda}{\partial\hat{\mathbf{n}}} = 0. \tag{7}$$

However, it is not clear which are the proper boundary conditions that λ must satisfy. Some authors [1,5,10,11] claim that Neumann boundary conditions are adequate for closed boundaries like the surface terrain, but in [19,20], it is shown that this condition is inconsistent in general and it must be changed by the more general one

$$-S^{-1}\nabla\lambda \cdot \hat{\mathbf{n}} = \mathbf{u}^0 \cdot \hat{\mathbf{n}}. \tag{8}$$

This boundary condition is reduced to (7) only when S is the identity matrix, and $\mathbf{u}^0 \cdot \hat{\mathbf{n}} = 0$ over the surface terrain. However, these values are not always considered in meteorology. Also, the Dirichlet boundary condition (6) has been widely used in numerous simulations with this model without enough justification. Even though, there have been several sophisticated developments in the numerical simulations of this model as, for instance, the application of multigrid methods [16], it seems that the analysis of the imposed boundary conditions has not attracted the attention of the community in meteorology.

In this work, we study how the boundary conditions for the multiplier λ in the elliptic problem (5) can affect the reconstruction of the wind field in (3). We consider two numerical approaches to generate the adjusted wind field from horizontal data. The first one is the solution by a finite element method of the elliptic problem for the multiplier. The second approach is based on the saddle-point formulation of the optimization problem to which we apply a numerical method derived from a successful methodology in computational fluid dynamics [21]. This new approach does not require boundary conditions for the multiplier, produces better results, and, furthermore, it helped us to establish new and "natural" boundary conditions on the vertical walls for problem (5) .

The organization of this work is as follows. In Section II we write a mathematical model of the problem which is based on a least squares formulation, but with the cost functional defined on a suitable Hilbert space where a unique solution is guarantied. In Section III we derive, from this functional, a saddle-point problem which is, in fact, a degenerated Stokes-like problem. This derivation is based on the orthogonal decomposition of $\mathbf{L}_2(\Omega)$ where the multiplier λ arises in a natural way from the mathematical point of view. From this saddle-point problem the elliptic problem for the multiplier is easily derived. Then, in Section IV we present a numerical result obtained by a finite element method. This result shows that the boundary condition (6) on the vertical walls of the finite domain does not yield a good reconstruction of the wind field there since very high gradients may be introduced by the term $S^{-1}\nabla\lambda$ in (3). In Section V we study the problem following another approach: we find the stationary point of the Lagrangian (2), where no

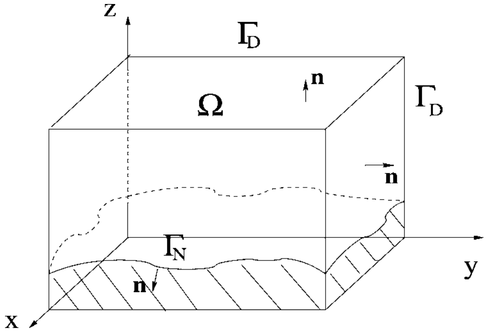

FIG. 1. General domain Ω.

boundary conditions are needed for λ. After reformulating the problem, we introduce a conjugate gradient algorithm which, to our knowledge, has not been applied in meteorology to this kind of problems. However a mixed finite element method is needed in this case to obtain numerical results, as shown in Section VI. A numerical experiment shows that this new approach produces an excellent solution in the interior of the domain as well as on its boundary. From this new formulation we show, in Section VII, that the boundary conditions on the vertical walls must be $\mathbf{u} \cdot \mathbf{n} = \mathbf{u}^0 \cdot \mathbf{n}$ instead of $\lambda = 0$. This new boundary condition implies that the boundary condition for λ on the vertical walls in the elliptic problem (5) must be the natural boundary condition (8) instead of (6). The numerical results show a reduction on the relative error of two orders of magnitude with respect to the solution with the usual boundary condition (6). Finally, in Section VIII we give some concluding remarks.

II. MATHEMATICAL FORMULATION

Let Ω be an open, simply connected and bounded region in $\mathbb{R}^d$ ($d = 2$ or 3) with Lipschitz boundary $\partial\Omega$ which is given by $\partial\Omega = \Gamma_N \cup \Gamma_D$, where Γ_N is the part of the boundary associated to the surface terrain (topography) and Γ_D is the rest of the boundary (the artificial vertical boundaries and the top boundary), as shown in Fig. 1. Given an initial vector field $\mathbf{u}^0$ in Ω (which can be obtained by interpolating atmospheric data, or by other means), our goal is to generate a solenoidal field $\mathbf{u}$—called adjusted field—as close to $\mathbf{u}^0$ as possible in a sense that will be clarified below, such that $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ on Γ_N.

Following Girault and Raviart [22], we define the following vector function spaces

$$\mathbf{L}_2(\Omega) = L_2(\Omega)^d, \qquad \text{with } d = 2 \text{ or } 3,$$
$$\mathbf{H}(div; \Omega) = \{\mathbf{v} \in \mathbf{L}_2(\Omega) : \nabla \cdot \mathbf{v} \in L_2(\Omega)\}.$$

Then, the adjusted wind field $\mathbf{u}$ must belong to the space

$$\mathbf{V} = \{\mathbf{v} \in \mathbf{H}(div; \Omega) : \nabla \cdot \mathbf{v} = 0 \text{ and } \mathbf{v} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N\}, \tag{9}$$

which is equipped with the norm $\| \cdot \|_{S,\Omega}$ associated to the inner product

$$\langle \mathbf{u}, \mathbf{v} \rangle = \int_\Omega (S\mathbf{u}) \cdot \mathbf{v} d\mathbf{x},$$

where $\mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{d} v_i w_i$ is the usual scalar product in $\mathbb{R}^d$, and $S = S(x) \in \mathbb{R}^{d\times d}$ is a symmetric positive definite matrix (usually diagonal). Defining the functional $J : \mathbf{V} \to \mathbb{R}$, by

$$J(\mathbf{v}) = \frac{1}{2} \parallel \mathbf{v} - \mathbf{u}^0 \parallel_{S,\Omega}^2 = \frac{1}{2} \int_\Omega (S(\mathbf{v} - \mathbf{u}^0)) \cdot (\mathbf{v} - \mathbf{u}^0) d\mathbf{x}, \tag{10}$$

the problem to generate the adjusted wind field $\mathbf{u}$ can be stated as follows:

$$\text{Given } \mathbf{u}^0 \in \mathbf{H}(div; \Omega), \text{ find } \mathbf{u} \in \mathbf{V} \text{ such that } J(\mathbf{u}) \leq J(\mathbf{v}), \forall \mathbf{v} \in \mathbf{V}. \tag{11}$$

