An Adaptive Uzawa Algorithm for Output Functionals Control*

Stefano Micheletti, Simona Perotto and Marco Verani♯

3rd May 2005

♯ MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy
stefano.micheletti,simona.perotto,marco.verani@mate.polimi.it

Keywords:Galerkin finite element methods; Dual-based goal-oriented analysis; Uzawa algorithm; mesh adaption.

AMS Subject Classification: 65N12, 65N15, 65N50

Abstract

In this paper we address the approximation of a linear output functional $J(u)$, where u is the solution of an elliptic problem, through suitable duality arguments. In particular, moving from the adaptive Uzawa algorithm proposed by E. Bänsch et al. and S. Dahlke et al., we first cast the original problem in a saddle-point formulation and then we provide two new algorithms for computing an approximation u_h to u, such that $|J(u) - J(u_h)|$ be below a prescribed tolerance τ. Some test cases are included to assess the reliability of the proposed method in the 1D case together with some preliminary two dimensional results.

1 Introduction

It is well known that, when dealing with a partial differential problem, the computation of its solution requires a cost which gets higher and higher as the demand on accuracy, and the complexity of the problem increase. For a given accuracy, an efficient technique to minimize the computational cost is mesh

*This work has been supported by the project Cofin2003: "Numerical Modelling for Scientific Computing and Advanced Applications".

adaption driven by either heuristic criteria (for instance, the control of the gradient or the control of the Hessian of the solution) or by proper a priori or a posteriori error estimators. In the sequel, we focus on this last approach. An a posteriori error estimation typically considers a suitable energy norm induced by the underlying differential operator and is expressed in terms of computable local quantities (the residuals). However, this kind of estimate does not provide any information useful to control the error associated with physical quantities of real interest in applications, such as, e.g., the drag or the lift in aerodynamics or the wall shear stress in haemodynamics. To this end, using duality arguments, several procedures have been developed by R. Becker and R. Rannacher [2, 3], M.B. Giles and N.A. Pierce [9], M.B. Giles and E. Süli [10], Y. Maday and A.T. Patera [13], L. Machiels et al. [12], J.T. Oden and S. Prudhomme [16], M. Paraschivoiu et al. [17, 18], D.A. Venditti and D.L. Darmofal [19]. In more detail, a dual (adjoint) problem associated with the (primal) problem under examination is exploited to bound the error on the functional of interest in terms of the residual with respect to the primal problem and the solution of the adjoint problem.

In this paper we present an adaptive algorithm, that stems from an equivalent saddle-point formulation of the primal problem, aiming to approximate the goal quantity (together with the corresponding approximation of the primal problem) within a given tolerance. The exact solution u to the saddle-point problem and thus the exact value of the goal quantity $J(u)$ may be in principle obtained by an application of the Uzawa algorithm. Our approximation to $J(u)$ is computed as $J(u_h)$, where u_h is the approximate solution to the saddle-point problem, obtained via the adaptive Uzawa algorithm introduced in [1, 8].

Based on suitable approximations to the saddle-point problem, we introduce two new algorithms. The first one (**ago**) may be considered as the basic version: given an approximation z_H to the dual solution, the algorithm, after $\mathtt{m}$ iterations, computes an approximation $U_{\mathtt{m}}$ to the primal problem solution u, such that $|J(u) - J(U_{\mathtt{m}})| \leq \tau$, where τ is a user-defined accuracy. Notice that the number $\mathtt{m}$ of iterations can be determined a priori, as a function of the data only. However, this number can be sensitive to the choice of the approximation z_H. The second algorithm (`nago`), and actually the feasible one, improves on the first one in the following sense. In `ago` most of the work is supported by the approximation to the primal problem, as the dual problem is solved only once, in the beginning. Algorithm `nago` tries to equilibrate the work load between the primal and dual problems. To obtain this balance, a fractional step method is introduced, so that the tolerance τ is diluted through a user-defined number of iterations, each hinging on the `ago` algorithm.

The outline of the paper is as follows. In Sec. 2 we introduce the model problem, the duality framework and the basic idea of the algorithm proposed in this paper to estimate the goal quantity $J(u)$ within a precision τ. In Sec. 3 we show how to reformulate the primal problem as a constrained minimization problem, where the constraint is trivially satisfied by the primal solution u. This

constraint depends also on a given approximate solution, say z_H, of the dual problem. Then the equivalence between the primal problem and the constrained minimization one is proved after introducing a suitable saddle-point formulation of the minimization problem, for which the existence and uniqueness of the solution is proved, as well. In Sec. 4 we prove that to estimate the quantity $J(u)$ within the tolerance τ, it suffices to solve in an inexact way the saddle-point problem introduced in the previous section. With this aim we provide the basic version of our adaptive goal-oriented algorithm, named `ago`, discussing its main building blocks: an adaptive solver (`ads`) for the dual problem and an adaptive Uzawa algorithm (`aua`) to approximate the saddle-point problem. In particular we establish under what conditions the approximate solution to the saddle point, obtained via the `ago` algorithm, guarantees that the accuracy τ is met. Such an analysis provides us with an estimate for the maximum number of iterations, say `m`. In Sec. 5 we introduce and analyze a new algorithm, named `nago`, improving on `ago` in terms of flexibility, as it does not require any more a priori a fixed approximate solution z_H. The `nago` algorithm may be interpreted as a fractional step method, the primal and dual problem being solved successively in an iterative fashion until the accuracy τ on the goal-quantity $J(u)$ is reached. Finally, in Sec. 6 we present some numerical tests assessing the reliability of both the algorithms `ago` and `nago` on one-dimensional problems. Some promising 2D results are also provided.

2 The Model Problem

Let X be a Hilbert space endowed with norm $\|\cdot\|_X$ and with duality pair ${}_{X'}\!<\cdot,\cdot>_X$, X' denoting the dual space of X, that is the space of linear and continuous functionals defined on X, equipped with norm $\|\cdot\|_{X'}$. Let $A : X \to X'$ be a linear self-adjoint elliptic operator on X and let $f \in X'$. Let $u \in X$ be the solution of the differential problem

$$Au = f. \tag{1}$$

The bilinear form $a : X \times X \to \mathbb{R}$ defined by

$$a(u, v) = {}_{X'}\!< Au, v >_X, \qquad \forall\, u, v \in X,$$

can be associated with the operator A. Thus, the variational formulation of problem (1) reads as: find $u \in X$ such that

$$a(u, v) = {}_{X'}\!< f, v >_X, \qquad \forall\, v \in X. \tag{2}$$

We assume that the bilinear form $a(\cdot,\cdot)$ is continuous and coercive on $X \times X$, i.e., we have

$$|a(u, v)| \le C_A \|u\|_X \|v\|_X \quad \text{and} \quad a(u, u) \ge \alpha \|u\|_X^2, \qquad \forall\, u, v \in X, \tag{3}$$

with $C_A > 0$ and $\alpha > 0$ being the continuity and the coercivity constants, respectively. These two requirements guarantee the existence and the uniqueness of the solution u of the weak problem (2).

Let us introduce the adjoint differential operator A^* of A, such that $A^* : X \to X'$ is defined by the Lagrange identity [14]

$$_{X'}< A^*v, u >_X = \ _{X'}< Au, v >_X, \qquad \forall\ u, v \in X,$$

or, equivalently, by the relation

$$a^*(v, u) = a(u, v), \qquad \forall\ u, v \in X, \tag{4}$$

$a^*(\cdot, \cdot)$ being the bilinear form associated with the operator A^*, identified by the relation $a^*(v, u) = \ _{X'}< A^*v, u >_X$, $\forall u, v \in X$.

Let $J : X \to \mathbb{R}$ be a linear *goal functional* given by $J(v) = \ _{X'}< G, v >_X$, with $G \in X'$ its Riesz representant. Throughout, we are concerned with the approximation of the goal quantity $J(u)$, where u is the solution of (2). An equivalent way of computing $J(u)$ is to evaluate the quantity $_{X'} < f, z >_X$, where $z \in X$ satisfies the adjoint equation

$$A^*z = G, \tag{5}$$

or, likewise, the relation

$$a^*(z, v) = \ _{X'}< G, v >_X, \qquad \forall\ v \in X. \tag{6}$$

The equivalence stated above is an immediate consequence of the definition (4), as

$$_{X'}< f, z >_X = a(u, z) = a^*(z, u) = \ _{X'}< G, u >_X = J(u).$$

To approximate the goal quantity $J(u)$ within a given accuracy, we look for an approximation $u_h \in X$ of u such that

$$|J(e_h)| \leq \tau, \tag{7}$$

where $e_h = u - u_h$ is the approximation error, while τ denotes a user-prescribed tolerance. Thanks to the linearity of the functional J and to (2) and (6), it immediately follows that

$$J(e_h) = \ _{X'}< f, z >_X - a(u_h, z). \tag{8}$$

Let $z_H \in X$ be an approximation of the dual solution z and let $\{\phi_\lambda\}$ and $\{\varphi_\lambda\}$ two bases of suitable finite dimensional subspaces of X, such that

$$u_h = \sum_{\lambda \in \Lambda(u_h)} u_\lambda \phi_\lambda \quad \text{and} \quad z_H = \sum_{\lambda \in \Lambda(z_H)} z_\lambda \varphi_\lambda, \tag{9}$$

with $\#(\Lambda(u_h)) = N_h$, $\#(\Lambda(z_H)) = N_H$. Notice that this framework is quite general since it allows for different approximation spaces for the primal and dual problems. For example, in the case of finite elements, the discrete solutions u_h and z_H may be characterized by different degree and/or computed on different meshes.
Let us now introduce the residual operator $r_h \in X'$ defined as

$$_{X'}< r_h, v >_X = {}_{X'}< f, v >_X - a(u_h, v), \qquad \forall\, v \in X.$$

Since, from (8)

$$J(e_h) = {}_{X'}< r_h, z >_X = {}_{X'}< r_h, z - z_H >_X + {}_{X'}< r_h, z_H >_X,$$

via the Cauchy-Schwarz inequality it follows that

$$|J(e_h)| \leq \|r_h\|_{X'} \|z - z_H\|_X + |\, {}_{X'}< r_h, z_H >_X\, |. \tag{10}$$

The right-hand side of (10) is employed in Sec. 6 as an a posteriori error estimator for the quantity $J(e_h)$, provided that $\|z - z_H\|_X$ is suitably estimated in terms of computable quantities.

Remark 2.1 *We move from the same framework as in* [10]. *However, while in this latter paper it is suggested that the computable term* $_{X'}< r_h, z_H >_X$ *may be considered as a correction to the actual value of the approximation* $J(u_h)$, *in our approach it is part of the error bound for* $J(e_h)$. *Moreover, while the approach in* [10] *can be considered as a post-processing of the quantity* $J(u_h)$, *providing us with the improved value* $\widetilde{J}$ *for* $J(u_h)$, *it does not furnish an explicit expression for the approximation* $\widetilde{u}_h$ *corresponding to the corrected value* $\widetilde{J}$. *On the contrary, the analysis below guarantees that the approximate value for the functional* $J(u)$ *is always computed as* $J(\widetilde{u}_h)$, $\widetilde{u}_h$ *being now a computable quantity. However, in the numerical tests considered in Sec. 6 we show both types of approximations.*

To introduce the basic idea of the algorithm proposed in this paper, let us make the following

Assumption 2.1 *For a given tolerance* $\eta_H > 0$, *let the approximation* z_H *defined in* (9) *be such that*

$$\|z - z_H\|_X \leq \eta_H. \tag{11}$$

We remark that relation (11) is just an accuracy requirement.

