AUTOMATED GOAL-ORIENTED ERROR CONTROL I: STATIONARY VARIATIONAL PROBLEMS

MARIE E. ROGNES* AND ANDERS LOGG*†

Abstract. This article presents a general and novel approach to the automation of goal-oriented error control in the solution of nonlinear stationary finite element variational problems. The approach is based on automated linearization to obtain the linearized dual problem, automated derivation and evaluation of a posteriori error estimates, and automated adaptive mesh refinement to control the error in a given goal functional to within a given tolerance. Numerical examples representing a variety of different discretizations of linear and nonlinear partial differential equations are presented, including Poisson's equation, a mixed formulation of linear elasticity, and the incompressible Navier–Stokes equations.

Key words. finite element method, a posteriori, error control, dual problem, adaptivity, automation, nonlinear

AMS subject classifications. 65N30, 68N30

1. Introduction.

For any numerical method, it is of critical importance that the accuracy of computed solutions may be assessed. For the numerical solution of differential equations, the accuracy is typically assessed manually by computing a sequence of solutions using successive refinement until it is judged that the solution has "converged". This approach is unreliable as well as time-consuming. It may also be impossible, since computing resources may be exhausted long before convergence has been reached.

For finite element discretizations, classic *a posteriori* error analysis provides a framework for controlling the approximation error measured in some Sobolev norm, cf. [1]. Over the last two decades, *goal-oriented* error control has been developed as an extension of the classic *a posteriori* analysis [9, 12]. The problem of goal-oriented error control for stationary variational problems can be posed as follows. Consider the following canonical variational problem: find $u \in V$ such that

$$F(u; v) = 0 \quad \forall\, v \in \hat{V}, \tag{1.1}$$

where $F : V \times \hat{V} \to \mathbb{R}$ is a semilinear form (linear in v) on a pair of trial and test spaces $(V, \hat{V})$. A goal-oriented adaptive algorithm seeks to find an approximate solution $u_h \approx u$ of (1.1) such that

$$\eta = |\mathcal{M}(u) - \mathcal{M}(u_h)| \leq \epsilon,$$

where $\mathcal{M} : V \to \mathbb{R}$ is a given goal functional, $\epsilon > 0$ is a given tolerance, and η is here defined as the error in the given goal functional. In other words, goal-oriented error control allows the construction of an adaptive algorithm that targets a simulation to the efficient computation of a specific quantity of interest.

The framework developed in [9, 12] provides a general method for deriving an *a posteriori* estimate of the error and adaptive refinement indicators, based on the

*Center for Biomedical Computing at Simula Research Laboratory, P.O. Box 134, 1325 Lysaker, Norway (meg@simula.no, logg@simula.no). This work is supported by an Outstanding Young Investigator grant from the Research Council of Norway, NFR 180450. This work is also supported by a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory.

†Department of Informatics, University of Oslo, Norway.

solution of an auxiliary linearized adjoint (dual) problem. This framework is directly applicable to a large class of finite element variational problems. However, a certain level of expertise is required to derive the error estimate for a particular problem and to implement the corresponding adaptive solver. In particular, the derivation of the dual problem involves the linearization of a possibly complicated nonlinear problem. Furthermore, both the derivation and evaluation of the *a posteriori* error estimate remain nontrivial (at least in practice). Moreover, each derivation must be carried out on a problem-by-problem basis. As a result, goal-oriented error control remains a tool for experts and usually requires a substantial effort to implement.

In this work, we seek to automate goal-oriented error control. We present a fully automatic approach for computation of error estimates and adaptive refinement; that is, without any manual analysis, preparation, or intervention. To our knowledge, no such automation has previously been presented, and in particular not realized. The strategy presented here requires a minimal amount of input and expert knowledge; the only input required by our adaptive algorithm is the semilinear form F, the functional $\mathcal{M}$, and the tolerance ϵ. Based on the given input, the adaptive algorithm automatically generates the dual problem, the *a posteriori* error estimate, and attempts to compute an approximate solution u_h that meets the given tolerance for the given functional. In particular, problem-tuned error estimates and indicators are generated without any manual derivations. This has the potential of rendering state-of-the-art goal-oriented error control fully accessible to non-experts at no additional implementational cost.

We emphasize that although one may, in principle, manually carry out the necessary analysis and implementation in any particular case, automation plays an important role since it (i) makes expert knowledge accessible to non-experts, (ii) speeds up development cycles, and (iii) enables more complex problems to be tackled which would otherwise require considerable effort to analyze and implement. A similar and successful automation effort as part of the FEniCS Project [27, 19, 20, 29] has led to the development of methodology and tools that automate the discretization of a large class of partial differential equations by the finite element method.

In this paper, we limit the discussion to stationary variational problems. The error estimates and indicators generated by the automated algorithm can be viewed as a version of the dual-weighted-residual estimates of [9]. We emphasize that the main target of this paper is to present an automation of a method for goal-oriented error control; in particular, it is not our intention to improve the method theoretically nor in detail examine the cases where the method is known to work poorly. Also, the question of automated error control for time-dependent problems will be considered in later works.

The remainder of this introduction describes the organization of the paper. The first two sections, Sections 2 and 3, establish the abstract problem setting by defining notation and summarizing the well-established goal-oriented error estimation framework for linear variational problems. The linear setting is discussed first for the sake of clarity and brevity: the extension to the nonlinear case is summarized in Section 6. The primary novel contributions of this paper are contained in Sections 4, 5, and 7: the main result of this paper is an automated strategy for the computation of error estimates and indicators. This strategy relies on two key components: first, a procedure, applicable to a general class of stationary variational problems, for the derivation of a strong residual representation from a weak residual representation (Section 4). Second, the evaluation of the error estimates relies on a dual approximation: an af-

fordable strategy for obtaining an improved dual approximation by extrapolation for a general class of finite element spaces is described in Section 5.

Combining the goal-oriented error estimation framework with the automated techniques introduced in Section 4 and 5, yields the automated adaptive algorithm described in Section 7. A realization of this automated algorithm has been implemented as part of the FEniCS Project, and is, in particular, available through both the Python and C++ versions of the DOLFIN library. A simple example of its use and some aspects of the implementation are also discussed in this section. In Section 8, we apply the presented framework to three examples of varying complexity: the Poisson equation, a three-field mixed formulation for the linear elasticity equations, and the stationary Navier–Stokes equations. Finally, we conclude and discuss further work in Section 9.

2. Notation. Throughout this paper, $\Omega \subset \mathbb{R}^d$ denotes an open, bounded domain with boundary $\partial\Omega$. We will generally assume that Ω is polyhedral such that it can be exactly represented by an admissible, simplicial tessellation $\mathcal{T}_h$. The boundary will typically be the union of two disjoint parts, denoted $\partial\Omega_D$ and $\partial\Omega_N$.