The functional J is clearly a convex quadratic one. More precisely

$$J(\mathbf{u} + \epsilon\mathbf{v}) = J(\mathbf{u}) + \epsilon J^{(1)}(\mathbf{u}; \mathbf{v}) + \frac{\epsilon^2}{2} J^{(2)}(\mathbf{u}; \mathbf{v}), \tag{12}$$

for all $\epsilon \in \mathbb{R}$ and $\mathbf{u}, \mathbf{v} \in \mathbf{V}$. Above $J^{(1)}$ and $J^{(2)}$ indicate the first and second variations of J given by $J^{(1)}(\mathbf{u}; \mathbf{v}) = \int_\Omega (S(\mathbf{u} - \mathbf{u}^0)) \cdot \mathbf{v}\, d\mathbf{x}$ and $J^{(2)}(\mathbf{u}; \mathbf{v}) = \int_\Omega (S\mathbf{v}) \cdot \mathbf{v} d\mathbf{x}$ respectively. Therefore a necessary and sufficient condition for J to have a minimizer $\mathbf{u} \in \mathbf{V}$ is that

$$\int_\Omega (S(\mathbf{u} - \mathbf{u}^0)) \cdot \mathbf{v}\, d\mathbf{x} = 0, \quad \forall \mathbf{v} \in \mathbf{V}. \tag{13}$$

The Lax–Milgram theorem guaranties that this equation has a unique solution. In the next sections we present two different approaches to study problem (13), from which two corresponding numerical algorithms will be derived to find the solution.

III. THE ELLIPTIC PROBLEM FOR THE MULTIPLIER

The first approach is based on a Helmholtz–type decomposition of the Hilbert vector space $\mathbf{L}_2(\Omega)$:

Proposition 1. *The orthogonal complement in* $\mathbf{L}_2(\Omega)$ *of the closed subspace* $\mathbf{V}$ *is*

$$\mathbf{V}^\perp = \{\nabla q : q \in H^1(\Omega), \quad q = 0 \text{ } on \text{ } \Gamma_D\}.$$

An argument very similar to that given by Girault and Raviart [22], shows that this decomposition is valid (details are given in [23]). As a consequence, from (13), we get

$$S(\mathbf{u} - \mathbf{u}^0) = \nabla\lambda, \tag{14}$$

with λ in

$$H_D^1(\Omega) \equiv \{q \in H^1(\Omega) : q = 0 \text{ on } \Gamma_D\}. \tag{15}$$

According to (1), (9), and (14), $\mathbf{u}$ and λ must satisfy the saddle–point problem

$$S\mathbf{u} - \nabla\lambda = S\mathbf{u}^0 \quad \text{in } \Omega, \tag{16}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \tag{17}$$

$$\mathbf{u} \cdot \hat{\mathbf{n}} = 0 \quad \text{on } \Gamma_N, \tag{18}$$

$$\lambda = 0 \quad \text{on } \Gamma_D, \tag{19}$$

which is a degenerated Stokes-like problem. Eliminating **u** from (16)–(19) we obtain the elliptic equation (5) with the boundary conditions (6) on Γ_D and (8) on Γ_N. Thus, according to the previous discussion, λ is the solution of the following elliptic problem:

$$-\nabla\cdot(S^{-1}\nabla\lambda) = \nabla\cdot\mathbf{u}^0 \quad \text{in } \Omega, \tag{20}$$

$$\lambda = 0 \quad \text{on } \Gamma_D, \tag{21}$$

$$-S^{-1}\nabla\lambda\cdot\hat{\mathbf{n}} = \mathbf{u}^0\cdot\hat{\mathbf{n}} \quad \text{on } \Gamma_N. \tag{22}$$

Once λ is calculated from this problem, the adjusted field is recovered from (3).

Equation (20) has also been formulated by Sasaki [8]. However, Sasaki arrived to such equation from a discussion in which it is not clear how to establish the proper boundary conditions for λ. The crucial argument in our discussion is the decomposition of $\mathbf{L}_2(\Omega)$ in orthogonal subspaces **V** and $\mathbf{V}^{\perp}$, from which the boundary conditions for λ arises in a natural way, from the mathematical point of view.

IV. FINITE ELEMENT SOLUTION OF THE ELLIPTIC PROBLEM

The variational formulation of the elliptic problem (20)–(22) is

$$\int_\Omega S^{-1}\nabla\lambda\cdot\nabla q\,d\mathbf{x} = -\int_\Omega \mathbf{u}^0\cdot\nabla q\,d\mathbf{x}, \quad \forall q\in H_D^1(\Omega). \tag{23}$$

We will consider only the two–dimensional case. Let $\mathcal{T}_h$ be a finite element triangulation of $\overline{\Omega}\subset\mathbb{R}^2$ [24], where h is taken as the space discretization step. Let's denote by P_1 the space of polynomials of degree less or equal than 1. Then, for the elliptic problem, the spaces $\mathbf{L}_2(\Omega)$ and $H_D^1(\Omega)$ will be approximated by the following finite dimensional spaces

$$\mathbf{L}_h = \{\mathbf{v}_h\in\mathcal{C}^0(\bar{\Omega})^2 : \mathbf{v}_h|_T\in P_1\times P_1, \quad \forall T\in\mathcal{T}_h\},$$

$$H_{Dh}^1 = \{q\in\mathcal{C}^0(\bar{\Omega}) : q|_T\in P_1, \quad \forall T\in\mathcal{T}_h,\ q=0 \text{ on } \Gamma_D\}, \tag{24}$$

respectively. Thus the finite element algorithm is:

Given $\mathbf{u}_h^0\in\mathbf{L}_h$, find $\lambda_h\in H_{Dh}^1$ such that

$$\int_\Omega S^{-1}\nabla\lambda_h\cdot\nabla q\,d\mathbf{x} = -\int_\Omega \mathbf{u}_h^0\cdot\nabla q\,d\mathbf{x}, \quad \forall q\in H_{Dh}^1, \tag{25}$$

where $\mathbf{u}_h^0\in\mathbf{L}_h$ is the interpolant of the given initial velocity field $\mathbf{u}^0$. The resulting system of linear equations is solved by the conjugate gradient algorithm adapted to a sparse matrix [25]. Once λ_h is computed, the numerical approximation $\mathbf{u}_h$ to the adjusted field **u** is computed by the weak version of (3) as follows:

Find $\mathbf{u}_h\in\mathbf{L}_h$ with $\mathbf{u}_h\cdot\hat{\mathbf{n}}=0$ on Γ_N such that

$$\int_\Omega (S\mathbf{u}_h)\cdot\mathbf{v}\,d\mathbf{x} = \int_\Omega (S\mathbf{u}_h^0)\cdot\mathbf{v}\,d\mathbf{x} - \int_\Omega \lambda_h\nabla\cdot\mathbf{v}\,d\mathbf{x}, \quad \forall\mathbf{v}\in\mathbf{L}_h \tag{26}$$

where $\mathbf{v}\cdot\hat{\mathbf{n}}=0$ on Γ_N. We call (25)–(26) the *E–algorithm* to compute the adjusted wind field **u**.