Thus, in order to satisfy (7), thanks to (10), it suffices to find u_h such that

$$\|r_h\|_{X'} \leq \theta \frac{\tau}{\eta_H}, \tag{12}$$

$$|\, {}_{X'}< r_h, z_H >_X\, | \leq (1 - \theta)\tau. \tag{13}$$

Notice that, so far, $0 < \theta < 1$ is a user-defined parameter to be chosen according to an optimality criterion in Remark 4.4.

Remark 2.2 *In Sec. 5 we remove Assumption 2.1 while providing an alternative algorithm where the dual problem is iteratively solved within a variable tolerance via a suitable adaptive strategy.*

In the next section we show how to reformulate problem (2) as a constrained minimization problem, where the constraint is trivially satisfied by the primal solution u. This constraint depends on the approximate solution z_H of the dual problem (5). In Sec. 4, we will show that solving this minimization problem in an approximate manner guarantees that both relations (12) and (13) are satisfied.

3 The Constrained Minimization Problem

Let us first show that problem (2) is equivalent to solving *exactly* the following (trivial) constrained minimization problem:

$$\text{find} \quad \text{the} \quad \underset{w \in W}{\text{argmin}} \, \| \, _{X'} < f, \cdot >_X - a(w, \cdot) \|_{X'}, \tag{14}$$

where $W = \{w \in X : \ _{X'} < f, z_H >_X - a(w, z_H) = 0\}$ coincides with the space of functions for which the primal residual is equal to zero when tested against z_H. With this aim, we split the proof in two parts: *i*) we show that the solution u to (2) is also the unique solution to (14); *ii*) after introducing a suitable saddle-point formulation of the constrained minimization problem (14), we first prove the existence and the uniqueness of the solution (u, p) of this saddle-point problem and then we show that it coincides with the solution of (2), p being equal to 0.

Concerning point *i*), the statement follows trivially by observing that, $z_H \in X$. Now we deal with point *ii*). For this purpose we rewrite the constrained minimization problem (14) as a *saddle-point problem*. With this aim, let us introduce the energy functional

$$E(v) = \frac{1}{2} a(v, v) - \ _{X'} < f, v >_X, \qquad \forall \, v \in X.$$

and the Lagrangian multiplier q to enforce the constraint $_{X'} < f, z_H >_X - a(v, z_H) = 0$, so that the corresponding Lagrangian functional $\mathcal{L} : X \times \mathbb{R} \to \mathbb{R}$ is given by

$$\mathcal{L}(v, q) = E(v) + \Phi(v, q), \qquad \forall \, (v, q) \in X \times \mathbb{R}, \tag{15}$$

with $\Phi : X \times \mathbb{R} \to \mathbb{R}$ defined as

$$\Phi(v, q) = q \, (\, _{X'} < f, z_H >_X - a(v, z_H)), \qquad \forall \, (v, q) \in X \times \mathbb{R}.$$

We aim to minimize $\mathcal{L}$ over v and maximize it over q, i.e., we look for the saddle-point $(u, p) \in X \times \mathbb{R}$ of $\mathcal{L}$, such that

$$\mathcal{L}(u, q) \leq \mathcal{L}(u, p) \leq \mathcal{L}(v, p), \qquad \forall \, (v, q) \in X \times \mathbb{R}. \tag{16}$$

Let us now discuss the existence and the uniqueness of a saddle-point of the Lagrangian $\mathcal{L}$ defined in (15). To this end, we need to reformulate (16) as a system of variational equations. Let us introduce the operator $B : X \to \mathbb{R}$, such that $Bv = a(v, z_H), \forall v \in X$. Such an operator defines the bilinear form $b : X \times \mathbb{R} \to \mathbb{R}$ given by

$$b(v,q) = {}_{\mathbb{R}}< Bv, q >_{\mathbb{R}} = q\, Bv, \qquad \forall\ (v,q) \in X \times \mathbb{R}.$$

The adjoint operator $B^T : \mathbb{R} \to X'$ is identified by the relation ${}_{X'}< B^T q, v >_X = {}_{\mathbb{R}}< Bv, q >_{\mathbb{R}}, \forall v \in X$. Hence it follows that

$${}_{X'}< B^T q, v >_X = q\, a(v, z_H).$$

Letting $g = {}_{X'}< f, z_H >_X$, then problem (16) becomes equivalent to the following variational one: find $(u,p) \in X \times \mathbb{R}$, such that

$$\begin{aligned} a(u,v) + \quad b(v,p) &= {}_{X'}< f, v >_X, \qquad \forall\ v \in X, & (17)\\ b(u,q) &= {}_{\mathbb{R}}< g, q >_{\mathbb{R}}, \qquad \forall\ q \in \mathbb{R}. & (18) \end{aligned}$$

Thanks to the continuity of the bilinear form $a(\cdot,\cdot)$, it can be checked that $b(\cdot,\cdot)$ is continuous too, i.e.,

$$|b(v,q)| \le C_B\, \|v\|_X\, |q|, \qquad \forall\ (v,q) \in X \times \mathbb{R}, \tag{19}$$

with $C_B = C_A\, \|z_H\|_X$ and $|q|$ denoting the absolute value of the Lagrangian multiplier q. Moreover

$$a(v,v) \ge \alpha \|v\|_X^2, \qquad \forall\ v \in Ker(B) \subset X,$$

i.e., the bilinear form $a(\cdot,\cdot)$ is coercive on the kernel of B, while the form $b(\cdot,\cdot)$ satisfies the *inf-sup condition*

$$\inf_{q \in \mathbb{R},\, q \neq 0} \sup_{v \in X,\, v \neq 0} \frac{b(v,q)}{\|v\|_X |q|} \ge C_i^S, \tag{20}$$

with $C_i^S = \alpha \|z_H\|_X$, provided that $z_H \neq 0$. Indeed, taking $v = z_H/q$ and using the coercivity of $a(\cdot,\cdot)$, we get

$$\inf_{q \in \mathbb{R},\, q \neq 0} \sup_{v \in X,\, v \neq 0} \frac{b(v,q)}{\|v\|_X |q|} \ge \frac{a(z_H, z_H)}{\|z_H\|_X} \ge \alpha \|z_H\|_X.$$

Thanks to the properties (19)-(20), problem (17)-(18) admits a unique solution $(u,p) \in X \times \mathbb{R}$ (see [5]).

It is now possible to prove the last statement of point *ii*), that is:

Proposition 3.1 *If (u,p) is the unique saddle-point of the Lagrangian $\mathcal{L}$ defined in (15), then u is the solution of (2) and $p=0$.*

Proof 3.1 *If (u,p) is the unique saddle-point of the Lagrangian $\mathcal{L}$, then from (15)-(16), $\Phi(u,q) \leq \Phi(u,p)$, $\forall q \in \mathbb{R}$. In particular, taking $q=0$ and $q=2p$, we get*

$$\Phi(u,p) = 0. \tag{21}$$

On the other hand, if (u,p) is such a saddle-point, then from (16)

$$E(u) + \Phi(u,p) \leq E(v) + \Phi(v,p), \qquad \forall\, v \in W.$$

Thus, thanks to (21), the definition of the space W, and that $u = \mathrm{argmin}_{v\in X}\ E(v) = \mathrm{argmin}_{v\in W}\ E(v)$, it holds $E(u) \leq E(v)$, $\forall\ v \in X$. Hence u is the solution of (2). Moreover, from relations (17) and (2), it follows that $b(v,p)=0$, $\forall v \in X$, thus, thanks to condition (20), that $p=0$.

The desired equivalence between problems (2) and (14) is thus established.

Finally, to simplify the notations in the following, we rewrite problem (17)-(18) in block form: find $U=(u,p) \in X \times \mathbb{R}$ such that

$$\mathcal{A}U = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} = \mathcal{F}. \tag{22}$$

Thanks to (19)-(20), $\mathcal{A}$ is an isomorphism from $X \times \mathbb{R}$ into its dual $X' \times \mathbb{R}$, i.e., there exist two positive constants $c^1_{\mathcal{A}}, c^2_{\mathcal{A}}$ such that, $\forall U \in X \times \mathbb{R}$,

$$c^1_{\mathcal{A}}(\|u\|^2_X + |p|^2)^{1/2} \leq \|\mathcal{A}U\|_{X'\times\mathbb{R}} \leq c^2_{\mathcal{A}}(\|u\|^2_X + |p|^2)^{1/2}, \tag{23}$$

$\|\cdot\|_{X'\times\mathbb{R}}$ being the norm defined on the dual space $X' \times \mathbb{R}$.

4 An Adaptive Goal-oriented Algorithm: the Basic Scheme

In this section we come back to the aim of this paper, i.e., to the estimation of the goal quantity $J(u)$ within a prescribed tolerance τ. In more detail, we prove that to satisfy conditions (12) and (13) it suffices to solve *inexactly* the saddle-point problem (22), where the level of inexactness depends on τ itself. With this aim, let us first introduce a basic version of our adaptive goal-oriented algorithm, coupling an adaptive solver (`ads`) for the elliptic dual problem (5) together with an adaptive Uzawa algorithm (`aua`) to approximate the saddle-point problem (22).

According to Assumption 2.1, let η_H be a given tolerance and let z_H be the corresponding approximate solution of the dual problem (5). Thus the procedure `ads`, via an iterative algorithm, returns an approximate solution $[z_H] = \texttt{ads}(\eta_H)$

satisfying condition (11).
Once the approximation z_H is computed, the procedure `aua`, still through an iterative algorithm, provides us with an approximate solution $[u_h] = \texttt{aua}(z_H, \tau)$ of the saddle-point problem (22) verifying relations (12)-(13), and thus (7).
The following *adaptive goal-oriented* (`ago`) *algorithm* can thus be identified:

given η_H and τ, find z_H and u_h as

1. $[z_H] = \texttt{ads}(\eta_H)$;
2. $[u_h] = \texttt{aua}(z_H, \tau)$.

Remark 4.1 *In Sec. 5 we remove Assumption 2.1 while providing an alternative algorithm where the dual problem is iteratively solved within a variable tolerance via a suitable adaptive strategy.*

4.1 The `aua`

Aim of this section is to analyze in more detail the adaptive Uzawa algorithm `aua`. Concerning the adaptive solver `ads`, we refer to [4, 6, 7, 15].