In general, the notation $V(X;Y)$ is used to denote the space of fields $X \to Y$ with regularity properties specified by V. If $Y = \mathbb{R}$, this argument is omitted. For $L^2(K;\mathbb{R}^d)$; that is, the space of d-vector fields on $K \subseteq \Omega$ in which each component is square integrable, the inner product reads $\langle\cdot,\cdot\rangle_K$, and the norm is denoted $||\cdot||_K$. If $K = \Omega$, the subscript is omitted. For $m = 1, 2, \ldots$, $H^m(\Omega)$ denotes the space of square integrable functions with m square integrable distributional derivatives. Also, $H^1_{g,\Gamma} = \{u \in H^1(\Omega) : u|_\Gamma = g\}$. Similarly, $H(\mathrm{div}, \Omega)$ denotes the space of square integrable vector fields with square integrable divergence. Note that both the gradient of a vector field and the divergence of a matrix field are applied row-wise.

A form $a : W_1 \times \cdots \times W_n \times V_\rho \times \cdots \times V_1 \to \mathbb{R}$, written $a(w_1, \ldots, w_n; v_\rho, \ldots, v_1)$, is (possibly) nonlinear in all arguments preceding the semi-colon, but linear in all arguments following the semi-colon.

3. A framework for goal-oriented error control. In this section, we present a general framework for goal-oriented error control for conforming finite element discretizations of stationary variational problems. The framework is a summary of the paradigm developed in [9, 12]. For clarity, we restrict our attention to linear variational problems and linear goal functionals. Extensions to nonlinear problems and nonlinear goal functionals are made in Section 6.

Let V and $\hat{V}$ be Hilbert spaces of functions or fields defined on a domain $\Omega \subset \mathbb{R}^d$ for $d = 1, 2, 3$. In this section, we consider the following linear variational problem: find $u \in V$ such that

$$a(u, v) = L(v) \quad \forall\, v \in \hat{V}. \tag{3.1}$$

We assume that $a : V \times \hat{V} \to \mathbb{R}$ is a continuous, bilinear form, and that $L : \hat{V} \to \mathbb{R}$ is a continuous, linear form. We shall further assume that the problem is well-posed; that is, there exists a unique solution u that depends continuously on any given data. The variational problem defined by (3.1) will be referred to as the *primal problem* and u will be referred to as the *primal solution*.

Let $\mathcal{T}_h$ be an admissible simplicial tessellation of Ω (to be determined) and assume that $V_h \subset V$ and $\hat{V}_h \subset \hat{V}$ are finite element spaces defined relative to $\mathcal{T}_h$. The finite element approximation of (3.1) then reads: find $u_h \in V_h$ such that

$$a(u_h, v) = L(v) \quad \forall\, v \in \hat{V}_h. \tag{3.2}$$

We assume that the spaces V_h and $\hat{V}_h$ satisfy an appropriate discrete inf–sup condition such that a unique discrete solution exists. The problem (3.2) will be referred to as the discrete primal problem and u_h the discrete primal solution.

We are interested in estimating the magnitude of the error in a given goal functional $\mathcal{M} : V \to \mathbb{R}$. Moreover, for a given tolerance $\epsilon > 0$, we aim to find $(V_h, \hat{V}_h)$ such that the corresponding finite element approximation u_h, as defined by (3.2), satisfies

$$\eta \equiv |\mathcal{M}(u) - \mathcal{M}(u_h)| \leq \epsilon. \tag{3.3}$$

In addition, we would like to compute the value of the goal functional $\mathcal{M}(u_h)$ efficiently, ideally using a minimal amount of work.

In order to estimate the magnitude of the error η, we first define the (weak) residual relative to the approximation u_h,

$$r(v) = L(v) - a(u_h, v). \tag{3.4}$$

Some remarks are in order. First, r is a bounded, linear functional by the continuity and linearity of a and L. Second, as a consequence of the Galerkin orthogonality implied by $\hat{V}_h \subset \hat{V}$, the residual vanishes on $\hat{V}_h$. In other words,

$$r(v) = 0 \quad \forall\, v \in \hat{V}_h. \tag{3.5}$$

Next, we define the (weak) dual problem: find $z \in V^*$ such that

$$a^*(z, v) = \mathcal{M}(v) \quad \forall\, v \in \hat{V}^*, \tag{3.6}$$

where $(V^*, \hat{V}^*)$ is the pair of dual trial and test spaces, and a^* denotes the adjoint of a; that is, $a^*(v, w) = a(w, v)$. We shall assume that the dual problem (3.6) is well-posed, and that there thus exists a *dual solution* z solving (3.6) with continuous dependence on the input data. Moreover, we assume that the dual trial and test spaces are chosen such that $u - u_h \in \hat{V}^*$ and $z \in \hat{V}$. This holds if $\hat{V}^* = V_0 = \{v - w : v, w \in V\}$ and $V^* = \hat{V}$.

Combining (3.6), (3.4), and (3.1), we find that

$$\mathcal{M}(u) - \mathcal{M}(u_h) = a^*(z, u - u_h) = a(u - u_h, z) = L(z) - a(u_h, z) \equiv r(z).$$

The error $\mathcal{M}(u) - \mathcal{M}(u_h)$ is thus equal to the (weak) residual r evaluated at the dual solution z. By the Galerkin orthogonality (3.5), we obtain the following error representation:

$$\mathcal{M}(u) - \mathcal{M}(u_h) = r(z) = r(z - \pi_h z). \tag{3.7}$$

Here, $\pi_h z \in \hat{V}_h$ is an arbitrary test space field, typically an interpolant of the dual solution.

An identical error representation is obtained for nonlinear variational problems and nonlinear goal functionals with a suitable definition of the dual problem. We return to this issue in Section 6. It follows that if one can compute (or approximate) the solution of the dual problem, one may estimate the size of the error by a direct evaluation of the residual. However, some concerns remain that require special attention. First, the error representation (3.7) is not directly useful as an error indicator. The derivation of an *a posteriori* error estimate and corresponding error indicators from the error representation has traditionally required manual analysis, typically involving

some form of integration by parts and a redistribution of boundary terms (fluxes) over cell facets. Second, for (3.7) to give a useful estimate of the size of the error, care must be taken when solving the dual problem (3.6). In particular, the error representation evaluates to zero if the dual solution is approximated in $\hat{V}_h$. Finally, the derivation of the dual problem may involve the differentiation of a nonlinear variational form. We discuss how each of these issues can be automated in the subsequent sections.

4. Automated derivation of error estimates and error indicators. This section presents a novel approach to the derivation of an *a posteriori* error estimate and corresponding error indicators for a general class of stationary variational problems. The starting point is the general abstract error representation (3.7). The proposed generic representation for the estimate and indicators is motivated and introduced in Section 4.1, followed by a strategy for computing this representation in Sections 4.2–4.3. The key idea is the computation of a problem-tuned strong residual representation using only ingredients from the general abstract form; this concept constitutes one of the pillars for the complete automated strategy. We emphasize that the resulting error estimate coincides with classical duality-based error estimates that may be derived manually (using integration by parts) for standard problems such as the Poisson problem. However, we present here an approach that allows these error estimates to be generated and evaluated automatically. We also note that the automatically derived error indicators differ from the standard duality-based error estimates resulting from an integration by parts followed by one or more inequalities.