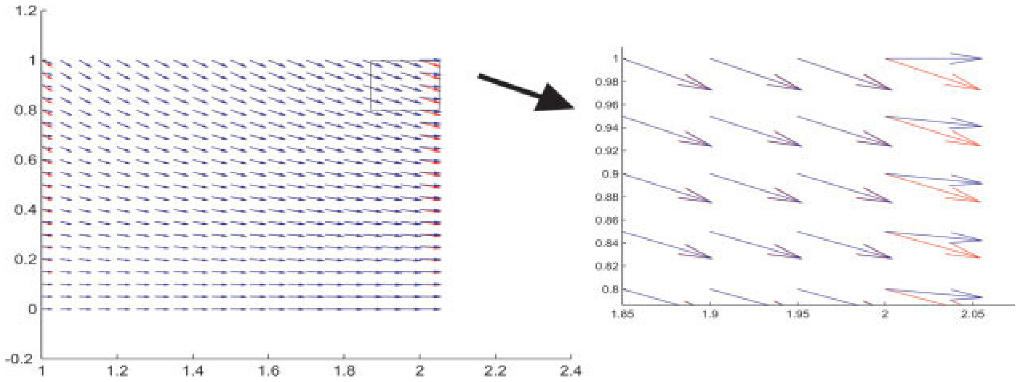

FIG. 2. Exact field $\mathbf{u} = (x, -z)$ in red, adjusted field obtained by the E–algorithm in blue. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Example 1. We consider the two dimensional solenoidal vector field $\mathbf{u}(x, z) = (x, -z)$ defined in $\Omega = (1, 2) \times (0, 1)$, which satisfies the conditions (17)–(18). Assuming that we have $\mathbf{u}^0(x, z) = (x, 0)$ as an initial horizontal wind field, we want to see how much we can recover of the vertical component of $\mathbf{u}$ by applying the previous algorithm. For this calculation, Ω is divided by a 80×80 triangular mesh, and the Gaussian precision moduli are given the values $\alpha_1 = 1$ and $\alpha_3 = 0.001$. Figure 2 shows the exact field in red and the computed adjusted field in blue. Both fields agree very well almost everywhere except on the vertical artificial boundaries $x = 1$ and $x = 2$. To measure the global difference between the exact field $\mathbf{u}$ and the computed adjusted field $\mathbf{u}_h$ we take the relative error

$$e_r = \frac{\|\mathbf{u} - \mathbf{u}_h\|_2}{\|\mathbf{u}\|_2}, \tag{27}$$

For the present example we get $e_r = 1.9 \times 10^{-2}$. We also computed a mean value of the divergence of $\mathbf{u}_h$, defined as

$$mdiv = \underset{\mathbf{x}_i}{\text{mean}}\{\nabla \cdot \mathbf{u}_h(\mathbf{x}_i) | \mathbf{x}_i \text{ is a interior vertex of the computational mesh}\} \tag{28}$$

where the point-wise divergence is computed in a weak sense

$$\nabla \cdot \mathbf{u}_h(\mathbf{x}_i) = -\int_\Omega \mathbf{u}_h \cdot \nabla \phi_i \, d\mathbf{x}, \tag{29}$$

where ϕ_i is the piece-wise linear base function associated to node $\mathbf{x}_i$. For the present example we get $mdiv = 4.1 \times 10^{-2}$. The values for the Gaussian precision moduli were chosen based on numerical performance. Table I shows the behavior of e_r and $mdiv$ for different values of α_3 when α_1 is kept constant to one. It is clear that the best result is obtained with $\alpha_3 = 0.001$. Additional discussion about the election of these parameters can be found in [20, 23].

V. SADDLE-POINT FORMULATION OF THE PROBLEM

The second method to solve problem (13), or equivalently problem (11), is based on the usual methodology to solve optimization problems subject to constraints. The key step is to relax the

TABLE I. Numerical performance of E–algorithm for different values of α_3.

α_3	e_r	mdiv
0.001	1.9×10^{-2}	4.1×10^{-2}
0.01	9.6×10^{-2}	-6.1×10^{-2}
0.1	1.4×10^{-1}	2.9×10^{-1}
1	5.2×10^{-1}	5.4×10^{-1}
100	6.4×10^{-1}	7.8×10^{-1}
1000	9.8×10^{-1}	9.8×10^{-1}

incompressibility condition (1), introducing a new unknown: the Lagrange multiplier λ. More precisely, we introduce the space of vector functions

$$\mathbf{V}_N = \{\mathbf{v} \in \mathbf{H}(div;\Omega) : \mathbf{v}\cdot\hat{\mathbf{n}} = 0 \text{ on } \Gamma_N\}, \tag{30}$$

together with the Lagrangian L defined on $\mathbf{V}_N \times L_2(\Omega)$ as

$$L(\mathbf{v},q) \equiv J(\mathbf{v}) + \langle q, \nabla\cdot\mathbf{v}\rangle = \frac{1}{2}\int_\Omega (S(\mathbf{v}-\mathbf{u}^0))\cdot(\mathbf{v}-\mathbf{u}^0)\,d\mathbf{x} + \int_\Omega q\nabla\cdot\mathbf{v}\,d\mathbf{x}.$$

A stationary point $(\mathbf{u},\lambda)$ of L solves the following saddle–point problem

$$\int_\Omega S\mathbf{u}\cdot\mathbf{v}\,d\mathbf{x} + \int_\Omega \lambda\nabla\cdot\mathbf{v}\,d\mathbf{x} = \int_\Omega S\mathbf{u}^0\cdot\mathbf{v}\,d\mathbf{x}, \quad \forall\mathbf{v}\in\mathbf{V}_N, \tag{31}$$

$$\int_\Omega q\nabla\cdot\mathbf{u}\,d\mathbf{x} = 0, \quad \forall q \in L_2(\Omega). \tag{32}$$

The solution $\mathbf{u}$ is the minimizer of J, and now is obtained from the enlarged space $\mathbf{V}_N$ where free divergence is not required. Instead the condition $\nabla\cdot\mathbf{u} = 0$ is relaxed by the introduction of the Lagrange multiplier λ so that $\mathbf{u}$ must satisfy the weaker condition (32). To solve (31)–(32) we introduce a method inspired by Glowinski's proposal which has shown to be very effective for solving Stokes problems in computational fluid dynamics (CFD) [21]. The idea is as follows: assuming that $(\mathbf{u},\lambda)$ is solution of problem (31)–(32), the vector field $\mathbf{u}$ is decomposed as $\mathbf{u} = \mathbf{u}^0 + \mathbf{u}_\lambda$, where $\mathbf{u}^0$ is the given initial vector field, and $\mathbf{u}_\lambda \in \mathbf{V}_N$ solves

$$\int_\Omega S\mathbf{u}_\lambda\cdot\mathbf{v}\,d\mathbf{x} = -\int_\Omega \lambda\nabla\cdot\mathbf{v}\,d\mathbf{x}, \quad \forall\mathbf{v}\in\mathbf{V}_N. \tag{33}$$