For simplicity, let us first describe the *exact Uzawa algorithm*, that is an iterative procedure to solve the (dimensionally infinite) saddle-point problem (22). It reads as:

>> Input: $p_0 \in \mathbb{R}$, $\omega > 0$;

>> `for` $\mathtt{i} \geq 1$

1. Find $u_{\mathtt{i}} \in X$ such that $Au_{\mathtt{i}} = f - B^T p_{\mathtt{i}-1}$;
2. Update: $p_{\mathtt{i}} = p_{\mathtt{i}-1} + \omega(Bu_{\mathtt{i}} - g)$;

`end`

Let $S = BA^{-1}B^T$ be the Schur complement associated with the operator $\mathcal{A}$ in (22). Then it follows that $\|S\|_{\mathcal{L}(\mathbb{R};\mathbb{R})} = a(z_H, z_H)$ and, due to the continuity and the coercivity of the bilinear form $a(\cdot,\cdot)$, S is an isomorphism on $\mathbb{R}$, i.e.,

$$\alpha \, \|z_H\|_X^2 \, |q| \leq |Sq| \leq C_A \, \|z_H\|_X^2 \, |q|, \qquad \forall \, q \in \mathbb{R}. \tag{24}$$

We recall that with $\mathcal{L}(\mathbb{R};\mathbb{R})$ we mean the space of the real-valued linear and continuous functionals defined on $\mathbb{R}$, with corresponding norm $\|\cdot\|_{\mathcal{L}(\mathbb{R};\mathbb{R})}$. The iteration of the exact Uzawa algorithm can be written in terms of S as

$$p_i = (I - \omega S)\, p_{i-1} + \omega\, (BA^{-1}f - g), \tag{25}$$

where I is the identity operator. Therefore, if $0 < \omega < 2/\|S\|_{\mathcal{L}(\mathbb{R};\mathbb{R})}$, then

$$\beta = \|I - \omega S\|_{\mathcal{L}(\mathbb{R};\mathbb{R})} < 1, \tag{26}$$

i.e., the iterative method (25) is convergent and it turns out that

$$|p_i - p| \leq \beta^i \, |p_0 - p|.$$

Remark 4.2 *The quantity $\omega = 1/a(z_H, z_H)$ represents the optimal value for (25), because we are dealing with a scalar problem, since $p \in \mathbb{R}$. By exploiting the expression provided by system (22) for the exact value p together with Proposition 3.1, it is straightforward to verify that, if we choose $\omega = 1/a(z_H, z_H)$ in (25), then $p_1 = 0$, $\forall p_0 \in \mathbb{R}$, and $u_2 = u$. Thus, with this choice for ω, we get the exact values for both u and p after only two iterations. This is no longer true by considering an inexact version of the Uzawa scheme.*

Moving from the theory developed in [1] and [8], we consider the inexact Uzawa algorithm, called the adaptive Uzawa algorithm (`aua`). The main idea is to replace in the exact Uzawa algorithm the equation $Au_i = f - B^T p_{i-1}$ with a suitable approximation, to get rid of the dimensionally infinite quantities.

The `aua` consists of three main ingredients: *i)* the approximate primal solver (`aps`); *ii)* the compression algorithm (`comp`) and *iii)* the update procedure (`update`). Let us analyze separately these three steps.

i) Let ε_i be an error tolerance and let P_{i-1} be a tentative value for the Lagrange multiplier p. Then the procedure `aps`, identified formally by

$$[U_i] = \texttt{aps}(\varepsilon_i, f, P_{i-1}),$$

yields, via an adaptive algorithm, a dimensionally finite approximate solution U_i to the elliptic problem

$$Au_i = f - B^T P_{i-1}, \tag{27}$$

within the prescribed tolerance ε_i, that is we have,

$$\|U_i - u_i\|_X \leq \varepsilon_i,$$

in the worst case up to a constant C independent of i.

ii) Let γ_i be a second error tolerance. The compression step

$$[Q_i] = \texttt{comp}(\gamma_i, U_i)$$

returns a dimensionally finite approximation Q_i for the dimensionally infinite quantity BU_i, such that

$$|Q_i - BU_i| \leq \gamma_i,$$

the inequality being verified up to a constant independent of i.

iii) The Lagrange multiplier update is performed by the procedure

$$[P_i] = \texttt{update}(P_{i-1}, g, \omega, Q_i),$$

consisting of evaluating the new approximation P_i for the multiplier p as

$$P_i = P_{i-1} + \omega\,(Q_i - g). \tag{28}$$

The `aua` can thus be summarized as:

>> Input: $0 < \xi < 1,\ \varepsilon_0,\ \gamma_0 > 0,\ P_0 \in \mathbb{R},\ 0 < \omega < 2/\|S\|_{\mathcal{L}(\mathbb{R};\mathbb{R})}$;

>> `for i=1:m`

1. $\varepsilon_{\mathtt{i}} \leftarrow \xi\,\varepsilon_{\mathtt{i}-1}$;
2. $\gamma_{\mathtt{i}} \leftarrow \xi\,\gamma_{\mathtt{i}-1}$;
3. `Approximation` : $[U_{\mathtt{i}}] = \texttt{aps}(\varepsilon_{\mathtt{i}}, f, P_{\mathtt{i}-1})$;
4. `Compression` : $[Q_{\mathtt{i}}] = \texttt{comp}(\gamma_{\mathtt{i}}, U_{\mathtt{i}})$;
5. `Update` : $P_{\mathtt{i}} = \texttt{update}\ (P_{\mathtt{i}-1}, g, \omega, Q_{\mathtt{i}})$;
6. $\mathtt{i} \leftarrow \mathtt{i} + 1$

`end`

>> Output : $U_{\mathtt{m}}$.

Concerning the input parameters, notice that the quantity ξ is an error reduction factor governing the update of the error tolerances ε_i and γ_i. Moreover, we remark that no stopping criterion controls the `for` loop: it will be repeated exactly `m` times, `m` being an integer related with the fulfillment of the inequalities (12) and (13) and whose value will become clear in the next section. Finally, the output $U_{\mathtt{m}}$ represents the desired approximation of the solution u of problem (2) such that relation (7) is verified (we refer to Proposition 4.1 for the proof of this result).

Remark 4.3 *As for the* `comp` *step, note that as* $BU_i = a(U_i, z_H)$ *is actually an integral,* Q_i *is obtained thanks to a suitable quadrature rule. The main difficulty of such a step is related to the fact that* U_i *and* z_H *are a priori associated with two different computational grids, independently of the choice of the shape functions* ϕ_λ *and* φ_λ *in (9), which may also be different (see Sec. 6 and [10] for more details).*

We end this section by recalling the following result providing us with an error estimate for the `aua` procedure [1, 8].

Theorem 4.1 Let z_H be an approximate solution to the dual problem (5) and let U_i be an approximation to the solution u_i of the elliptic problem (27), such that

$$\|U_i - u_i\|_X \leq \varepsilon_0 \, \xi^i, \tag{29}$$

where $\varepsilon_0 > 0$ is a prescribed tolerance while $0 < \xi < 1$ is an assigned error reduction factor. Let Q_i be an approximation of the quantity BU_i such that

$$|Q_i - BU_i| \leq \varepsilon_0 \, \xi^i. \tag{30}$$

Then the inequalities

$$|p - P_i| \leq C_H \, \delta^i, \tag{31}$$

$$\|f - AU_i\|_{X'} \leq D_H \, \delta^i, \tag{32}$$

hold for some positive constants C_H and D_H depending on z_H only, and for a suitable parameter δ, with $0 < \xi < \delta < 1$.

For the reader's ease we provide the proof of this result in A.

4.2 Analysis of `ago`

Now we are in a position to state our main result: we prove that, given an approximation z_H for the solution z of the dual problem (5), `m aua` iterations suffice to get an approximation u_h for the solution u of the primal problem (2), such that (7) is fulfilled. In particular, `m` depends on the tolerance τ in (7) and on η_H in (11).

Proposition 4.1 *Let z_H be an approximation of the solution z of the dual problem* (5) *such that Assumption* 2.1 *is satisfied. Let τ be a user-defined tolerance. Then after* `m` *iterations of the* `aua`*, the approximation $U_{\mathtt{m}}$ is such that*

$$|J(u - U_{\mathtt{m}})| \leq \tau, \tag{33}$$

with

$$\mathtt{m} = \left\lceil \log_\delta \left(\frac{\tau}{C_A(C_H \, \|z_H\|_X + \varepsilon_0)} \, \min \left\{ \frac{(1-\theta)}{\|z_H\|_X}, \frac{\theta}{\eta_H} \right\} \right) + 1 \right\rceil,$$

and all the constants being defined as in Theorem 4.1.

Proof 4.1 *Thanks to relations (12)-(13) and Assumption 2.1, it suffices to prove the inequalities*

$$\|f - AU_{\mathtt{m}}\|_{X'} \leq \theta \frac{\tau}{\eta_H}, \tag{34}$$

$$|\, {}_{X'}\!< f - AU_{\mathtt{m}}, z_H >_X \,| \leq (1-\theta)\,\tau. \tag{35}$$

First notice that, due to relation $u = A^{-1}f - A^{-1}B^T p$ *and problem (22), we get*

$$BA^{-1}f - g = BA^{-1}f - Bu = BA^{-1}B^T p. \tag{36}$$

Thus, by exploiting the definition of the Schur complement, it follows that

$$\begin{aligned} Bu_{\mathtt{m}} - g &= BA^{-1}f - g - BA^{-1}B^T P_{\mathtt{m}-1} \\ &= BA^{-1}B^T(p - P_{\mathtt{m}-1}) = S(p - P_{\mathtt{m}-1}), \end{aligned} \tag{37}$$

$u_{\mathtt{m}}$ *denoting the exact solution of (27) for* $i = \mathtt{m}$.
Let us begin by proving inequality (35). Using the definitions of B *and* g*, relations (19), (37) and (24), together with the error estimate (29), for* $i = \mathtt{m}$*, and the estimate (31), we have*

$$\begin{aligned} |\,_{X'}\!< f - AU_{\mathtt{m}}, z_H >_X| &= |BU_{\mathtt{m}} - g| \\ &\leq |Bu_{\mathtt{m}} - g| + |B(U_{\mathtt{m}} - u_{\mathtt{m}})| \\ &\leq |S(p - P_{\mathtt{m}-1})| + C_B \, \|U_{\mathtt{m}} - u_{\mathtt{m}}\|_X \\ &\leq C_A \, \|z_H\|_X^2 \, |p - P_{\mathtt{m}-1}| + C_B \, \varepsilon_0 \, \xi^{\mathtt{m}} \\ &\leq C_A \, \|z_H\|_X^2 \, C_H \, \delta^{\mathtt{m}-1} + C_B \, \varepsilon_0 \, \delta^{\mathtt{m}-1}, \end{aligned}$$

the quantities ξ *and* δ *being defined according to Theorem 4.1, so that* $0 < \xi < \delta < 1$*. Now, in order to satisfy relation (35), it suffices to choose* $\mathtt{m}$ *such that*

$$(C_A \, C_H \, \|z_H\|_X^2 + C_B \, \varepsilon_0) \, \delta^{\mathtt{m}-1} \leq (1 - \theta) \, \tau,$$

i.e.,

$$\delta^{\mathtt{m}-1} \leq \frac{(1 - \theta) \, \tau}{C_A \, C_H \, \|z_H\|_X^2 + C_B \, \varepsilon_0}, \tag{38}$$

C_H *being explicitly defined in A.*
Now, let us prove inequality (34). Thanks to the error estimate (32) and the definition of D_H*, also provided in A, we have to guarantee that the quantity*

$$D_H \, \delta^{\mathtt{m}} = \left(\frac{C_B \, C_H}{\delta} + C_A \, \varepsilon_0 \right) \, \delta^{\mathtt{m}} \leq (C_B \, C_H + C_A \, \varepsilon_0) \, \delta^{\mathtt{m}-1}$$

be less than or equal to $\theta \, \tau / \eta_H$*, i.e., we have to choose* $\mathtt{m}$ *such that*

$$\delta^{\mathtt{m}-1} \leq \frac{\theta \, \tau}{(C_B \, C_H + C_A \, \varepsilon_0) \, \eta_H}. \tag{39}$$

Thus, in order to satisfy simultaneously relations (34) and (35), we have to combine inequalities (38) and (39), namely to choose the integer $\mathtt{m}$ *such that*

$$\delta^{\mathtt{m}-1} \leq \frac{\tau}{C_A(C_H \, \|z_H\|_X + \varepsilon_0)} \min \left\{ \frac{(1 - \theta)}{\|z_H\|_X}, \frac{\theta}{\eta_H} \right\},$$

that is

$$\mathtt{m} \geq \log_\delta \left(\frac{\tau}{C_A(C_H \, \|z_H\|_X + \varepsilon_0)} \min \left\{ \frac{(1 - \theta)}{\|z_H\|_X}, \frac{\theta}{\eta_H} \right\} \right) + 1 \, . \tag{40}$$

Remark 4.4 *So far no suggestion has been provided to choose the parameter θ characterizing both the relations* (12) *and* (13). *Moving from result* (40), *it is easy to verify that an optimal choice for θ, minimizing the number* `m` *of the* `aua` *iterations, is $\theta = \eta_H/(\|z_H\| + \eta_H)$. In such a case the integer* `m` *has to satisfy the relation*

$$\mathtt{m} \geq \log_\delta \left(\frac{\tau}{C_A(C_H \|z_H\|_X + \varepsilon_0)} \frac{1}{\|z_H\|_X + \eta_H} \right) + 1 \,. \tag{41}$$

Remark 4.5 *Relation (41) establishes a link between the target tolerance τ on the goal quantity $J(u)$ and the accuracy η_H for the approximation z_H of the solution of the dual problem* (5). *Thus, once the dual problem has been approximated, i.e., the quantities η_H and $\|z_H\|_X$ are known, the number* `m` *of the* `aua` *iterations guaranteeing relation (33) can be determined a priori.*

5 An Alternative ago Algorithm

In this section we generalize the basic scheme of Sec. 4 to design an improved adaptive goal-oriented algorithm, where no a priori choice is made for the tolerance η_H associated with the dual problem. This new algorithm turns out to be more versatile than the basic one `ago`, though in the numerical Sec. 6 we provide results obtained with both the algorithms.