4.1. A generic residual representation. For motivational purposes, we start by considering the standard derivation of a dual-weighted residual error estimator for Poisson's equation: $-\Delta u = f$ and its corresponding variational problem defined by $a(u,v) = \langle \operatorname{grad} u, \operatorname{grad} v\rangle$ and $L(v) = \langle f, v\rangle$ on $V = \hat{V} = H_0^1(\Omega)$. By integrating the weak residual by parts cell-wise, one obtains

$$\begin{aligned} r(z) &\equiv L(z) - a(u_h, z) \equiv \langle f, z\rangle - \langle \operatorname{grad} u_h, \operatorname{grad} z\rangle \\ &= \sum_{T\in\mathcal{T}} \langle f, z\rangle_T - \langle \operatorname{grad} u_h, \operatorname{grad} z\rangle_T = \sum_{T\in\mathcal{T}} \langle f + \Delta u_h, z\rangle_T + \langle -\partial_n u_h, z\rangle_{\partial T} \\ &= \sum_{T\in\mathcal{T}} \langle f + \Delta u_h, z\rangle_T + \langle [-\partial_n u_h], z\rangle_{\partial T}, \end{aligned}$$

where $[-\partial_n u_h]$ denotes an appropriate redistribution of the flux over cell facets. Several choices are possible, see for example [1, Chap. 6], but we here make the simplest possible choice and distribute the flux equally. In particular, we define $[\partial_n u_h]|_S = \frac{1}{2}(\operatorname{grad} u_h|_T \cdot n + \operatorname{grad} u_h|_{T'} \cdot n')$ over all internal facets S shared by two cells T and T', and $[\partial_n u_h]|_S = \partial_n u_h|_S$ on external facets (facets on the boundary of Ω). Hence, one may estimate the error by

$$|\mathcal{M}(u) - \mathcal{M}(u_h)| \leq \sum_{T\in\mathcal{T}} \eta_T, \tag{4.1}$$

where the error indicator η_T is given by

$$\eta_T = |\langle f + \Delta u_h, z - \pi_h z\rangle_T + \langle [-\partial_n u_h], z - \pi_h z\rangle_{\partial T}|. \tag{4.2}$$

We note that although one may in principle use $\eta_T = |\langle f, z\rangle_T - \langle \operatorname{grad} u_h, \operatorname{grad} z\rangle_T|$ as an error indicator (without integrating by parts and redistributing the normal derivative), that indicator is much less efficient than the error indicator defined

in (4.2). Both indicators will sum up to the same value (if taken with signs), but only as a result of cancellation. The error indicator (4.2) is generally smaller in magnitude, scales better with mesh refinement, and gives a sharper error bound when summed without signs. See [39] for an extended discussion.

Estimates similar to (4.1) have been derived by hand (originally for use with norm-based error indicators) for a variety of equations. A non-exhaustive list of examples (including purely norm-based indicators) includes standard finite element discretizations of the Poisson equation [3], various mixed formulations for the Stokes equations and stationary Navier–Stokes equations [37], $H(\mathrm{div})$-based discretizations of the mixed Poisson and mixed elasticity equations [10, 30], and $H(\mathrm{curl})$-based discretizations for problems in electromagnetics [7]. Duality-based goal-oriented error estimates have been derived for a number of applications, including ordinary differential equations [13], plasticity [32], hyperbolic systems [22], reactive compressible flow [36], systems of nonlinear reaction–diffusion equations [14, 35], eigenvalue problems [17], wave propagation [4], radiative transfer [33], nonlinear elasticity [25], the incompressible Navier–Stokes equations [8, 18], variational multiscale problems [24], and multiphysics problems [23].

These estimates share a common factor, namely that the error is expressed as a sum of contributions from the cells and the facets of the mesh. Moreover, each of these estimates has been derived manually for the specific problem at hand. Here, we aim to demonstrate that for a large class of variational problems, one may automatically compute an equivalent residual representation. The representation takes the following generic form:

$$r(v) = \sum_{T \in \mathcal{T}_h} \langle R_T, v \rangle_T + \langle R_{\partial T}, v \rangle_{\partial T} = \sum_{T \in \mathcal{T}_h} \langle R_T, v \rangle_T + [\langle R_{\partial T}, v \rangle_{\partial T}], \tag{4.3}$$

where

$$[\langle R_{\partial T}, v \rangle_{\partial T}] = \sum_{S \subset \partial T \cap \Omega} \frac{1}{2} \left(\langle R_{\partial T}, v|_T \rangle_S + \langle R_{\partial T'}, v|_{T'} \rangle_S \right) + \sum_{S \subset \partial T \cap \partial\Omega} \langle R_{\partial T}, v \rangle_S.$$

It follows that one may use as error indicators

$$\eta_T = |\langle R_T, z - \pi_h z \rangle_T + [\langle R_{\partial T}, z - \pi_h z \rangle_{\partial T}]|. \tag{4.4}$$

In these expressions, R_T denotes a residual contribution evaluated over the domain of a cell T, whereas $R_{\partial T}$ denotes a residual contribution evaluated over a cell boundary ∂T.

4.2. Automatic computation of the residual representation. We shall focus our attention on a class of residuals r satisfying the following assumptions:
A1 (Global decomposition) The residual is a sum of local contributions:

$$r = \sum_{T \in \mathcal{T}_h} r_T.$$

A2 (Local decomposition) Each local residual r_T offers a local decomposition:

$$r_T(v) = \langle R_T, v \rangle_T + \langle R_{\partial T}, v \rangle_{\partial T} \quad \forall\, v \in \hat{V}|_T. \tag{4.5}$$

We note that A1 is satisfied if the bilinear and linear forms a and L are expressed as integrals over the cells and facets of the tessellation $\mathcal{T}_h$. We also note that A2 is

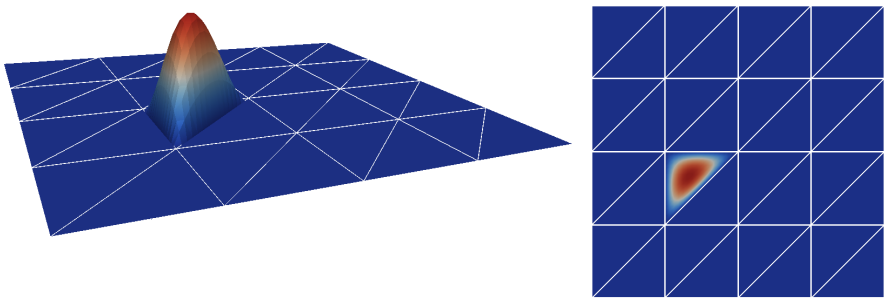


FIG. 4.1. *The bubble function* b_T.

satisfied if the variational problem (3.1) has been derived by testing a partial differential equation against a test function and (possibly) integrating by parts to move derivatives onto the test function.