Furthermore, $\mathbf{u}_\lambda$ must satisfy (32) which has an equivalent strong version

$$\nabla\cdot\mathbf{u}_\lambda = -\nabla\cdot\mathbf{u}^0. \tag{34}$$

Problem (33)–(34) for $\mathbf{u}_\lambda$ can be formulated as a functional equation. For this we define the linear operator A from $L_2(\Omega)$ into $L_2(\Omega)$ given by

$$Aq = \nabla\cdot\mathbf{u}_q, \tag{35}$$

where $\mathbf{u}_q \in \mathbf{V}_N$ is the solution of

$$\int_\Omega S\mathbf{u}_q\cdot\mathbf{v}\,d\mathbf{x} = -\int_\Omega q\nabla\cdot\mathbf{v}\,d\mathbf{x}, \quad \forall\mathbf{v}\in\mathbf{V}_N. \tag{36}$$

With this definition it is clear from (33)–(34) that the multiplier λ satisfies the functional equation

$$A\lambda = -\nabla \cdot \mathbf{u}^0, \tag{37}$$

where the operator A is symmetric and positive definite. An iterative conjugate gradient algorithm can be applied to solve (37) and, in turn, to obtain the solution of the saddle–point problem (31)–(32):

Conjugate gradient algorithm.

1. $\lambda^0 \in L_2(\Omega)$ given.
2. Solve for $\mathbf{u}_{\lambda^0} \in \mathbf{V}_N$

$$\int_\Omega (S\mathbf{u}_{\lambda^0}) \cdot \mathbf{v}\, d\mathbf{x} = \int_\Omega \lambda^0 \nabla \cdot \mathbf{v}\, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N.$$

3. $g^0 = \nabla \cdot (\mathbf{u}^0 + \mathbf{u}_{\lambda^0})$.
4. $d^0 = g^0$.
   For $m \geq 0$, assuming we know λ^m, g^m, d^m and $\mathbf{u}^m$, find λ^{m+1}, g^{m+1}, d^{m+1} and $\mathbf{u}^{m+1}$ by:
5. Solve for $\bar{\mathbf{u}}^m \in \mathbf{V}_N$:

$$\int_\Omega (S\bar{\mathbf{u}}^m) \cdot \mathbf{v}\, d\mathbf{x} = \int_\Omega d^m \nabla \cdot \mathbf{v}\, d\mathbf{x}, \quad \forall \mathbf{v} \in \mathbf{V}_N.$$

6. $\bar{g}^m = \nabla \cdot \bar{\mathbf{u}}^m$.
7. $\alpha_m = \int_\Omega |g^m|^2\, d\mathbf{x} \,/ \int_\Omega \bar{g}^m\, d^m\, d\mathbf{x}$.
8. $\lambda^{m+1} = \lambda^m - \alpha_m d^m$.
9. $\mathbf{u}^{m+1} = \mathbf{u}^m - \alpha_m \bar{\mathbf{u}}^m$.
10. $g^{m+1} = g^m - \alpha_m \bar{g}^m$.
    If $\int_\Omega |g^{m+1}|^2\, d\mathbf{x} \,/ \int_\Omega |g^0|^2\, d\mathbf{x} < \varepsilon$, take $\lambda = \lambda^{m+1}$, $\mathbf{u} = \mathbf{u}^{m+1}$ and stop. Otherwise continue with:
11. $\beta_m = \int_\Omega |g^{m+1}|^2\, d\mathbf{x} \,/ \int_\Omega |g^m|^2\, d\mathbf{x}$.
12. $d^{m+1} = g^{m+1} + \beta_m d^m$.
13. Do $m = m + 1$ and return to 5.

Observe that the adjusted field $\mathbf{u}$ is also computed as an intermediate step in the algorithm. In this algorithm no boundary condition is imposed on the multiplier λ, contrary to what it was done in the first approach. This fact has a very important effect in the numerical solution as we will show in the next section.

VI. A MIXED FINITE ELEMENT METHOD TO SOLVE THE SADDLE-POINT PROBLEM

To approximate the functions in $\mathbf{V}_N$ and $L_2(\Omega)$, considered in the previous section, we make use of the Bercovier-Pironneau finite element approximation [26]: functions in $L_2(\Omega)$ are approximated by continuous piecewise linear polynomials over a triangulation $\mathcal{T}_h$ of Ω, while the elements in $\mathbf{V}_N$ are also approximated by linear polynomials but now over a twice finer triangulation $\mathcal{T}_{h/2}$ of Ω. The fine triangulation $\mathcal{T}_{h/2}$ is obtained from a regular subdivision of each triangle $T \in \mathcal{T}_h$, as

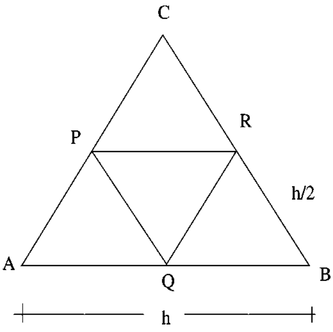

FIG. 3. Element in $\mathcal{T}_h$: triangle ABC. Elements in $\mathcal{T}_{h/2}$: triangles AQP, PRC, PQR, and QBR

is shown in Fig. 3. Then, the functional spaces $\mathbf{V}_N$ and $L_2(\Omega)$ will be approximated by the finite dimensional spaces

$$\mathbf{V}_{Nh} = \{\mathbf{v}_h \in C^0(\bar{\Omega})^2 : \mathbf{v}_h|_T \in P_1 \times P_1, \quad \forall T \in \mathcal{T}_{h/2}, \quad \mathbf{v}_h \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N\},$$

and

$$L_h = \{q_h \in C^0(\bar{\Omega}) : q_h|_T \in P_1, \forall T \in \mathcal{T}_h\},$$

respectively.

We apply this mixed method particularly in Steps 2 and 5 as well as in the weak version of Steps 3 and 6 of the conjugate gradient algorithm. Also, if the trapezoidal rule is applied to calculate the integrals of the left hand side in Steps 2 and 5, we obtain a system of algebraic equations with diagonal matrix, and the cost to solve them is just a vector multiplication. We call this new algorithm the CG–algorithm.