Let us briefly explain the idea behind the new algorithm. We aim to meet the tolerance τ on the target quantity $J(u)$ via a fractional step-like method (see Fig. 1), after, say, $\mathtt{k}_{\max}$ iterations. For this purpose we introduce the intermediate tolerance $C_0\,\rho^{\mathtt{k}}$, where `k` is the iteration counter, C_0 a given constant and ρ a suitable error reduction factor. Thus $\mathtt{k}_{\max}$ will be determined by requiring that $C_0\,\rho^{\mathtt{k}_{\max}} = \tau$. In more detail, at each iteration, given the approximation $U_{\mathtt{k}-1}$ for the primal solution u with associated residual $r_h^{(\mathtt{k}-1)}$, let $z_H^{(\mathtt{k})}$ be the approximation to the dual solution z such that

$$\|z - z_H^{(\mathtt{k})}\|_X \leq \eta_H^{(\mathtt{k})},$$

with $\eta_H^{(\mathtt{k})} = C_0\rho^{\mathtt{k}}/\|r_h^{(\mathtt{k}-1)}\|_{X'}$. Then the `ago` algorithm is applied to compute the approximation $U_{\mathtt{k}}$ such that

$$\begin{aligned} \|r_h^{(\mathtt{k})}\|_{X'} &\leq \theta_{\mathtt{k}} \frac{C_0\rho^{\mathtt{k}}}{\eta_H^{(\mathtt{k})}}, \\ |\,_{X'}\!< r_h^{(\mathtt{k})}, z_H^{(\mathtt{k})} >_X\,| &\leq (1-\theta_{\mathtt{k}}) C_0\rho^{\mathtt{k}}, \end{aligned} \tag{42}$$

$0 < \theta_{\mathtt{k}} < 1$ being a dynamic acceleration parameter. Notice that, using the value $\eta_H^{(\mathtt{k})}$, relation (42) guarantees that

$$\|r_h^{(\mathtt{k})}\|_{X'} \leq \theta_{\mathtt{k}} \|r_h^{(\mathtt{k}-1)}\|_{X'},$$

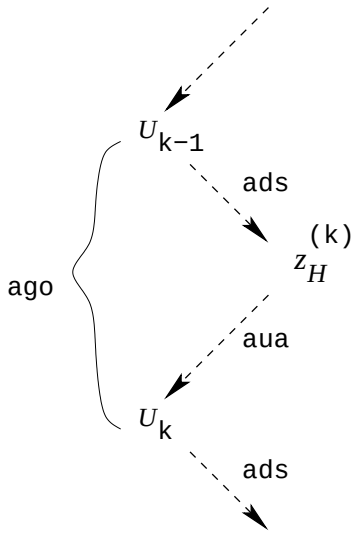

Figure 1: Schematic representation of the nago algorithm.

guaranteeing a strictly monotone reduction of the primal residual. Thus the *new adaptive goal-oriented* (nago) algorithm can be sketched as

>> Input: $\tau,\ 0 < \rho < 1,\ C_0 > 0,\ U_0$;

>> for $k = 1 : k_{max}$

1. $[z_H^{(k)}] = \mathtt{ads}\left(\dfrac{C_0\,\rho^k}{\|r_h^{(k-1)}\|_{X'}}\right)$;
2. $[U_k] = \mathtt{aua}\left(z_H^{(k)}, C_0\,\rho^k\right)$;

end

>> Output: $U_{k_{max}}$.

The number of the iterations necessary to guarantee (7) is given by $k_{max} = \lceil \log_\rho (\tau/C_0) \rceil$.

Thanks to Proposition 4.1, it is easy now to prove the following

Proposition 5.1 *Let $\tau > 0$ be a given tolerance and let $k_{max} = \lceil \log_\rho (\tau/C_0) \rceil$, with C_0 a given constant and ρ an error reduction factor. Then after k_{max} iterations, the approximation $U_{k_{max}}$ yielded by the* nago *algorithm is such that $|J(u - U_{k_{max}})| \leq \tau$.*

Proof 5.1 *At each iteration* $\mathtt{k}$*, since for computing* $U_{\mathtt{k}}$ *we use the* `ago` *algorithm, we can exploit Proposition 4.1 with*

$$z_H = z_H^{(\mathtt{k})} \quad \textit{and} \quad \eta_H = \frac{C_0\,\rho^{\mathtt{k}}}{\|r_h^{(\mathtt{k}-1)}\|_{X'}}\,.$$

This yields

$$\|z - z_H^{(\mathtt{k})}\|_X \; \|r_h^{(\mathtt{k})}\|_{X'} + |\,_{X'}\!< r_h^{(\mathtt{k})}, z_H^{(\mathtt{k})} >_X\,| \le C_0\,\rho^{\mathtt{k}},$$

that is the desired result after $\mathtt{k}_{\max}$ *iterations, as* $C_0\,\rho^{\mathtt{k}_{\max}} = \tau$.

The main advantage of the `nago` algorithm with respect to the basic `ago` version is that now we are relieved from choosing a priori the tolerance for the dual problem. Only the quantities τ, ρ, C_0 and U_0 have now to be provided by the user, where U_0 is just an initial guess for the primal solution, convergence being guaranteed independently of U_0, after $\mathtt{k}_{\max}$ iterations.

6 Numerical Validation

In the sequel, one- and two-dimensional test cases are considered in order to validate the algorithms `ago` and `nago` proposed in Sec. 4 and 5, respectively. Throughout, the differential operator A in (1) coincides with the Laplacian on the unit interval $(0,1)$, or unit square $(0,1)^2$, completed with homogeneous Dirichlet boundary conditions, while piecewise linear finite elements are used to approximate both the primal u and dual z solutions.

6.1 Implementation issues

Before delving into the numerical results, let us address some details concerning the implementation of the algorithms `ago` and `nago`, both in one and in two dimensions.

The numerical 1D code is entirely based on Matlab routines. For the procedures `aps` and `ads`, starting on a uniform mesh consisting of 100 elements, a mesh adaptive algorithm is employed for computing the trial solutions $U_{\mathtt{k}}$ and $z_H^{(\mathtt{k})}$, respectively. In particular, at each iteration, the algorithm computes the *optimal mesh density function* which, by suitably distributing the mesh nodes, minimizes the number of elements while at the same time satisfying the accuracy constraint on the numerical solution, as driven by the global procedure. This choice allows for both refining and coarsening of each mesh during computation.
As for the `comp` procedure, and for all the computations involving both the primal $\mathcal{T}_h$ and the dual $\mathcal{T}_H$ meshes, we extend the quantities involved to the union mesh $\mathcal{T}_h \cup \mathcal{T}_H$. Given the choice of the operator A and of the finite elements, this technique allows us to compute exactly the quantity BU_i in the `comp` procedure

of the `aua` algorithm, so that the tolerances γ_i are actually not necessary. The Matlab procedure `quadl`, implementing an adaptive Lobatto quadrature with tolerance 10^{-6}, is used to approximate all the integrals at hand.
The evaluation of the quantity $\|r_h\|_{X'}$ is approximated via the estimate $\|r_h\|_{X'} \simeq \|hf\|_{L^2(\Omega)}$, where $h = h(x)$ is the function describing the variation of the mesh size.

As far as the numerical tests in two dimensions are concerned, the software is based on the C++ code `FreeFem++` [11]. One of the key features of this software is the possibility of dealing simultaneously with different meshes and/or different degrees of finite elements easily. On the other hand it is not under control of the user what is actually being done in the interpolation process of one finite element solution onto another mesh.
In particular, we base both the algorithms `ads` and `aps` on the function `adaptmesh` with error control driven by an isotropic mesh adaption procedure, while for the `comp` algorithm a high-order Gaussian quadrature is employed, so that, also in this case, the tolerances γ_i are not necessary. To deal with the integrals involving quantities associated with different meshes, we resort to the following strategy. Both the primal and dual solutions are interpolated on a uniform triangular structured mesh consisting of 80000 elements, and the integrals are computed on this mesh via a high-order Gaussian quadrature.

Now, let us describe the numerical tests, starting with the one-dimensional validation.

6.2 One-dimensional test cases

Let us consider two test cases. For both of them it holds that the primal and dual solutions are the same, i.e., $u = z$. Concerning the data identifying the primal and dual problems (1) and (5), for the first test case we choose

- $f = G = c^2 \dfrac{e^{cx}}{e^c - 1}$,

with $c \gg 1$ and $x \in \Omega = (0, 1)$, with associated solutions given by

- $u = z = x - \dfrac{e^{cx} - 1}{e^c - 1}$,

and

- $J(v) = {}_{X'}\!< c^2 \dfrac{e^{cx}}{e^c - 1}, v >_X, \quad \forall v \in H^1(0, 1)$,

while, for the second test case, we have

- $f = G = \pi^2 \sin(\pi x)$,

that is

- $u = z = \sin(\pi x)$,

and

- $J(v) = {}_{X'}\!< \pi^2 \sin(\pi x), v >_X, \quad \forall v \in H^1(0,1)$.

Throughout, for the first test case, the value $c = 100$ is assumed, so that $J(u) = 49$, up to $\mathcal{O}(ce^{-c})$ terms that are neglected. Let us first consider the `ago` algorithm.