For the sake of a simplified analysis, we also make the following assumption:

A3 (Polynomial representation) The residual contributions (or, in the case of vector or tensor fields, each scalar component of these) are piecewise polynomial:

$$R_T \in \mathcal{P}^p(T), \quad R_{\partial T}|_S \in \mathcal{P}^q(S) \quad \forall\, S \in \partial T \quad \forall\, T \in \mathcal{T}_h, \quad p, q \in \mathbb{N}.$$

We discuss the implications of this assumption below, but note that one of the numerical examples presented does not satisfy this assumption.

In the following, we shall show that a residual representation may be automatically computed if assumptions A1–A2 are satisfied. More precisely, if assumptions A1–A3 are satisfied, we shall show that one may automatically compute the exact residual representation (4.3) for a given variational problem (3.1). In particular, one may directly compute the cell and facet residuals R_T and $R_{\partial T}$ by solving a set of local problems on each cell T of the tessellation $\mathcal{T}_h$. If A3 fails; that is, if only assumptions A1–A2 are satisfied, the automated procedure computes weighted L^2-projections of the residual decomposition terms and hence an approximate residual representation.

To compute the cell residual R_T, let $\{\phi_i\}_{i=1}^m$ be a basis for $\mathcal{P}^p(T)$ and let b_T denote the bubble function on T. We recall that for a simplex $T \subset \mathbb{R}^d$, the bubble function b_T is defined by

$$b_T = \prod_{i=1}^{d+1} \lambda_{x_i}^T$$

where $\lambda_{x_i}^T$ is the barycentric coordinate function on T associated with vertex x_i (the ith linear Lagrange nodal basis function on T). Note that b_T vanishes on the boundary of T. See Figure 4.1 for an illustration. Testing the local residual r_T against $b_T\phi_i$, we obtain the following local problem for the cell residual R_T: find $R_T \in \mathcal{P}^p(T)$ such that

$$\langle R_T, b_T\phi_i\rangle_T = r_T(b_T\phi_i), \quad i = 1, \ldots m. \tag{4.6}$$

To obtain a local problem for the facet residual $R_{\partial T}$, we define for each facet S on T the *cone function* β_S^T by

$$\beta_S^T = \prod_{i \in I_S^T} \lambda_{x_i}^T, \tag{4.7}$$

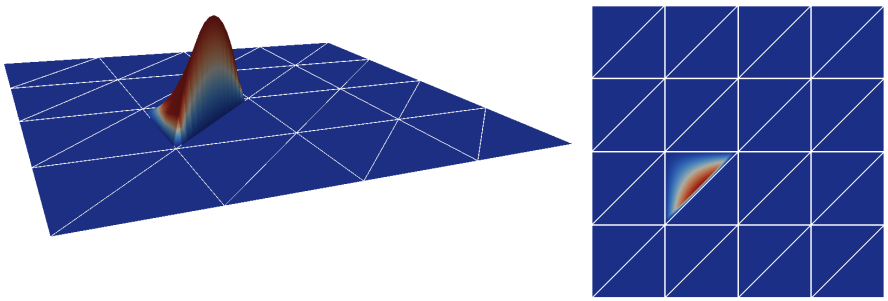


FIG. 4.2. *The cone function* β_S^T.

where I_S^T is a suitably defined index set such that $\beta_S^T|_f \equiv 0$ on all facets f of T but S. For an illustration, see Figure 4.2. Clearly, $\beta_S^T|_S = b_S$. Next, let $\{\phi_i\}_{i=1}^n$ be a basis for $\mathcal{P}^q(T)$. Testing the local residual r_T against $\beta_S^T\phi_i$, we obtain the following local problem for each facet residual: find $R_{\partial T}|_S \in \mathcal{P}^q(S)$ such that

$$\langle R_{\partial T}|_S, \beta_S^T\phi_i\rangle_S = r_T(\beta_S^T\phi_i) - \langle R_T, \beta_S^T\phi_i\rangle_T \quad \forall\, i \in I_S^T. \tag{4.8}$$

We prove below that by assumptions A1–A3, the local problems (4.6) and (4.8) uniquely define the cell and facet residuals R_T and $R_{\partial T}$ of the residual representation (4.3).

One may thus compute the residual representation (4.3) by solving a set of local problems on each cell T. First, one local problem for the cell residual R_T, and then local problems for the facet residual $R_{\partial T}$ restricted to each facet of T. If the test space is vector-valued, the local problems are solved for each scalar component. We emphasize that the computation of the residual representation (4.3) and thus the error indicator (4.4) may be computed automatically given only the variational problem (3.1) in terms of the pair of bilinear and linear forms a and L. In particular, the derivation of the error indicators does not involve any manual analysis.

We remark that the use of bubble and cone functions to localize the weak residual is a standard technique for proving reliability and efficiency for norm-based error estimates (see e.g. [38]). The crucial observation here is that this technique can also be used to automatically generate the residual decomposition given only the weak residual. We also remark that the local problems (4.6) and (4.8) are different from the local problems that were introduced in [6] to represent the cell and facet residuals R_T and $R_{\partial T}$ as a single residual.

4.3. Solvability of the local problems. To prove that the local problems (4.6) and (4.8) uniquely determine the cell and facet residuals, we recall the following result regarding bubble-weighted L^2-norms. For a proof, we refer to [1, Theorems 2.2, 2.4].

LEMMA 4.1. *Let T be a d-simplex and let b_T denote the bubble function on T. There exist positive constants c and C, independent of T, such that*

$$c||\phi||_T^2 \leq \langle b_T\phi, \phi\rangle_T \leq C||\phi||_T^2 \tag{4.9}$$

for all $\phi \in \mathcal{P}^p(T)$.

We may now prove the following theorem.

THEOREM 4.1. *If assumptions A1–A3 hold, then the cell and facet residuals of the residual representation* (4.3) *are uniquely determined by the local problems* (4.6) *and* (4.8).

Proof. Consider first the cell residual R_T. Take $v = b_T\phi_i$ in (4.5) for $i = 1, \ldots, m$. Since v vanishes on the cell boundary ∂T, we obtain (4.6). By assumption, $R_T \in \mathcal{P}^p(T)$ and is thus a solution of the local problem (4.6). It follows from Lemma 4.1, that it is the unique solution. We similarly see that the facet residual $R_{\partial T}$ is a solution of the local problem (4.8) and uniqueness follows again from Lemma 4.1. □

In the cases where A3 fails, such as if the variational problem contains non-polynomial data, the local problems (4.6) and (4.8) uniquely determine the projections of R_T and $R_{\partial T}|_S$ onto $\mathcal{P}^p(T)$ and $\mathcal{P}^q(S)$ respectively. The accuracy of the approximation may then be controlled by the polynomial degrees p and q. In the numerical examples presented in Section 8, we let $p = q$ be determined by the polynomial degree of the finite element space. We have not observed any significant errors introduced by this approximation in our numerical experiments.