Example 2. We consider again the problem in Example 1 where the initial horizontal field is $\mathbf{u}^0 = (x, 0)$. We want to see how much we can reconstruct the solenoidal exact vector field $\mathbf{u} = (x, -z)$ in $\Omega = (1, 2) \times (0, 1)$. To compare the numerical results with those obtained previously with the E–algorithm, we chose $h = 1/40$ and $h/2 = 1/80$ in this case. To stop the iterations we choose $\varepsilon = 10^{-8}$ at step 10 in the CG–algorithm (this tolerance value is also used in all the subsequent examples solved by this method). Figure 4 shows the exact field $\mathbf{u}$ (in red) and the adjusted field $\mathbf{u}_h$ (in blue). The agreement is excellent even at the boundaries $x = 1$ and $x = 2$, in contrast to the numerical results obtained in Section IV. This time we obtained $e_r = 5.9 \times 10^{-4}$ and $mdiv = -5.3 \times 10^{-12}$, and we may say that the CG–algorithm yields numerical solutions which are much better than those obtained by the E–algorithm.

VII. NEW BOUNDARY CONDITIONS FOR λ IN THE ELLIPTIC PROBLEM

From Eqs. (31)–(32), we obtain

$$\int_\Omega (S\mathbf{u} - \nabla\lambda - S\mathbf{u}^0) \cdot \mathbf{v}\, d\mathbf{x} = \int_{\Gamma \setminus \Gamma_N} \lambda \mathbf{v} \cdot \hat{\mathbf{n}}\, d\Gamma, \quad \forall \mathbf{v} \in \mathbf{V}_N, \tag{38}$$

$$\int_\Omega q \nabla \cdot \mathbf{u}\, d\mathbf{x} = 0, \quad \forall q \in L_2(\Omega). \tag{39}$$

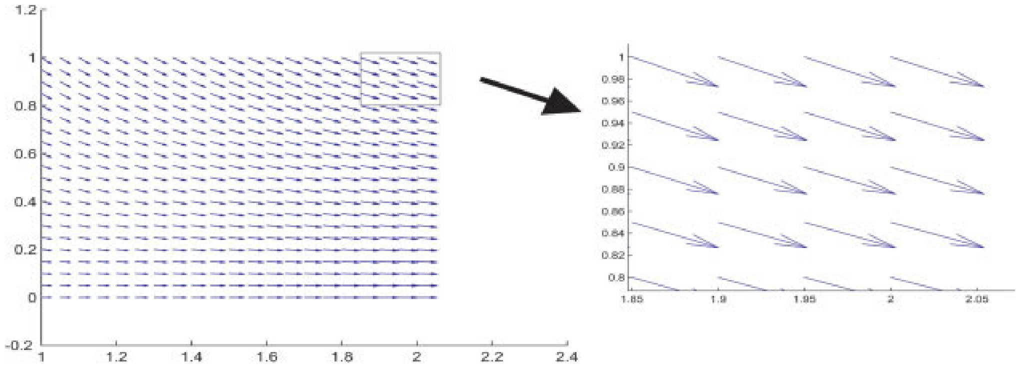

FIG. 4. Exact field $\mathbf{u} = (x, -z)$ in red, adjusted field obtained by the CG–algorithm in blue.

The last integral in (38) vanishes in two cases, namely: $\mathbf{v} \cdot \hat{\mathbf{n}} = 0$ or $\lambda = 0$ on $\Gamma \setminus \Gamma_N$. The first case is not possible since it holds only on Γ_N, and the second case is not a good choice on vertical boundaries as we have seen in the previous sections. So, if we want the boundary integral in (38) to vanish, there is a possibility: redefine Γ_D as the top boundary of the domain and impose $\lambda = 0$ there, and let Γ_V denote the vertical boundaries where we still need to impose a boundary condition. Since the only information we have on those vertical walls is $\mathbf{u}^0$, it is reasonable to impose $\mathbf{u} \cdot \hat{\mathbf{n}} = \mathbf{u}^0 \cdot \hat{\mathbf{n}}$ on Γ_V. Therefore, with this choice we obtain the saddle-point problem

$$S\mathbf{u} - \nabla\lambda = S\mathbf{u}^0, \text{ in } \Omega, \tag{40}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \tag{41}$$

$$\lambda = 0 \quad \text{on } \Gamma_D, \tag{42}$$

$$\mathbf{u} \cdot \hat{\mathbf{n}} = \mathbf{u}^0 \cdot \hat{\mathbf{n}} \quad \text{on } \Gamma_V, \tag{43}$$

$$\mathbf{u} \cdot \hat{\mathbf{n}} = 0 \quad \text{on } \Gamma_N. \tag{44}$$

Eliminating $\mathbf{u}$ from this set of equations we get the following elliptic problem for λ with new boundary conditions:

$$-\nabla \cdot (S^{-1}\nabla\lambda) = \nabla \cdot \mathbf{u}^0 \text{ in } \Omega, \tag{45}$$

$$\lambda = 0 \text{ on } \Gamma_D, \tag{46}$$

$$-S^{-1}\nabla\lambda \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_V, \tag{47}$$

$$-S^{-1}\nabla\lambda \cdot \hat{\mathbf{n}} = \mathbf{u}^0 \cdot \hat{\mathbf{n}} \text{ on } \Gamma_N. \tag{48}$$

The finite element algorithm for this problem is:

Given $\mathbf{u}_h^0 \in \mathbf{L}_h$, find $\lambda_h \in H_{Dh}^1$ such that

$$\int_\Omega S^{-1}\nabla\lambda_h \cdot \nabla q \, d\mathbf{x} = -\int_\Omega \mathbf{u}_h^0 \cdot \nabla q \, d\mathbf{x} + \int_{\Gamma_V} q\mathbf{u}_h^0 \cdot \mathbf{n} \, d\Gamma, \quad \forall q \in H_{Dh}^1, \tag{49}$$

where H_{Dh}^1 is defined as in (24), but with Γ_D representing only the top boundary of the domain Ω. Notice that Eq. (49) differs from Eq. (25) only by the surface integral on Γ_V (actually a line

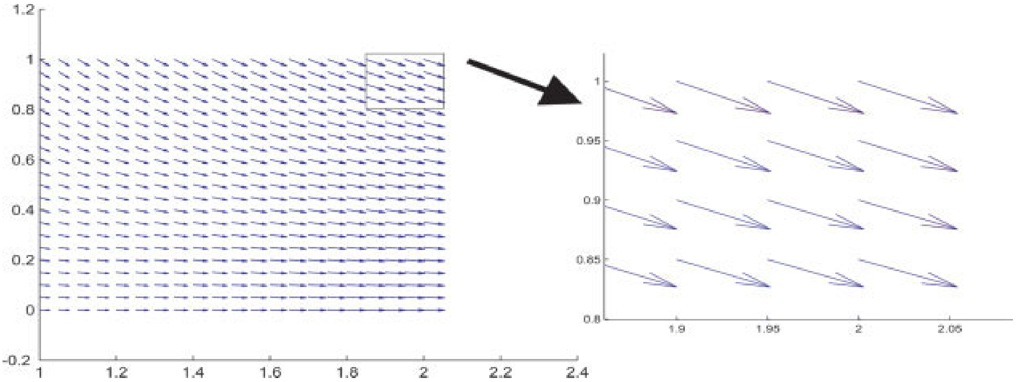

FIG. 5. Exact field $\mathbf{u} = (x, -z)$ in red, adjusted field compued by the E2–algorithm in blue.

integral in 2D), and we hope this surface integral improves the computed adjusted wind field on the artificial vertical boundaries. We call (49) along with (26) the E2–algorithm.