The `ago` *algorithm*
The following values of the parameters are set: $\tau = 10^{-2}, \varepsilon_0 = 1, \xi = 0.5, P_0 = 0$ and $\eta_H = 10^{-2}$. Tables 1-2 gather the results of the successive iterations of

i	$\lvert J(u - U_{\mathtt{i}})\rvert$	$\eta_{\mathtt{i}}$	$\#\mathcal{T}_h^{(\mathtt{i})}$	$\lvert J(u) - \widetilde{J}_{\mathtt{i}}\rvert$
1	$1.944 \cdot 10^{-2}$	$2.424 \cdot 10^{-2}$	48	$1.799 \cdot 10^{-5}$
2	$1.377 \cdot 10^{-2}$	$1.623 \cdot 10^{-2}$	85	$1.173 \cdot 10^{-5}$
3	$4.203 \cdot 10^{-3}$	$5.450 \cdot 10^{-3}$	158	$7.456 \cdot 10^{-6}$
4	$1.039 \cdot 10^{-3}$	$1.664 \cdot 10^{-3}$	305	$2.937 \cdot 10^{-6}$
5	$2.963 \cdot 10^{-4}$	$6.075 \cdot 10^{-4}$	599	$4.987 \cdot 10^{-7}$
6	$1.415 \cdot 10^{-4}$	$2.970 \cdot 10^{-4}$	1187	$4.855 \cdot 10^{-7}$
7	$5.518 \cdot 10^{-4}$	$6.303 \cdot 10^{-4}$	2362	$3.911 \cdot 10^{-7}$
8	$3.226 \cdot 10^{-4}$	$3.616 \cdot 10^{-4}$	4714	$1.696 \cdot 10^{-8}$
9	$2.162 \cdot 10^{-4}$	$2.357 \cdot 10^{-4}$	9417	$2.135 \cdot 10^{-8}$
10	$1.135 \cdot 10^{-4}$	$1.265 \cdot 10^{-4}$	18823	$2.709 \cdot 10^{-8}$
11	$6.289 \cdot 10^{-5}$	$6.936 \cdot 10^{-5}$	37635	$2.723 \cdot 10^{-8}$
12	$4.143 \cdot 10^{-5}$	$4.465 \cdot 10^{-5}$	75259	$2.640 \cdot 10^{-8}$

Table 1: First 1D test case: some relevant quantities for the `ago` algorithm.

the `ago` algorithm, corresponding to the different rows in the tables, for the first and second test cases, respectively. Notice that the meaning of the quantities through the columns, from left to right is: the current iteration `i` ($1 \leq \mathtt{i} \leq \mathtt{m}$), the error on the goal quantity $|J(u - U_{\mathtt{i}})|$, the a posteriori estimator defined as $\eta_{\mathtt{i}} = \|r_h^{(\mathtt{i})}\|_{X'} \eta_H + |\, {}_{X'}\!< r_h^{(\mathtt{i})}, z_H >_X |$, the cardinality of the meshes $\mathcal{T}_h^{(\mathtt{i})}$ for the primal problem, and the error on the target quantity $J(u)$ with respect to the corrected functional $\widetilde{J}_{\mathtt{i}} = J(U_{\mathtt{i}}) + {}_{X'}\!< r_h^{(\mathtt{i})}, z_H >_X$, $r_h^{(\mathtt{i})}$ being the residual associated with the approximation $U_{\mathtt{i}}$. Moving from Remark 2.1, in the last column we particularize the modified functional proposed in [10] to the case when different meshes are used to approximate the primal and dual solutions.
The cardinality $\#\mathcal{T}_H$ of the mesh associated with the dual problem is 1848 and 595, for the first and the second test case, respectively. Note that in both the test cases, the tolerance on the goal quantity is met at the third iteration, and this information is also captured by the estimator $\eta_{\mathtt{i}}$. Moreover the estimator turns out to be reliable and quite sharp throughout all the iterations, the values

i	$\lvert J(u-U_{\mathtt{i}})\rvert$	$\eta_{\mathtt{i}}$	$\#\mathcal{T}_h^{(\mathtt{i})}$	$\lvert J(u)-\widetilde{J}_{\mathtt{i}}\rvert$
1	$1.964\cdot 10^{-2}$	$2.448\cdot 10^{-2}$	13	$2.656\cdot 10^{-5}$
2	$1.181\cdot 10^{-2}$	$1.379\cdot 10^{-2}$	25	$4.414\cdot 10^{-4}$
3	$5.937\cdot 10^{-3}$	$7.223\cdot 10^{-3}$	48	$3.909\cdot 10^{-5}$
4	$1.803\cdot 10^{-3}$	$2.389\cdot 10^{-3}$	96	$3.531\cdot 10^{-5}$
5	$3.583\cdot 10^{-4}$	$6.616\cdot 10^{-4}$	191	$8.499\cdot 10^{-6}$
6	$1.170\cdot 10^{-5}$	$1.726\cdot 10^{-4}$	381	$4.925\cdot 10^{-6}$
7	$4.285\cdot 10^{-5}$	$1.218\cdot 10^{-4}$	762	$1.017\cdot 10^{-6}$
8	$4.236\cdot 10^{-5}$	$8.271\cdot 10^{-5}$	1523	$1.370\cdot 10^{-6}$
9	$2.236\cdot 10^{-5}$	$4.342\cdot 10^{-5}$	3044	$1.565\cdot 10^{-6}$
10	$1.088\cdot 10^{-5}$	$2.213\cdot 10^{-5}$	6088	$1.504\cdot 10^{-6}$
11	$4.956\cdot 10^{-6}$	$1.133\cdot 10^{-5}$	12175	$1.5028\cdot 10^{-6}$

Table 2: Second 1D test case: some relevant quantities for the ago algorithm.

predicted by $\eta_{\mathtt{i}}$ being always greater than but very close to the actual error. The behavior of the errors in the last column is remarkable: in particular in the first test case, a higher accuracy with respect to the corresponding values in the first column is shown. This agrees with the theory in [10].
The approximate solutions z_H and $U_{\mathtt{m}}$ and the mesh nodes (superimposed to the solutions) are shown in Fig. 2. Notice that, as a result of the adaptive procedure, for the first test case, the mesh spacing in both the primal and dual mesh, is smaller close to the sharp boundary layer of the solution. For the second test case the mesh nodes density increases around the maximum of the solutions u and z, where the second derivatives are larger. Moreover, for the sake of graphical clarity, in the figure on the bottom-left, the circles have been shown only one out of every 25.

The nago *algorithm*
Let us now consider the nago algorithm applied to the same test cases as above. The values of the parameters are chosen as $\tau = 10^{-2}$, $\rho = 0.5$, $C_0 = 1$, so that $\mathtt{k}_{\mathtt{max}} = 7$, while for the inner ago algorithm we let $\varepsilon_0 = 1, \xi = 0.5$, and $P_0 = 0$. Tables 3-4 collect the results corresponding to the successive iterations for the first and second test case, respectively. The meaning of the quantities through the columns, from left to right is, the current iteration $\mathtt{k}$ ($1 \le \mathtt{k} \le \mathtt{k}_{\mathtt{max}}$), the quantity $C_0\rho^{\mathtt{k}}$, the error on the goal quantity $|J(u-U_{\mathtt{k}})|$, the a posteriori estimator defined as $\eta_{\mathtt{k}} = \|r_h^{(\mathtt{k})}\|_{X'}\,\eta_H^{(\mathtt{k})} + |\,_{X'}< r_h^{(\mathtt{k})}, z_H^{(\mathtt{k})} >_X|$, the cardinality of the meshes $\mathcal{T}_h^{(\mathtt{k})}$ and $\mathcal{T}_H^{(\mathtt{k})}$ for the primal and dual problem, respectively, the tolerance $\eta_H^{(\mathtt{k})}$ for the approximate dual solution $z_H^{(\mathtt{k})}$, and finally, the error on the target quantity $J(u)$ with respect to the corrected functional $\widetilde{J}_{\mathtt{k}} = J(U_{\mathtt{k}}) + {}_{X'}< r_h^{(\mathtt{k})}, z_H^{(\mathtt{k})} >_X$, $r_h^{(\mathtt{k})}$ being the residual associated with the approximation $U_{\mathtt{k}}$. We observe that, in both tables, for any rows, the errors on the goal quantity in the

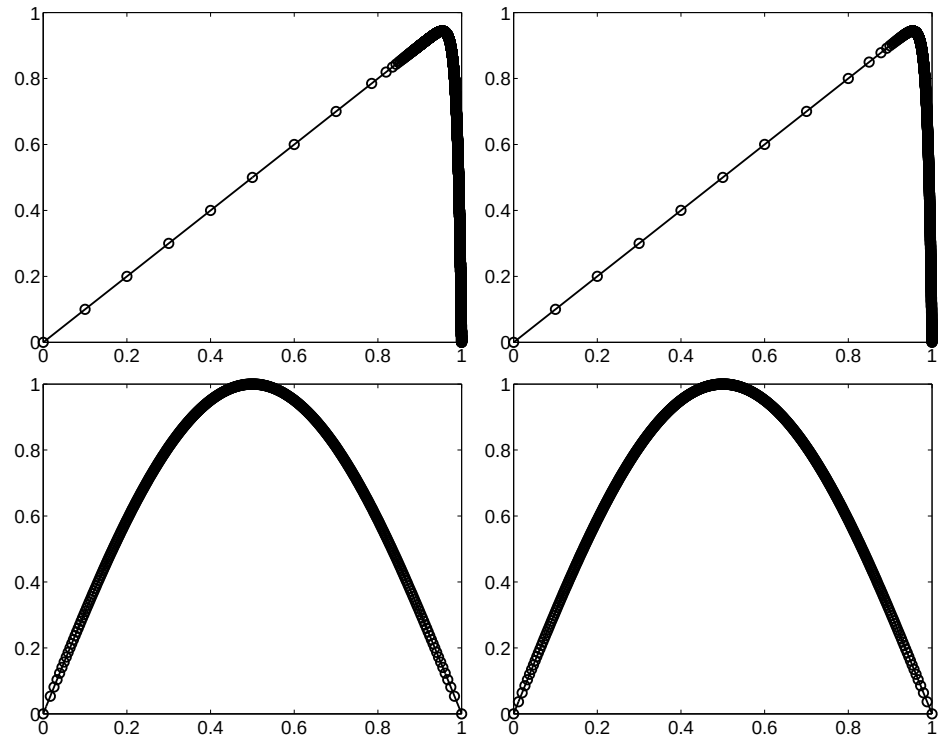

Figure 2: Computational meshes for the first (top) and second (bottom) 1D test case and for the primal (left) and dual (right) problem for the `ago` algorithm.

third column are always less than the corresponding tolerances in the second column, thus assessing the reliability of the whole procedure. The error with respect to the corrected functional $\widetilde{J}_k$ behaves similarly to the `ago` algorithm, showing again a higher accuracy, compared to the corresponding values in the third column. We point out the the cardinalities of the dual meshes, shown in the sixth column, are virtually constant through the iterations, and smaller than the corresponding values for the `ago` algorithm (1848 and 595 in the two test cases, respectively). This is a consequence of the more flexibility of the `nago` algorithm. Moreover, also the meshes for the primal problem show a far less growing rate than in the `ago` algorithm case, and even a higher accuracy is obtained. The approximate solutions and the mesh nodes, superimposed to the solution are shown in Fig. 3. Analogously to the `ago` algorithm, as a result of the adaptive procedure, the mesh nodes crowd in correspondence with the sharp boundary layer of the primal/dual solution in the first test case, and around the maximum of the solution in the second test case. For the sake of clarity, in the figure on the bottom-left, the circles have been shown only one out of every 26.