Looking back at the special case of Poisson's equation (4.2), the cell and facet residuals derived by hand are given by $R_T = f + \Delta u_h$ and $R_{\partial T} = -\partial_n u_h$, respectively. We emphasize that, by what is just shown, this indeed coincides with the representation defined by (4.6) and (4.8) if f is polynomial.

5. Approximating the dual solution. In order to evaluate the error representation (3.7) and to compute the error indicators (4.4), one must compute, or in practice approximate, the solution z of the dual problem (3.6). The natural discretization of (3.6) reads: find $z_h \in V_h^* = \hat{V}_h$ such that

$$a^*(z_h, v) = \mathcal{M}(v) \quad \forall\, v \in \hat{V}_h^* = V_{h,0}. \tag{5.1}$$

However, since the residual r vanishes on $\hat{V}_h$, z_h is, for the purpose of error estimation, highly unsuitable as an approximation of the dual solution.

An immediate alternative is to solve the dual problem using a higher order method. If the dual solution is sufficiently regular, a higher order method would be expected to give a more accurate dual approximation. It is observed in practice that a more accurate dual approximation gives a better error estimate [9], although complete reliability cannot be guaranteed [31]. Other alternatives include approximation by hierarchic techniques [1, 5] or approximating the dual problem on a different mesh. In this work, we suggest a new alternative based on solving (5.1) using the same mesh and polynomial order as the primal problem and then extrapolating the computed solution z_h to a higher order function space. This strategy can be compared to the higher order interpolation procedure presented in [9] for regular quadrilateral/hexahedral meshes. The strategy presented here extends that of [9] however, as it can be applied to almost arbitrary (admissible) simplicial tessellations.

To define the extrapolation procedure, let V_h be a finite element space on a tessellation $\mathcal{T}_h$ and let $W_h \supset V_h$ be a higher order finite element space on the same tessellation $\mathcal{T}_h$. Furthermore, let $\{\phi_j^T\}_{j=1}^n$ be a local basis for W_h on T and let $\{\phi_j\}_{j=1}^N$ be the corresponding global basis. For $v_h \in V_h$, we define the extrapolation operator $E : V_h \to W_h$ as described in Algorithm 1. This algorithm computes the extrapolation by fitting local polynomials to the finite element function v_h on local patches. This yields a global multi-valued function which is then averaged to obtain the extrapolation Ev_h. We illustrate the extrapolation algorithm in Figure 5.1 for a one-dimensional case.

Algorithm 1 Extrapolation

1. (Lifting) For each cell $T \in \mathcal{T}_h$:
   (a) Define a patch of cells $\omega_T \supset T$ of sufficient size and let $\{\ell_i\}_{i=1}^m$ be the collection of degrees of freedom for V_h on the patch. The size of the patch ω_T should be such that the number of degrees of freedom m is greater than or equal to the local dimension n of $W_h|_T$.
   (b) Let $\{\phi_j^{\omega_T}\}_{j=1}^n$ be a smooth extension of $\{\phi_j^T\}_{j=1}^n$ to the patch ω_T.
   (c) Define $A_{ij} = \ell_i(\phi_j^{\omega_T})$ and $b_i = \ell_i(v_h)$ for $i = 1, \ldots, m$, $j = 1, \ldots, n$.
   (d) Compute the least-squares approximation ξ_T of the (overdetermined) $m \times n$ system $A\xi_T = b$.
2. (Smoothing)
   (a) For each global degree of freedom j, let X_j be the set of corresponding local expansion coefficients determined on each cell T by the local vector ξ_T. Define $\xi_j = \frac{1}{|X_j|} \sum_{x \in X_j} x$. We note that $|X_j| > 1$ for degrees of freedom that are shared between cells.
   (b) Define $Ev_h = \sum_{j=1}^N \xi_j \phi_j$.

Algorithm 1 may be used to compute a higher order approximation of the dual solution z as follows. First, we compute an approximation $z_h \in \hat{V}_h$ of the dual solution by solving (5.1). We then compute the extrapolation $Ez_h \in W_h$ where W_h is the finite element space on $\mathcal{T}_h$ obtained by increasing the polynomial degree by one. We then estimate the error by

$$\eta \equiv |\mathcal{M}(u) - \mathcal{M}(u_h)| = |r(z)| \approx |r(Ez_h)| \equiv \eta_h.$$

6. Extensions to nonlinear problems and goal functionals. We now turn to consider nonlinear variational problems and goal functionals. We consider the following general nonlinear variational problem: find $u \in V$ such that

$$F(u; v) = 0 \quad \forall\, v \in \hat{V}. \tag{6.1}$$

For a given nonlinear goal functional $\mathcal{M} : V \to \mathbb{R}$, we define the following dual problem: find $z \in V^*$ such that

$$\overline{F'}^*(z, v) = \overline{\mathcal{M}'}(v) \quad \forall\, v \in \hat{V}^*, \tag{6.2}$$

where, as before, $\hat{V}^* = V_0$ and $V^* = \hat{V}$. The bilinear form $\overline{F'}$ is an appropriate average of the Fréchet derivative $F'(u; \delta u, v) \equiv \frac{\partial F(u;v)}{\partial u}\, \delta u$ of F,

$$\overline{F'}(\cdot, \cdot) = \int_0^1 F'(su + (1-s)u_h; \cdot, \cdot)\, \mathrm{d}s. \tag{6.3}$$

We note that by the chain rule, we have $\overline{F'}(u - u_h, \cdot) = F(u; \cdot) - F(u_h; \cdot)$. The linear functional $\overline{M'}$ is defined similarly. Note that (6.2) reduces to (3.6) in the linear case where $F(u; v) = a(u, v) - L(v)$.