Example 3. Let us consider one more time the problem in Example 1 with $\mathbf{u}^0 = (x, 0)$, and let's take $h = 1/80$. We want to see how much we can recover of the vertical component of $\mathbf{u} = (x, -z)$ when applying the E2–algorithm. Figure 5 shows the exact field and the computed-adjusted field. The agreement is better now, since we recover the vertical component fairly well not only in the interior of the domain but also at the vertical boundaries $x = 1$ and $x = 2$, as can be verified in the amplification. The immediate effect of this improvement at the lateral boundaries is the reduction of the relative error by two orders of magnitude, since the relative error obtained now is $e_r = 4 \times 10^{-4}$. However, we do not obtain a comparable reduction of the mean divergence which value now is $mdiv = 1.8 \times 10^{-2}$ (about half of the one obtained in Example 1). As can be observed in Example 2, the CG–algorithm is a more effective algorithm to reduce the average weak divergence. A summary of the results in Examples 1, 2, and 3 is shown in Table II. In this and subsequent tables we also include the CPU time needed to compute the solution in each case. All numerical calculations were performed in a DELL Latitude D610 2.13 GHz laptop with an Intel Pentium M processor and 2 GB of RAM.

We consider two more "realistic" additional examples to test the reliability of the two algorithms considered in this work. The first one includes a domain with a topography of a cosine-shape, and the second one with a real topography. In both cases the "exact" vector wind field is synthetic and it was generated by a Stokes solver [21] with a numerical weak divergence of the order of 10^{-16}. We drop the vertical component of the computed vector field to obtain the initial horizontal vector field $\mathbf{u}^0$. Our objective is to apply the last two algorithms to see how much we can recover of the vertical component.

TABLE II. Summary of the numerical results obtained in the first three examples.

Ex.	Exact field	Algorithm	e_r	mdiv	No. iters.	CPU time(s)
1	$\mathbf{u} = (x, -z)$	E–algorithm	1.9×10^{-2}	4.1×10^{-2}	—	1.78
2	$\mathbf{u} = (x, -z)$	CG–algorithm	5.9×10^{-4}	-5.3×10^{-12}	1214	3.9
3	$\mathbf{u} = (x, -z)$	E2–algorithm	4×10^{-4}	1.8×10^{-2}	—	1.78

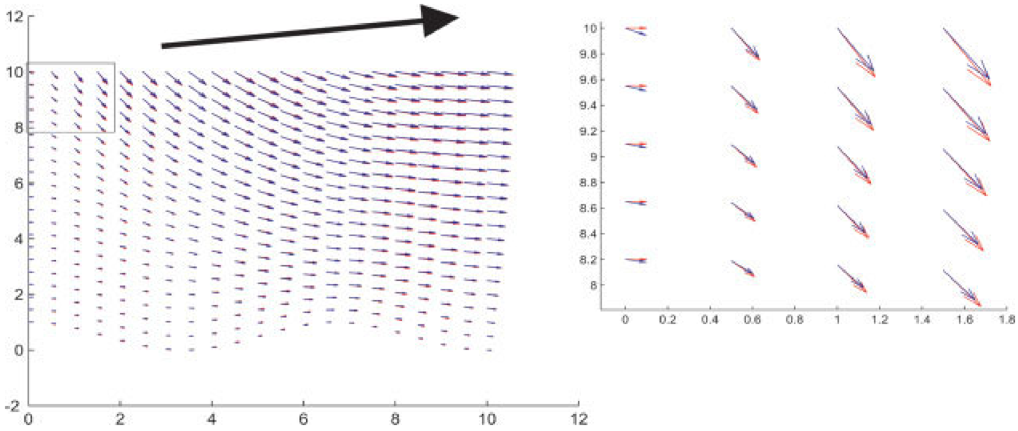

FIG. 6. "Exact" field for cosine topography in red, adjusted field obtained by the E2–algorithm in blue. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Example 4. In this case we define the domain as follows

$$\Omega = \left\{(x, y) \in \mathbb{R}^2 : 0 < x < 10, \quad \frac{1}{2}\cos\frac{3\pi x}{10} + \frac{1}{2} < y < 10\right\}.$$

The "exact" wind field satisfies $\nabla \cdot \mathbf{u} = 1.2 \times 10^{-16}$. First, we solve the problem (45)–(48) with the E2–algorithm and $h = 1/80$. Figure 6 shows the exact and adjusted wind fields. The highest difference between $\mathbf{u}$ and $\mathbf{u}_h$ occurs on the boundary, more precisely close to the left upper corner. Next, we solve the saddle–point problem (31)–(32) with the CG–algorithm. In this case we set $h = 1/40$ and $h/2 = 1/80$ to get the same numerical resolution. It is important to say that we modify the space of test functions $\mathbf{V}_N$ in (30) to be

$$\mathbf{V}_N = \{\mathbf{v} \in \mathbf{H}(div; \Omega) : \mathbf{v} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma_N \cup \Gamma_V\}.$$

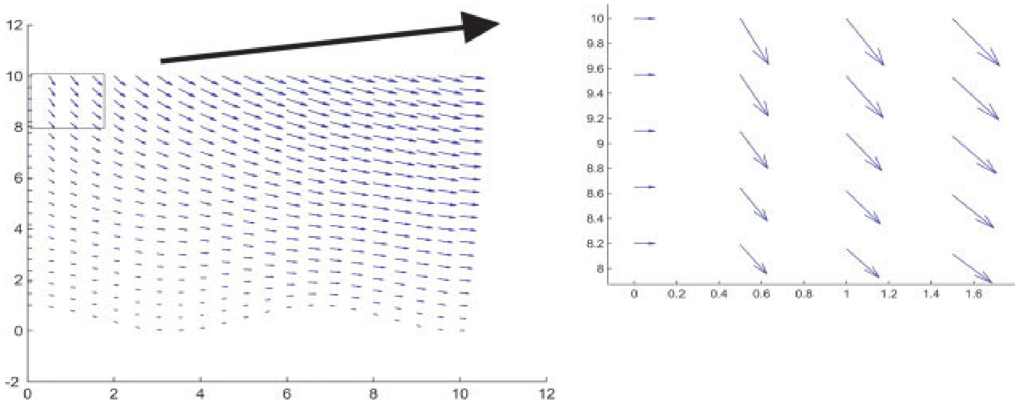

FIG. 7. "Exact" field for cosine topography in red, adjusted field obtained by the CG–algorithm in blue.