k	$C_0\rho^{\mathtt{k}}$	$\lvert J(u-U_{\mathtt{k}})\rvert$	$\eta_{\mathtt{k}}$	$\#\mathcal{T}_h^{(\mathtt{k})}$	$\#\mathcal{T}_H^{(\mathtt{k})}$	$\eta_H^{(\mathtt{k})}$	$\lvert J(u)-\widetilde{J}_{\mathtt{k}}\rvert$
1	$5.000\cdot 10^{-1}$	$2.001\cdot 10^{-2}$	$5.423\cdot 10^{-2}$	47	274	$7.000\cdot 10^{-2}$	$3.543\cdot 10^{-5}$
2	$2.500\cdot 10^{-1}$	$1.217\cdot 10^{-2}$	$1.378\cdot 10^{-1}$	85	47	$5.108\cdot 10^{-1}$	$7.174\cdot 10^{-4}$
3	$1.250\cdot 10^{-1}$	$4.321\cdot 10^{-3}$	$6.740\cdot 10^{-2}$	158	47	$5.111\cdot 10^{-1}$	$2.928\cdot 10^{-4}$
4	$6.250\cdot 10^{-2}$	$9.921\cdot 10^{-4}$	$3.237\cdot 10^{-2}$	305	48	$5.040\cdot 10^{-1}$	$2.024\cdot 10^{-5}$
5	$3.125\cdot 10^{-2}$	$2.866\cdot 10^{-4}$	$1.595\cdot 10^{-2}$	599	48	$5.022\cdot 10^{-1}$	$1.269\cdot 10^{-5}$
6	$1.562\cdot 10^{-2}$	$1.426\cdot 10^{-4}$	$7.959\cdot 10^{-3}$	1187	48	$5.012\cdot 10^{-1}$	$4.734\cdot 10^{-6}$
7	$7.812\cdot 10^{-3}$	$5.539\cdot 10^{-4}$	$4.465\cdot 10^{-3}$	2362	48	$5.006\cdot 10^{-1}$	$1.700\cdot 10^{-6}$

Table 3: First 1D test case: some relevant quantities for the `nago` algorithm.

k	$C_0\rho^{\mathtt{k}}$	$\lvert J(u-U_{\mathtt{k}})\rvert$	$\eta_{\mathtt{k}}$	$\#\mathcal{T}_h^{(\mathtt{k})}$	$\#\mathcal{T}_H^{(\mathtt{k})}$	$\eta_H^{(\mathtt{k})}$	$\lvert J(u)-\widetilde{J}_{\mathtt{k}}\rvert$
1	$5.000\cdot 10^{-1}$	$1.455\cdot 10^{-3}$	$4.421\cdot 10^{-1}$	96	10	$7.092\cdot 10^{0}$	$6.391\cdot 10^{-6}$
2	$2.500\cdot 10^{-1}$	$2.788\cdot 10^{-4}$	$1.257\cdot 10^{-1}$	191	10	$4.023\cdot 10^{0}$	$3.277\cdot 10^{-6}$
3	$1.250\cdot 10^{-1}$	$5.807\cdot 10^{-6}$	$6.257\cdot 10^{-2}$	381	10	$4.009\cdot 10^{0}$	$2.422\cdot 10^{-5}$
4	$6.250\cdot 10^{-2}$	$2.330\cdot 10^{-5}$	$3.125\cdot 10^{-2}$	762	10	$4.005\cdot 10^{0}$	$2.343\cdot 10^{-5}$
5	$3.125\cdot 10^{-2}$	$1.947\cdot 10^{-5}$	$1.567\cdot 10^{-2}$	1523	10	$4.010\cdot 10^{0}$	$2.332\cdot 10^{-5}$
6	$1.562\cdot 10^{-2}$	$1.326\cdot 10^{-8}$	$7.839\cdot 10^{-3}$	3044	10	$4.009\cdot 10^{0}$	$2.406\cdot 10^{-5}$
7	$7.812\cdot 10^{-3}$	$1.189\cdot 10^{-5}$	$3.918\cdot 10^{-3}$	6088	10	$4.007\cdot 10^{0}$	$2.385\cdot 10^{-5}$

Table 4: Second 1D test case: some relevant quantities for the `nago` algorithm.

6.3 Two-dimensional test cases

The main overhead characterizing the 2D numerical validation is due to the necessity of merging the information coming from two different computational grids at the `comp` step of the `aua` algorithm. This undoubtedly yields extra computational effort while introducing additional approximation errors.
Let us show in the sequel some preliminary results related to two different test cases. The operator A in (1) is identified with the Laplacian on the unit square, while piecewise linear finite elements are used to approximate both the primal u and dual z solutions.
For the first test case, the exact solution is $u = \sin(\pi x)\sin(\pi y)$, so that $f = 2\pi^2 u$, while the dual solution is given by

- $z = 10\,\exp(-10\,x)\,u$,

corresponding to

- $G = 20\,\exp(-10\,x)\,\sin(\pi\,y)\,[(\pi^2 - 50)\,\sin(\pi\,x) + 10\,\pi\,\cos(\pi\,x)]$,

- $J = {}_{X'}\!< G, u >_X = -\dfrac{\pi^4}{2}\dfrac{\exp(-10)-1}{25+\pi^2}$.

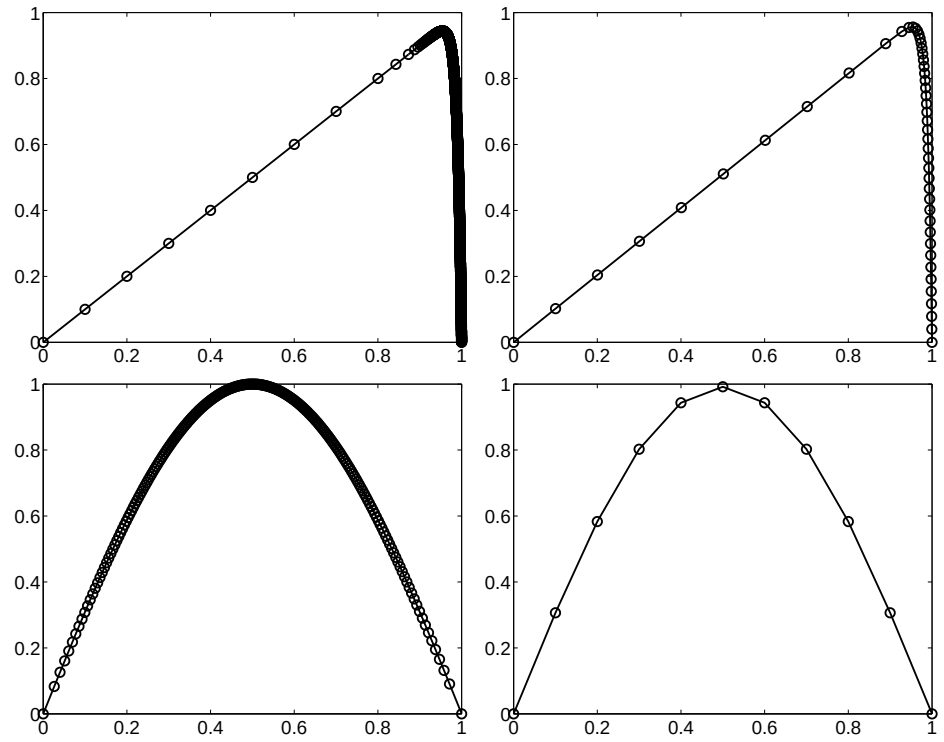

Figure 3: Computational meshes for the first (top) and second (bottom) 1D test case and for the primal (left) and dual (right) problem for the `nago` algorithm.

Concerning the second test case, the following choices are made for the primal and dual solutions

- $u = 4 \exp\Big(- \dfrac{(x-0.5)^2}{10^{-2}} \Big) y\,(1-y)$,
- $z = 10 \exp\Big(- \dfrac{(x-0.5)^2 + (y-0.1)^2}{10^{-4}} \Big)$,

namely, the forcing terms of the primal and dual problems and the target quantity are given by

- $f = 800 \exp(-25\,(2\,x-1)^2)\,[49\,y\,(y-1) + 200\,x\,y\,(1-y)\,(1-x)]$,
- $G = 4 \cdot 10^5\,[1 - 10^4\,((x-0.5)^2 + (y-0.1)^2)] \exp\Big(- \dfrac{(x-0.5)^2 + (y-0.1)^2}{10^{-4}} \Big)$,
- $J = {}_{X'}\!< G, u >_X = 2.477 \cdot 10^{-1}$,

respectively. Let us move from the basic ago algorithm.

The ago *algorithm*
For the first test case, the parameters ξ and ε_0 of the ago algorithm are both set equal to 0.5, while the tolerances η_H and τ are chosen equal to 10^{-1} and 10^{-3}, respectively. Moving from the value $P_0 = 1$, we get the results collected in Table 5. The number of actual iterations is 4, while the cardinality $\#\mathcal{T}_H$ of the dual mesh is 49977. The meaning of the quantities in the five columns is the same as in the one-dimensional case. The algorithm turns out to be

i	$\lvert J(u - U_{\mathtt{i}})\rvert$	$\eta_{\mathtt{i}}$	$\#\mathcal{T}_h^{(\mathtt{i})}$	$\lvert J(u) - \widetilde{J}_{\mathtt{i}}\rvert$
1	$1.331 \cdot 10^{+1}$	$1.328 \cdot 10^{+1}$	1738	$4.396 \cdot 10^{-2}$
2	$1.493 \cdot 10^{-1}$	$1.565 \cdot 10^{-1}$	1281	$2.258 \cdot 10^{-3}$
3	$1.458 \cdot 10^{-2}$	$1.915 \cdot 10^{-2}$	5242	$1.144 \cdot 10^{-4}$
4	$3.986 \cdot 10^{-4}$	$2.619 \cdot 10^{-3}$	21636	$5.806 \cdot 10^{-5}$

Table 5: First 2D test case: some relevant quantities for the ago algorithm.

reliable, as the tolerance on the target functional is met. Moreover, the corrected functional $\widetilde{J}_{\mathtt{i}}$ turns out to be more accurate, though not as much as in the one-dimensional case. This is probably related to the difficulty cited above of accurately computing the term $a(U_{\mathtt{i}}, z_H)$ involving the primal and dual meshes simultaneously.

We show in Fig. 4 the four meshes yielded by the ago algorithm (top-bottom left-right). Notice that, until the value P_i is relatively large, the i-th mesh takes both the primal and dual solutions into account and, in particular, this is evident in the first mesh. On the other hand, when P_i gets smaller and tends to the exact value $p = 0$, the contribution of the dual solution is gradually lost.