The following error representation now follows directly from the definition of the dual problem:

$$\begin{aligned}\mathcal{M}(u) - \mathcal{M}(u_h) &= \overline{\mathcal{M}'}(u - u_h) = \overline{F'}^*(z, u - u_h) = \overline{F'}(u - u_h, z) \\ &= F(u; z) - F(u_h; z) = -F(u_h; z) \equiv r(z).\end{aligned}$$

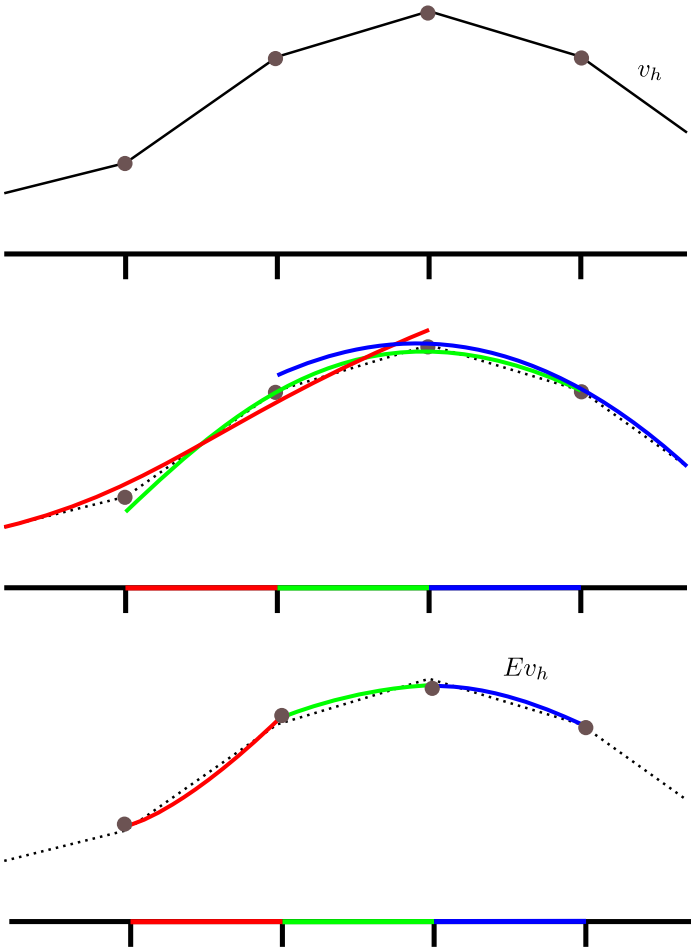


FIG. 5.1. *Extrapolation of a continuous piecewise linear function v_h to a continuous piecewise quadratic function Ev_h. The extrapolation is computed by first fitting a quadratic polynomial on each patch. In one dimension, each patch is a set of three intervals and each local quadratic polynomial is computed by solving an overdetermined 4×3 linear system. The continuous piecewise quadratic extrapolation Ev_h is then computed by averaging at the end-points of each interval.*

We thus recover the error representation (3.7).

In practice, the exact solution u is not known and must be approximated by the approximate solution u_h; that is, the linear operator $\overline{F'}$ is approximated by the derivative of F evaluated at $u = u_h$. The resulting linearization error may for the sake of simplicity be neglected, as we shall in this exposition, but doing so may reduce the accuracy (and reliability) of the computed error estimates. For a further discussion on the issue of linearization errors in the definition of the dual problem, we refer to [9].

It follows that the techniques described in Section 4 and 5 directly apply to the residual r and the dual approximation z_h also for the nonlinear case.

7. A complete algorithm for automated goal-oriented error control. Based on the above discussion, we may now phrase the complete algorithm for automated adaptive goal-oriented error control in Algorithm 2.

Algorithm 2 Adaptive algorithm

Let $F : V \times \hat{V} \to \mathbb{R}$ be a given semilinear form, let $\mathcal{M} : V \to \mathbb{R}$ be a given goal functional, and let $\epsilon > 0$ be a given tolerance.

1. Select an initial tessellation $\mathcal{T}_h$ of the domain Ω and construct the corresponding trial and test spaces $V_h \subset V$ and $\hat{V}_h \subset \hat{V}$ (for a given fixed finite element family and degree).
2. Compute the finite element solution $u_h \in V_h$ of the primal problem (6.1) satisfying $F(u_h; v) = 0$ for all $v \in \hat{V}_h$.
3. Compute the finite element solution $z_h \in V_h^*$ of the dual problem (6.2) satisfying $\overline{F'}^*(z, v) = \overline{\mathcal{M}'}(v)$ for all $v \in \hat{V}_h^*$.
4. Extrapolate $z_h \mapsto Ez_h$ using Algorithm 1.
5. Evaluate the error estimate $\eta_h = |F(u_h; Ez_h)|$.
6. If $|\eta_h| \leq \epsilon$, accept the solution u_h and break. (Stopping criterion)
7. Compute the cell and facet residuals R_T and $R_{\partial T}$ of the residual representation (4.3) by solving the local problems (4.6) and (4.8).
8. Compute the error indicators $\eta_T = |\langle R_T, Ez_h - \pi_h Ez_h\rangle_T + [\langle R_{\partial T}, Ez_h - \pi_h Ez_h\rangle_{\partial T}]|$.
9. Sort the error indicators in order of decreasing size and mark the first M cells for refinement where M is the smallest number such that $\sum_{i=1}^{M} \eta_{T_i} \geq \alpha \sum_{T \in \mathcal{T}_h} \eta_T$, for some choice of $\alpha \in (0, 1]$. (Dörfler marking [11])
10. Refine all cells marked for refinement (and propagate refinement to avoid hanging nodes).
11. Go back to step 2.

Algorithm 2 has been implemented within the FEniCS project [27, 26, 29], a collaborative project for the development of concepts and software for automated solution of differential equations. The implementation is freely available, and distributed as part of DOLFIN (version 0.9.11 and onwards). We discuss some of the features of the implementation here and provide a simple use case. More details of the implementation will be discussed in future work [34].

For the specification of variational problems, the Python interface of DOLFIN accepts as input variational forms expressed in the form language UFL [2]. Forms expressed in the UFL language are automatically passed to the FEniCS form compiler FFC [20, 21, 28] which generates efficient C++ code for finite element assembly of the corresponding discrete operators. For a detailed discussion, see [29]. Stationary discrete variational problems can be solved in DOLFIN by calling the `solve` function accepting as input a variational problem specified by a variational equation expressed by two variational forms (defining the left- and right-hand sides), the solution function `u` and any boundary conditions `bcs`. Our implementation adds the possibility of solving such problems adaptively with goal-oriented error control by adding a goal functional `M` and an error tolerance, say `1e-6`:

```
solve(a == L, u, bcs, tol=1.e-6, M=M) # Linear case
solve(F == 0, u, bcs, tol=1.e-6, M=M) # Nonlinear case
```

A simple complete example is listed in Figure 7.1. A number of optional parameters may be specified to control the behavior of the adaptive algorithm, including the marking strategy and the refinement fraction. The default marking strategy is Dörfler marking [11] with a refinement fraction of $\alpha = 0.5$.