TABLE III. Summary of numerical results for Example 4.

Ex.	Case with	Algorithm	e_r	mdiv	No. iters.	CPU time(s)
4	Cosine topography	E2–algorithm	9.2×10^{-2}	3.6×10^{-4}	—	2.08
4	Cosine topography	CG–algorithm	3.6×10^{-7}	9.8×10^{-9}	955	3.3

Cosine–shape topography.

Figure 7 shows that the difference between $\mathbf{u}$ and $\mathbf{u}_h$ has been reduced significantly with respect to the numerical calculation with the E2–algorithm. Also the numerical divergence obtained is reduced by five orders of magnitude. Table III shows a summary of the numerical results.

Example 5. In this case the domain is defined as

$$\Omega = \{(x, y) \in \mathbb{R}^2 : 0 < x < 10, \quad h(x) < y < 10\},$$

where $h(x)$ is a function constructed via cubic splines which interpolates discrete data over 10 km of real topography of certain region in Mexico, contained in a data base [27]. The "exact" wind field satisfies $\nabla \cdot \mathbf{u} = 6.1 \times 10^{-16}$. As in the previous example we drop the vertical component of this vector field and compute the adjusted field using the two methodologies with the same discretization parameters. Figure 8 shows the exact wind field $\mathbf{u}$ and the adjusted field $\mathbf{u}_h$ obtained by the E2–algorithm. The highest difference between $\mathbf{u}$ and $\mathbf{u}_h$ occurs again close to the vertical walls. When we solve the saddle–point problem (31)–(32) with the CG–algorithm we obtain the results shown on Figure 9. We have an excellent adjustment with an almost free divergence computed field. Table IV shows a summary of the numerical results.

VIII. CONCLUSIONS

We studied the problem of generating an adjusted wind field from horizontal wind data by two numerical methodologies. The usual methodology associated to the E–algorithm, does not produce satisfactory results close to the vertical boundaries due to the high gradients introduced

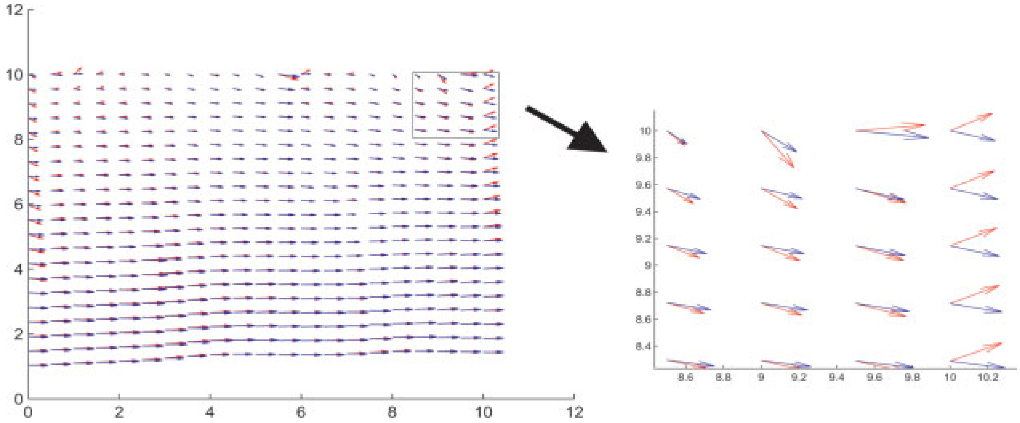

FIG. 8. "Exact" field for real topography in red, adjusted field obtained by the E2–algorithm in blue. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

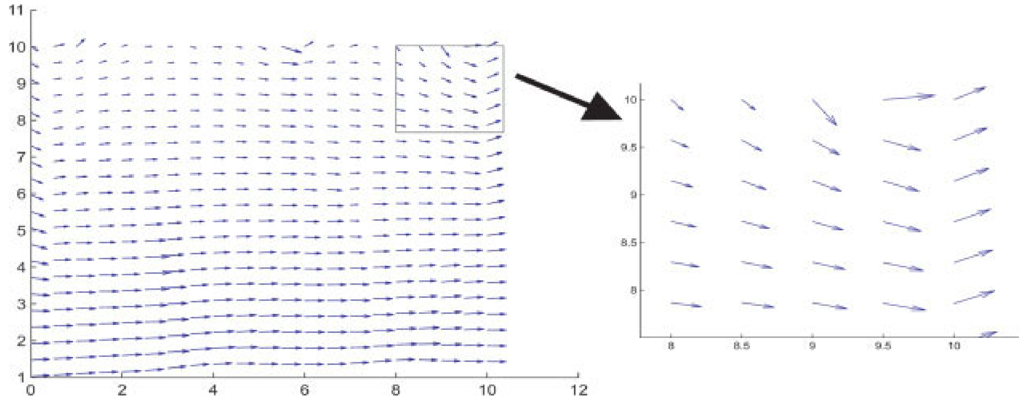

FIG. 9. "Exact" field for real topography in red, adjusted field obtained by the CG–algorithm in blue.

by the term $S^{-1}\nabla\lambda$ in (3) when the boundary condition (6) is imposed there. Also the discrete "solenoidal" vector field obtained with this method has a weak divergence only of the order of 10^{-2}. The formulation of the problem as a saddle-point one, with a functional equation that has a symmetric and positive definite operator, allows the use of an iterative conjugate gradient algorithm (the CG–algorithm) which produces much better results. For the recovery of the vertical component of the vector field $\mathbf{u} = (x, -z)$ the relative error is reduced by two orders of magnitude while the weak divergence is reduced by ten orders of magnitude, as Table II shows. We want to emphasize that this new methodology does not involve the solution of a differential equation and, therefore, no boundary conditions are required for the multiplier λ. Also, the CPU time to compute the solution with the CG–algorithm is slightly more than twice the one needed with the usual E–algorithm despite the large number of iterations done by the CG-algorithm.