As for the second test case, the choices $\xi = 0.8$, $\varepsilon_0 = 0.5$, $\eta_H = 1$ and $\tau = 10^{-2}$ are made for the ago parameters and tolerances, respectively, while the value $P_0 = 1$ is adopted as initial value for the approximate Lagrange multiplier.
Table 6 gathers the results associated with these values. The number of actual iterations is now 10 while the cardinality $\#\mathcal{T}_H$ of the dual mesh is 21742. Remarks analogous to the ones made for the previous test case hold also for Table 6, both about the reliability of the proposed algorithm, and the behaviour of the corrected functional $\widetilde{J}_{\mathtt{i}}$. Notice that the estimator $\eta_{\mathtt{i}}$ is almost always an upper bound for the true error $\lvert J(u - U_{\mathtt{i}})\rvert$. The slight discrepancies are possibly due to the tunable constant pertaining the residual estimator by which the quantity $\|r_h^{(\mathtt{i})}\|_{X'}$ is approximated.
Finally, in Fig. 5 four out of ten meshes yielded by the ago algorithm are furnished. The influence of the dual problem can be still appreciably detected until the iteration $\mathtt{i} = 5$ (that is for values of the approximate Lagrange multiplier P_i

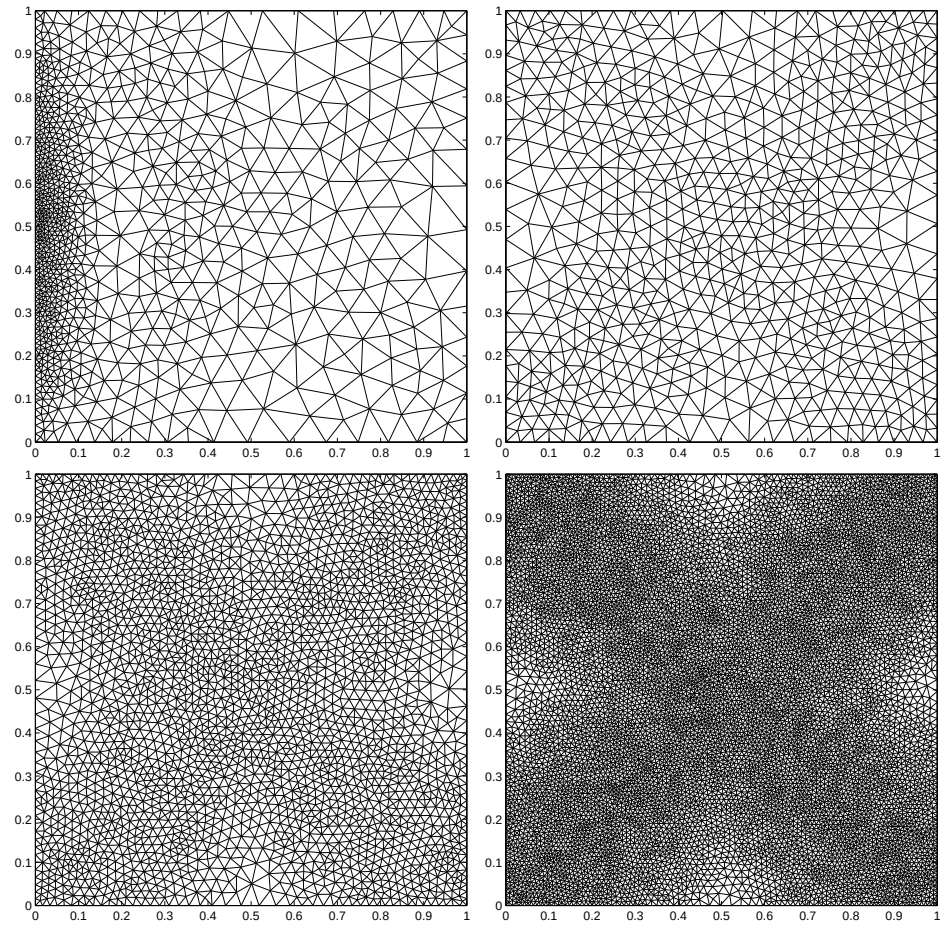

Figure 4: First 2D test case: the four computational meshes yielded by the `ago` algorithm.

not too small), while it is completely missed in the further iterations.

The `nago` *algorithm*

Let us now consider the results on the second test case obtained via the `nago` algorithm. The parameters are the same as for the `ago` case, except for the initial value adopted for the approximate Lagrange multiplier now chosen as $P_0 = 0$. Finally, both the values C_0 and ρ are set equal to 0.5, and thus $\mathtt{k_{max}} = 6$.

Tables 7-8 gather the main quantities provided for this choice of parameters from the `ago` and `nago` algorithms, respectively. Note that the `nago` algorithm is reliable and that the final computational mesh is coarser than the corresponding final mesh for the `ago` case, though this save is obtained at the expenses of a much finer dual mesh. However, this test case represents an extreme situation where the dual solution is nonsmooth, thus the cardinality of the associated mesh turns out to be very sensitive even to small variations of the tolerance η_H.

i	$\lvert J(u-U_{\mathtt{i}})\rvert$	$\eta_{\mathtt{i}}$	$\#\mathcal{T}_h^{(\mathtt{i})}$	$\lvert J(u)-\widetilde{J}_{\mathtt{i}}\rvert$
1	$3.135\cdot 10^{+2}$	$2.648\cdot 10^{+2}$	16483	$4.880\cdot 10^{+1}$
2	$4.892\cdot 10^{+1}$	$4.222\cdot 10^{+1}$	299	$7.607\cdot 10^{+0}$
3	$7.307\cdot 10^{+0}$	$6.341\cdot 10^{+0}$	2373	$1.156\cdot 10^{+0}$
4	$7.597\cdot 10^{-1}$	$7.807\cdot 10^{-1}$	3344	$1.302\cdot 10^{-1}$
5	$4.796\cdot 10^{-1}$	$6.958\cdot 10^{-1}$	5034	$9.424\cdot 10^{-2}$
6	$1.934\cdot 10^{-1}$	$2.293\cdot 10^{-1}$	7550	$3.239\cdot 10^{-1}$
7	$2.272\cdot 10^{-1}$	$1.661\cdot 10^{-1}$	11259	$3.132\cdot 10^{-1}$
8	$2.357\cdot 10^{-2}$	$1.021\cdot 10^{-1}$	20494	$1.942\cdot 10^{-2}$
9	$1.610\cdot 10^{-2}$	$6.409\cdot 10^{-2}$	30838	$1.520\cdot 10^{-4}$
10	$2.360\cdot 10^{-3}$	$4.277\cdot 10^{-2}$	46278	$1.189\cdot 10^{-3}$

Table 6: Second 2D test case: some relevant quantities for the ago algorithm.

i	$\lvert J(u-U_{\mathtt{i}})\rvert$	$\eta_{\mathtt{i}}$	$\#\mathcal{T}_h^{(\mathtt{i})}$	$\lvert J(u)-\widetilde{J}_{\mathtt{i}}\rvert$
1	$5.714\cdot 10^{0}$	$3.077\cdot 10^{-1}$	950	$5.725\cdot 10^{0}$
2	$1.056\cdot 10^{1}$	$2.778\cdot 10^{-1}$	1440	$1.052\cdot 10^{1}$
3	$8.809\cdot 10^{0}$	$2.541\cdot 10^{-1}$	2213	$8.875\cdot 10^{0}$
4	$1.068\cdot 10^{0}$	$1.788\cdot 10^{-1}$	3334	$1.096\cdot 10^{0}$
5	$1.625\cdot 10^{-1}$	$1.269\cdot 10^{-1}$	4996	$1.585\cdot 10^{-1}$
6	$1.976\cdot 10^{-2}$	$1.228\cdot 10^{-1}$	7588	$4.390\cdot 10^{-2}$
7	$2.001\cdot 10^{-2}$	$9.372\cdot 10^{-2}$	11457	$6.232\cdot 10^{-3}$
8	$2.079\cdot 10^{-2}$	$6.896\cdot 10^{-2}$	20534	$3.065\cdot 10^{-2}$
9	$9.505\cdot 10^{-4}$	$5.299\cdot 10^{-2}$	30818	$3.900\cdot 10^{-3}$

Table 7: Second 2D test case: some relevant quantities for the ago algorithm.

Finally Fig. 6 compares the final grid provided by the two algorithms: note how the mesh associated with the nago algorithm, though consisting of about the two thirds of the elements of the mesh corresponding to the ago algorithm, provides an approximation for the target quantity of like accuracy. As a final observation, we compare our approach against the one based on the following pure "primal" argument. From the definition of the target quantity, we have

$$|J(U-u_h)| = |\ _{X'}< G, u-u_h >_X\ | \leq \|G\|_{X'}\|u-u_h\|_X,$$

from which it follows that, in order to have $|J(U-u_h)| \leq \tau$, it suffices to require that $\|u-u_h\|_X \leq \dfrac{\tau}{\|G\|_{X'}}$. Thus, to compute the approximation u_h, one could employ a primal solver with a tolerance equal to $\dfrac{\tau}{\|G\|_{X'}}$. However, when the term G is rough, i.e., its norm is large, this tolerance may become prohibitively small. For example, for the two 2D test cases, it holds $\|G\|_{X'}$ is approximately

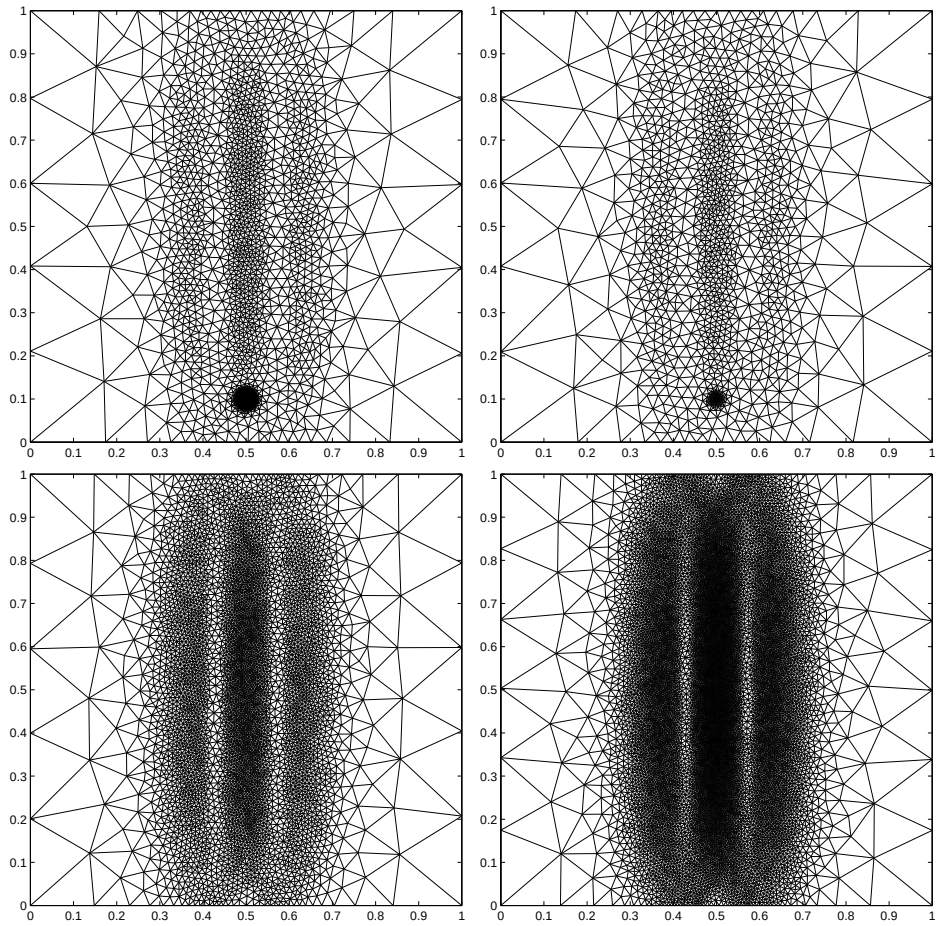

Figure 5: Second 2D test case: four computational meshes yielded by the `ago` algorithm. $\mathcal{T}_h^{(1)}$ (top-left), $\mathcal{T}_h^{(3)}$ (top-right), $\mathcal{T}_h^{(7)}$(bottom-left) and $\mathcal{T}_h^{(9)}$ (bottom-right).

equal to 80 and 3500, respectively. This yields the tolerances $\dfrac{10^{-3}}{80} = 1.25 \cdot 10^{-5}$ and $\dfrac{10^{-2}}{3500} \simeq 2.85 \cdot 10^{-6}$ in the first and second case, respectively. These values are some order of magnitude smaller than the tolerances employed for solving the primal problem in both the `ago` and `nago` algorithms, and this would likely produce a much larger number of elements.