Internally, the adaptive algorithm relies on the capabilities of the form lan-

```
from dolfin import *

mesh = UnitSquare(4, 4)
V = FunctionSpace(mesh, "CG", 1)
u = Function(V)
v = TestFunction(V)
f = Constant(1.0)

F = inner((1 + u**2)*grad(u), grad(v))*dx - f*v*dx
bc = DirichletBC(V, 0.0, "near(x[0], 0.0)")

M = u*dx
solve(F == 0, u, bc, tol=1.e-3, M=M)
```

FIG. 7.1. *Complete code for the automated adaptive solution of a nonlinear Poisson-like problem on the unit square with $f = 1.0$, homogeneous Dirichlet boundary conditions on the left boundary and homogeneous Neumann conditions on the remaining boundary, with goal functional $\mathcal{M} = \int_\Omega u \, \mathrm{d}x$.*

guage UFL for generating the dual problem, the local problems for the cell and facet residuals, and the computation of error indicators. As an illustration, we show here the code for generating the bilinear form $a^* = F'^*$ of the dual problem (6.2):

```
a_star = adjoint(derivative(F, u))
```

8. Numerical examples. In this section, we aim to investigate the performance of the automated algorithm. Since the theoretical properties of the proposed extrapolation procedure are largely unknown, the investigation here focuses on the quality of the error estimate and the sum of the error indicators on adaptively refined meshes. The total computational efficiency of the automated adaptive algorithm will be investigated in later works [34].

We present three numerical examples from three different application areas, aiming to illustrate different characteristics and varying levels of complexity. We begin by considering a basic example: a standard discretization of the Poisson equation, and evaluate the quality of the error estimates; the results show that the algorithm gives error indicators close to the optimal value of one. The second example is a discretization of a weakly symmetric formulation for linear elasticity. This discretization involves a nontrivial finite element space, namely, a mixed finite element space consisting of multiple Brezzi–Douglas–Marini elements, and multiple discontinuous and continuous elements. As far as the authors are aware, this is the first demonstration of goal-oriented error control for the discretization presented. The results show that the algorithm produces error estimates of optimal quality also for this far more complicated case. Finally, we consider a nonlinear, nonsmooth example of widespread use: a mixed discretization of the incompressible Navier–Stokes equations and evaluate both the quality of the error estimates and the performance of the adaptive algorithm.

8.1. The Poisson equation. We begin by considering the Poisson equation:

$$-\Delta u = f \text{ in } \Omega, \tag{8.1a}$$

$$u = 0 \text{ on } \partial\Omega_D, \tag{8.1b}$$

$$\partial_n u = g \text{ on } \partial\Omega_N. \tag{8.1c}$$

The standard variational formulation of (8.1) fits the framework of Section 3 with $V = \hat{V} = H^1_{0,\partial\Omega_D}(\Omega)$ and

$$a(u, v) = \langle \operatorname{grad} u, \operatorname{grad} v \rangle, \tag{8.2a}$$
$$L(v) = \langle f, v \rangle + \langle g, v \rangle_{\partial\Omega_N}. \tag{8.2b}$$

We consider the discretization of (8.2) using the space of continuous piecewise linear polynomials that satisfy the essential boundary condition for $V_h = \hat{V}_h$.

As a test case, we consider a three-dimensional L-shaped domain,

$$\Omega = ((-1, 1) \times (-1, 1) \setminus (-1, 0) \times (-1, 0)) \times (-1, 0),$$

with Dirichlet boundary $\partial\Omega_D = \{(x, y, z) : x = 1 \text{ or } y = 1\}$ and Neumann boundary $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$. Let $f(x, y, z) = -2(x - 1)$ and let $g = G \cdot n$ with $G(x, y, z) = \big((y - 1)^2, 2(x - 1)(y - 1), 0\big)$. The exact solution is then given by

$$u(x, y, z) = (x - 1)(y - 1)^2. \tag{8.3}$$

As a goal functional, we take the average value of the solution on the left boundary $\Gamma = \{(x, y, z) : x = -1\}$; that is,

$$\mathcal{M}(u) = \int_\Gamma u \, \mathrm{d}s. \tag{8.4}$$

It follows that the exact value of the goal functional is $\mathcal{M}(u) = -2/3$.

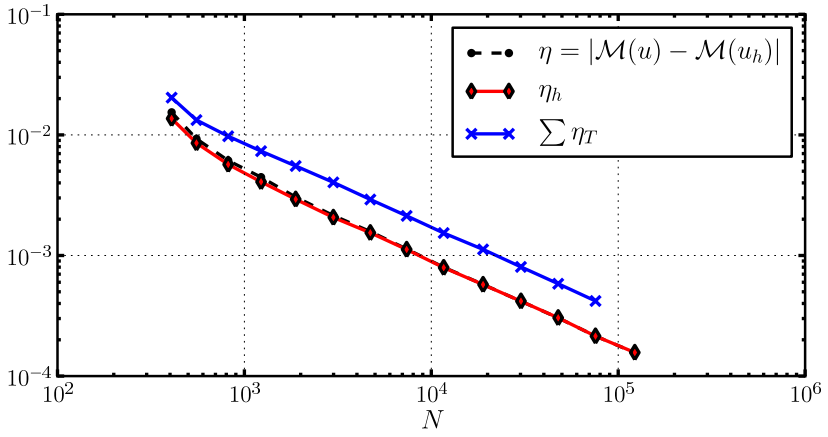
Figure 8.1 shows errors η, error estimates η_h, the sum of the error indicators $\sum_T \eta_T$, and efficiency indices η_h/η and $\sum_T \eta_T/\eta$ for a series of adaptively (and automatically) refined meshes. We first note that the error estimate η_h is very close to the error η. On the coarsest mesh, the efficiency index is $\eta_h/\eta \approx 0.89$ and as the mesh is refined, the efficiency index quickly approaches $\eta_h/\eta \approx 1$. We further note that the sum of the error indicators tends to overestimate the error, but only by a small constant factor. This demonstrates that the automatically computed error indicators are good indicators for refinement. We emphasize that since the error indicators are not used as a stopping criterion for the adaptive refinement, it is not important that they sum up to the error.

8.2. Weakly symmetric linear elasticity. As a more challenging test problem, we consider a three-field formulation for linear isotropic elasticity enforcing the symmetry of the stress tensor weakly. This gives rise to a mixed formulation that involves $H(\operatorname{div})$- and L^2-conforming spaces. For a domain $\Omega \subset \mathbb{R}^2$, the unknowns are the stress tensor $\sigma \in H(\operatorname{div}, \Omega; \mathbb{R}^{2\times 2})$, the displacement $u \in L^2(\Omega; \mathbb{R}^2)$, and the rotation $\gamma \in L^2(\Omega)$. The bilinear and linear forms read