In an attempt to improve the results obtained with the E–algorithm, we introduce new boundary conditions for the multiplier in the elliptic problem. Using the modified model (45)–(48) and the E2–algorithm, the reconstruction is quite good. The relative error is also reduced by two orders of magnitude, as shown in Table II, but there is no comparable reduction on the weak divergence. However for the cases in Examples 4 and 5, where the topography is more complicated, the mean weak divergence obtained with the E2–algorithm is of the order of 10^{-4} and 10^{-5}, respectively, which is satisfactory in many applications. The application of the CG–algorithm to these two cases produces an additional significant improvement as shown in Tables III and IV: the relative error is reduced by four to five orders of magnitude while the weak divergence is reduced by five to six orders of magnitude. Again, the CPU time to compute the solution with the CG–algorithm is about twice the CPU time needed with the traditional algorithm.

The next step is the introduction of a preconditioner in order to reduce the number of iterations in the CG–algorithm, and also to reduce the CPU time to at least a comparable time needed with the traditional algorithm. It can be proved that the elliptic problem is actually

TABLE IV. Summary of numerical results for Example 5.

Ex.	Case with	Algorithm	e_r	mdiv	No. iters.	CPU time(s)
5	Real topography	E2–algorithm	9.7×10^{-2}	6.8×10^{-5}	—	1.78
5	Real topography	CG–algorithm	4.1×10^{-6}	5.7×10^{-11}	1000	3.7

Real topography.

an optimal preconditioner of this iterative algorithm. Work is in progress for this issue. On the other hand, the application of the methodology presented here to the more realistic three-dimensional case is a continuation of present work. Another interesting issue is the potential application of these methodologies to other experimental fields such as fluid dynamics. In particular, recently we have been told about the importance of the reconstruction of solenoidal velocity fields from experimental data obtained by means of the PIV (Particle Image Velocimetry) technique [28].

References

1. C. A. Sherman, A mass–consistent model for wind fields over complex terrain, J Appl Meteor 17 (1978), 312–319.
2. S. Finardi, G. Tinarelli, A. Nanni, G. Brusasca, and G. Carboni, Evaluation of a 3–D flow and pollutant dispersion modelling system to estimate climatological ground level concentrations in complex coastal sites, Int J Environ Pollution 16 (2001), 472–482.
3. F. Castino, L. Rusca, and G. Solari, Wind climate micro–zoning: a pilot application to Liguria Region (North–Western Italy), J Wind Eng Ind Aerodynam 91 (2003), 1353–1375.
4. G. Gross, On the applicability of mass–consistent wind field models, Boundary–Layer Meteorol 77 (1996), 379–394.
5. C. F. Ratto, R. Festa, C. Romeo, O. A. Frumento, and M. Galluzzi, Mass–consistent models for wind fields over complex terrain: the state of the art, Environ Software 9 (1994), 247–268.
6. C. F. Ratto, An overview of mass-consistent models, Modeling of atmosphere flow fields, Lalas, D. P., Ratto, C. F. editors, World Scientific Publications, Singapore, 1996, pp. 379–400.
7. N. L. Seaman, Meteorological modeling for air–quality assessments, Atmospher Environ 34 (2000), 2231–2259.
8. Y. Sasaki, An objective analysis based on the variational method, J Met Soc Jpn 36 (1958), 77–88.
9. W. T. Pennel, An evaluation of the role of numerical wind field models in wind turbine siting, Technical Report PNL–SA–11129 (1983).
10. T. Kitada, A. Kaki, Ueda H., and Peters L. K., Estimation of the vertical air motion from limited horizontal wind data–A numerical experiment, Atmos Environ 17 (1983), 2181–2192.
11. T. Kitada and K. Igarashi, Numerical Analysis of Air Pollution in a Combined Field of Land/Sea Breeze and Mountain/Valley Wind, Climite Appl Met 25 (1986), 767–784.
12. D. G. Ross, I. N. Smith, P. C. Manins, and D. G. Fox, Diagnostic wind field modeling for complex terrain: model development and testing, J Appl Meteor 27 (1988), 785–796.
13. S. F. Burch and F. Ravenscroft, Computer modelling the UK Wind Energy Resource: Final Overview Report, Technical Report ETSU WN 7055, AEA Industrial Technology Harwell Laboratory, 1992.
14. A. Conte, A. Pavone, and C. F. Ratto, Numerical evaluation of the wind energy resource of Liguria, J Wind Eng Ind Aerodynam 74–76 (1998), 355–364.
15. R. Martens, H. Thielen, T. Sperling, and K. Maßmeyer, Validation of a mass consistent flow model as part of a decision support system, Int J Environ Pollut 14 (2000), 573–580.
16. Y. Wang, C. Williamson, D. Garvey, S. Chang, and J. Cogan, Application of a multigrid method to a mass–consistent diagnostic wind model, J Appl Meteorol 44 (2005), 1078–1089.
17. G. F. Homicz, Three–Dimensional Wind Field Modeling: A Review, report SAN2002–2597, Sandia National Laboratories, 2002.
18. M. A. Núñez, New scheme to compute mass–consistent models of geophysical flows, in “Proceedings of the I Workshop on Asymptotics for Parabolic and Hyperbolic Systems”, Laboratóiro Nacional de Computação Científica, Petrópolis-R. J., Brasil, 2008.

19. M. A. Núñez, G. Ramírez, and H. Hernández, Análisis de métodos de interpolación de datos de viento basados en la ecuación de conservación de la masa, Reporte de investigación, Universidad Autónoma Metropolitana, División de Ciencias Básicas e Ingeniería, 1–65, 2004.

20. M. A. Núñez, C. Flores, and H. Juárez, A study of hydrodynamic mass–consistent models, J Comp Methods Sci Eng 6 (2006), 365–385.

21. R. Glowinski, Numerical methods for fluids (Part 3), Handbook of numerical analysis, volume IX, North–Holland, Amsterdam, 2003.

22. V. Girault and P. A. Raviart, Finite element methods for the Navier–Stokes equations: theory and algorithms, Springer–Verlag, Berlin, 1986.

23. M. A. Núñez, C. Flores, and H. Juárez, Interpolation of hydrodynamic velocity data with the continuity equation, J Comp Methods Sci Eng 7 (2007), 21–42.

24. P. G. Ciarlet, The finite element method for elliptic problems, North–Holland, Amsterdam, 1970. Re–edited as Vol. 40, SIAM, Classics in Applied Mathematics, Philadelphia, PA., 2002.

25. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical recipies in Fortran 77, 2nd Ed., The Art of Scientific Computing, Cambridge University Press, USA, 1992.

26. M. Bercovier and Pironneau O., Error estimates for the finite element method solution of the Stokes problem in the primitive variables, Numer Math 33 (1979), 211–224.

27. GTOPO30 documentation, section 7, U. S. Geological Survey, Available at: http://www.scd.ucar.edu/dss/datasests/ds758.0hmtl (1997).

28. M. Raffel, C. E. Willert, and J. Kompenhans, Particle image velocimetry: a practical guide, Springer, Berlin, 1998.