k	$C_0\rho^{\mathtt{k}}$	$\lvert J(u-U_{\mathtt{k}})\rvert$	$\eta_{\mathtt{k}}$	$\#\mathcal{T}_h^{(\mathtt{k})}$	$\#\mathcal{T}_H^{(\mathtt{k})}$	$\eta_H^{(\mathtt{k})}$	$\lvert J(u)-\widetilde{J}_{\mathtt{k}}\rvert$
1	$2.500\cdot10^{-1}$	$9.425\cdot10^{-2}$	$8.528\cdot10^{-2}$	3324	35850	$5.106\cdot10^{-1}$	$8.636\cdot10^{-2}$
2	$1.250\cdot10^{-1}$	$1.216\cdot10^{-1}$	$1.121\cdot10^{-1}$	4982	158	$8.247\cdot10^{-1}$	$1.102\cdot10^{-1}$
3	$6.250\cdot10^{-2}$	$4.900\cdot10^{-2}$	$7.573\cdot10^{-2}$	5048	38720	$5.116\cdot10^{-1}$	$3.580\cdot10^{-2}$
4	$3.125\cdot10^{-2}$	$2.385\cdot10^{-3}$	$1.783\cdot10^{-2}$	17225	237792	$2.556\cdot10^{-1}$	$1.075\cdot10^{-3}$
5	$1.562\cdot10^{-2}$	$8.878\cdot10^{-3}$	$1.325\cdot10^{-2}$	30732	287	$2.417\cdot10^{-1}$	$7.279\cdot10^{-3}$
6	$7.812\cdot10^{-3}$	$1.020\cdot10^{-3}$	$1.520\cdot10^{-2}$	20532	373792	$1.619\cdot10^{-1}$	$4.585\cdot10^{-3}$

Table 8: Second 2D test case: some relevant quantities for the `nago` algorithm.

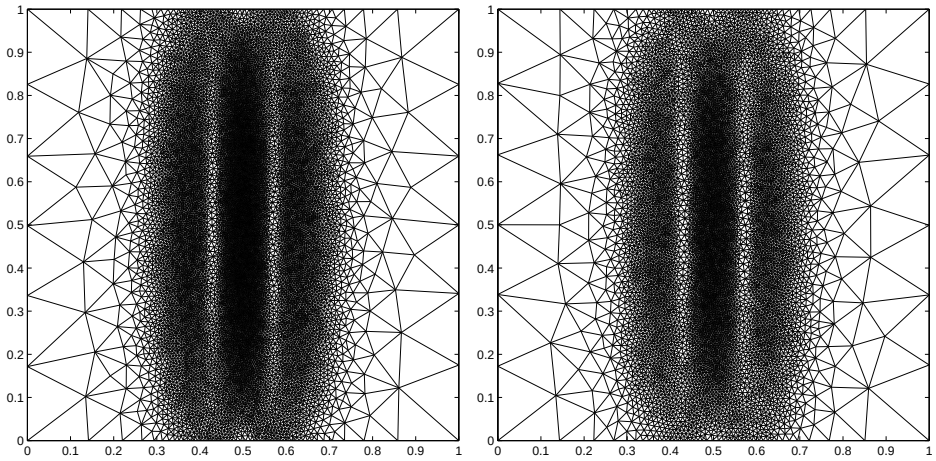

Figure 6: Second 2D test case: computational mesh provided by the `ago` (left) and `nago` (right) algorithms.

A Proof of Theorem 4.1

We start by proving (31). From equalities (28) and (27) and from the definition of the Schur complement S, we have

$$\begin{aligned} P_i &= P_{i-1} + \omega\,(Q_i - g) \\ &= (I - \omega BA^{-1}B^T)P_{i-1} + \omega BA^{-1}B^T P_{i-1} + \omega\,(BU_i - g) + \omega\,(Q_i - BU_i) \\ &= (I - \omega S)P_{i-1} + \omega\,(BA^{-1}f - Bu_i) + \omega\,(BU_i - g) + \omega\,(Q_i - BU_i) \\ &= (I - \omega S)P_{i-1} + \omega\,(BU_i - Bu_i) + \omega\,BA^{-1}f - \omega\,g + \omega\,(Q_i - BU_i). \end{aligned}$$

As $p = (I - \omega\,S)\,p + \omega\,(BA^{-1}f - g)$, it follows that

$$p - P_i = (I - \omega\,S)(p - P_{i-1}) + \omega\,(Bu_i - BU_i) + \omega\,(BU_i - Q_i),$$

that is

$$\begin{aligned} |p - P_i| &\leq \beta\,|p - P_{i-1}| + \omega\,|Bu_i - BU_i| + \omega\,|BU_i - Q_i| \\ &\leq \beta\,|p - P_{i-1}| + \omega\,C_B\,\|U_i - u_i\|_X + \omega\,|BU_i - Q_i| \\ &\leq \beta\,|p - P_{i-1}| + \omega\,(1 + C_B)\,\varepsilon_0\,\xi^i, \end{aligned}$$

where relations (19) and (26) together with assumptions (29) and (30) have been exploited. By recursion, we obtain

$$|p - P_i| \leq \beta^i\,|p - P_0| + \omega\,(1 + C_B)\,\varepsilon_0 \sum_{l=0}^{i-1} \beta^l\,\xi^{i-l}. \tag{43}$$

Taking $\eta = \max\{\beta, \xi\}$ and recalling Proposition 3.1 yields

$$|p - P_i| \leq \eta^i\,|p - P_0| + \omega\,(1 + C_B)\,\varepsilon_0\,i\,\eta^i \leq (|P_0| + \omega\,(1 + C_B)\,\varepsilon_0)\,\delta^i, \tag{44}$$

for a suitable $\delta = \delta(\eta)$, with $0 < \eta < \delta < 1$. Hence (31) holds, with $C_H = |P_0| + \omega\,(1 + C_B)\,\varepsilon_0$.

Let us now prove inequality (32). Using the equality $A(u_i - u) = B^T(p - P_{i-1})$ and thanks to inequalities (19) and (3), we get

$$\begin{aligned} \|f - AU_i\|_{X'} &= \|A(u - U_i)\|_{X'} \\ &\leq \|A(u - u_i)\|_{X'} + \|A(u_i - U_i)\|_{X'} \\ &\leq \|B^T(p - P_{i-1})\|_{X'} + \|A(u_i - U_i)\|_{X'} \\ &\leq C_B\,|p - P_{i-1}| + C_A\,\|u_i - U_i\|_X. \end{aligned}$$

From (43) and assumption (29), it follows

$$\|f - AU_i\|_{X'} \leq C_B \left(\beta^{i-1}\,|p - P_0| + \omega\,(1 + C_B)\,\varepsilon_0 \sum_{l=0}^{i-2} \beta^l\,\xi^{i-l} \right) + C_A\,\varepsilon_0\,\xi^i.$$

Now by using inequality (44), we get

$$\|f - AU_i\|_{X'} \leq C_B\,C_H\,\delta^{i-1} + C_A\,\varepsilon_0\,\xi^i \leq \left(\frac{C_B\,C_H}{\delta} + C_A\,\varepsilon_0 \right) \delta^i,$$

that is result (32) with $D_H = \dfrac{C_B\,C_H}{\delta} + C_A\,\varepsilon_0$. □

Conclusions

In this paper we have addressed the approximation of a linear output functional $J(u)$, typically representing physical meaningful quantities, associated with the

(primal) problem at hand, through a suitable duality framework. For this purpose, we first rewrite the primal problem as a saddle-point formulation depending on the approximate solution of a suitable dual problem. Then the adaptive Uzawa algorithm proposed in [1, 8] is employed as a building block of two new iterative algorithms, named `ago` and `nago`, with the aim of computing an approximation u_h to u, such that $|J(u - u_h)|$ be less than a given tolerance τ.
The `ago` algorithm, moving from a given approximation z_H to the dual solution, yields, after `m` iterations, `m` being estimated a priori from the data only, an approximation, $U_{\mathtt{m}}$ to the primal solution u, such that it holds $|J(u - U_{\mathtt{m}})| \leq \tau$.
The `nago` algorithm improves on the `ago` one in terms of flexibility, as it may be interpreted as `ago` combined with a fractional step method. In more detail, it does not require anymore z_H to satisfy any accuracy demand a priori . Both the algorithms are theoretically analyzed and numerically investigated. The numerical results address in detail the 1D case. The two dimensional setting is still under investigation though the preliminary results considered are promising.

Acknowledgments

We wish to thank Prof. Silvia Bertoluzza for helping us improve our work.

References

[1] E. Bänsch, P. Morin and R. H. Nochetto, An adaptive Uzawa FEM for the Stokes problem: convergence without the inf-sup condition, *SIAM J. Numer. Anal.* **40** 4 (2002) 1207–1229.

[2] R. Becker and R. Rannacher, A feed-back approach to error control in finite element methods: basic analysis and examples, *East-West J. Numer. Math.* **4** 4 (1996) 237–264.

[3] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numer.* **10** (2001) 1–102.

[4] P. Binev, W. Dahmen and R. DeVore, Adaptive finite element methods with convergence rates, *Numer. Math.* **97** 2 (2004) 219–268.

[5] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Series in Computational Mathematics **15** (Springer Verlag, New York, 1991).

[6] A. Cohen, W. Dahmen and R. DeVore, Adaptive wavelet methods for elliptic operator equations, *Math. Comp.* **70** (2001) 27–75.

[7] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, *SIAM J. Numer. Anal.* **33** 3 (1996) 1106–1124.

[8] S. Dahlke, W. Dahmen and K. Urban, Adaptive wavelet methods for saddle point problem – optimal convergence rates, *SIAM J. Numer. Anal.* **40** 4 (2002) 1230–1262.

[9] M. B. Giles and N. A. Pierce, Adjoint equations in CFD: duality, boundary conditions and solution behaviour, *AIAA Paper* **97-1850** (1997).

[10] M. B. Giles and E. Süli, Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality, *Acta Numer.* **11** (2002) 145–236.

[11] F. Hecht, O. Pironneau and K. Ohtsuka, FreeFem++ manual - version 1.45-7, http://www.ann.jussieu.fr/~hecht/freefem++.htm (2005).

[12] L. Machiels, Y. Maday and A. T. Patera, Output bounds for reduced-order approximations of elliptic partial differential equations, *Comput. Methods Appl. Mech. Engrg.* **190** 26-27 (2001) 3413–3426.

[13] Y. Maday and A. T. Patera, Numerical analysis of a posteriori finite element bounds for linear functional outputs, *Math. Models Methods Appl. Sci.* **10** (2000) 785–799.

[14] G. I. Marchuk, *Adjoint Equations and Analysis of Complex Systems* (Kluwer Academic Publishers, Dordrecht, 1995).

[15] P. Morin, R. H. Nochetto and K. G. Siebert, Data oscillation and convergence of adaptive FEM, *SIAM J. Numer. Anal.* **38** 2 (2000) 466-488.

[16] J. T. Oden and S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, *Comput. Math. Appl.* **41** 5-6 (2001) 735–756.

[17] M. Paraschivoiu and A. T. Patera, A hierarchical duality approach to bounds for the outputs of partial differential equations, *Comput. Methods Appl. Mech. Engrg.* **158** 3-4 (1998) 389–407.

[18] M. Paraschivoiu, J. Peraire and A. T. Patera, A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations, *Comput. Methods Appl. Mech. Engrg.* **150** 1-4 (1997) 289–312.

[19] D. A. Venditti and D. L. Darmofal, Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow, *J. Comput. Phys.* **164** 1 (2000) 204–227.