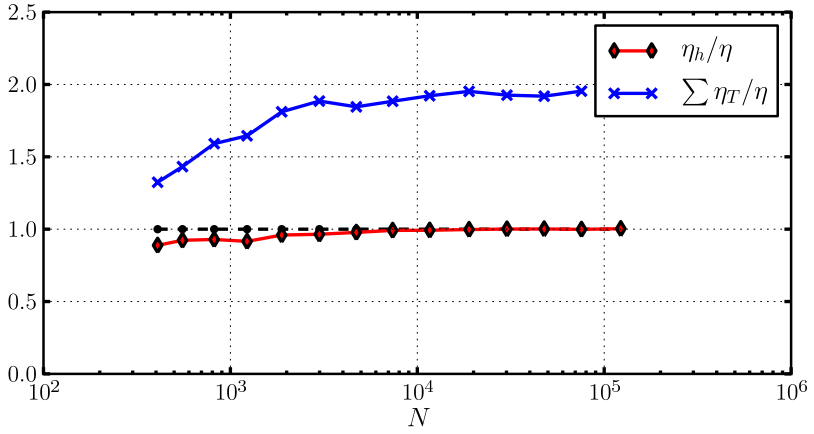
$$a((\sigma, u, \gamma), (\tau, v, \eta)) = \langle A\sigma, \tau \rangle + \langle \operatorname{div} \sigma, v \rangle + \langle u, \operatorname{div} \tau \rangle + \langle \sigma, \eta \rangle + \langle \gamma, \tau \rangle, \tag{8.5a}$$
$$L((\tau, v, \eta)) = \langle g, v \rangle + \langle u_0, \tau \cdot n \rangle_{\partial\Omega}. \tag{8.5b}$$

Here, g is a given body force, u_0 is a prescribed boundary displacement field, and A is the compliance tensor. For isotropic, homogeneous elastic materials with shear modulus μ and stiffness λ, the action of A reduces to

$$A\sigma = \frac{1}{2\mu} \left(\sigma - \frac{\lambda}{2(\mu + \lambda)} (\operatorname{tr} \sigma) I \right). \tag{8.6}$$



(a) Errors



(b) Efficiency indices

FIG. 8.1. *Errors, error estimates, and summed error indicators (top) and efficiency indices (bottom) versus the number of degrees of freedom N for adaptively refined meshes for the Poisson problem. Note the excellent agreement between the error η (dashed black curve) and the error estimate η_h (solid red curve), as well as the convergence of the efficiency index η_h/η towards* 1.

We consider the discretization of these equations by a mixed finite element space $V_h = \hat{V}_h$ consisting of the tensor fields composed of two first-order Brezzi–Douglas–Marini elements for the stress tensor, piecewise constant vector fields for the displacement, and continuous piecewise linears for the rotation [15, 16].

We consider the domain $\Omega = (0,1) \times (0,1)$ and the exact solution $u(x,y) = (xy\sin(\pi y), 0)$ for $\mu = 1$ and $\lambda = 100$, and insert

$$g = \operatorname{div} A^{-1}\varepsilon(u) = \begin{pmatrix} \pi\mu(2x\cos(\pi y) - \pi xy\sin(\pi y)) \\ \mu(\pi y\cos(\pi y) + \sin(\pi y)) + \lambda(\pi y\cos(\pi y) + \sin(\pi y)) \end{pmatrix}.$$

As a goal functional, we take a weighted measure of the average shear stress on the

right boundary,

$$\mathcal{M}((\sigma, u, \gamma)) = \int_\Gamma \sigma \cdot n \cdot (\psi, 0)\, \mathrm{d}s \approx -0.06029761071,$$

where $\Gamma = \{(x, y) : x = 1\}$ and $\psi = y(y - 1)$.

The resulting errors, error estimates, error indicators, and efficiency indices are plotted in Figure 8.2. Again, we note that the error estimate η_h is very close to the actual error η. We also note the good performance of the error indicators that overestimate the error by around a factor of $2-4$. This is remarkable, considering that that the error estimate and error indicators are derived automatically for a non-trivial mixed formulation and involve automatic extrapolation of the dual solution from a mixed $[\mathrm{BDM}_1]^2 \times \mathrm{DG}_0 \times P_1$ space to a mixed $[\mathrm{BDM}_2]^2 \times \mathrm{DG}_1 \times P_2$ space. As far as the authors are aware, this is the first demonstration of goal-oriented error control for this discretization of the formulation (8.5).

8.3. The stationary incompressible Navier–Stokes equations. Finally, we consider a stationary pressure-driven Navier–Stokes flow in a two-dimensional channel with an obstacle. We let $\Omega = \Omega_C \backslash \Omega_O$, where $\Omega_C = (0, 4) \times (0, 1)$ and $\Omega_O = (1.4, 1.6) \times (0, 0.5)$. We let $\Omega_N = \{(x, y) \in \partial\Omega, x = 0 \text{ or } x = 4\}$ denote the Neumann (inflow/outflow) boundary and let $\Omega_D = \Omega \setminus \Omega_N$ denote the Dirichlet (no-slip) boundary.

We consider the following nonlinear variational problem for the solution of the stationary incompressible Navier–Stokes equations: find $(u, p) \in V$ such that

$$F((u, p); (v, q)) = 0$$

for all $(v, q) \in \hat{V}$, where

$$F((u, p); (v, q)) = \nu\langle \mathrm{grad}\, u, \mathrm{grad}\, v\rangle + \langle \mathrm{grad}\, u \cdot u, v\rangle - \langle p, \mathrm{div}\, v\rangle + \langle \mathrm{div}\, u, q\rangle + \langle \bar{p}n, v\rangle_{\partial\Omega_N}.$$

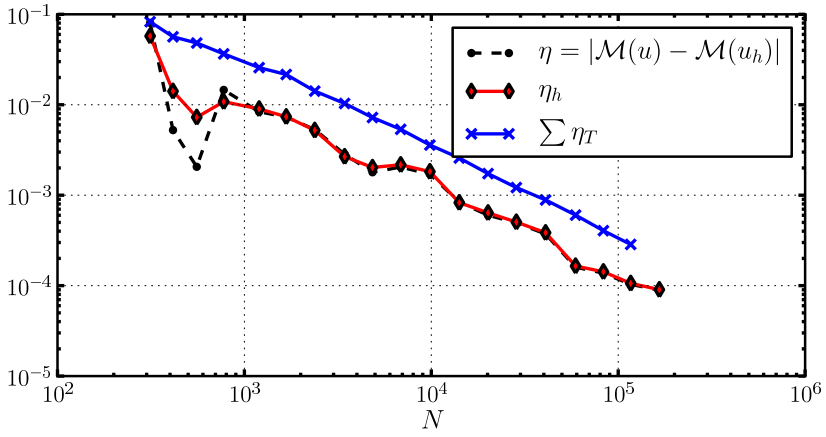
Here, $\bar{p}$ is a given boundary condition at the inflow/outflow boundary.

The trial and test spaces are given by $V = \hat{V} = H^1_{0,\partial\Omega_D}(\Omega; \mathbb{R}^2) \times L^2(\Omega)$. We let the (kinematic) viscosity be $\nu = 0.02$ and take $\bar{p} = 1$ at $x = 0$ (inflow) and $\bar{p} = 0$ at $x = 4$ (outflow). The quantity of interest is the outflux at $x = 4$,

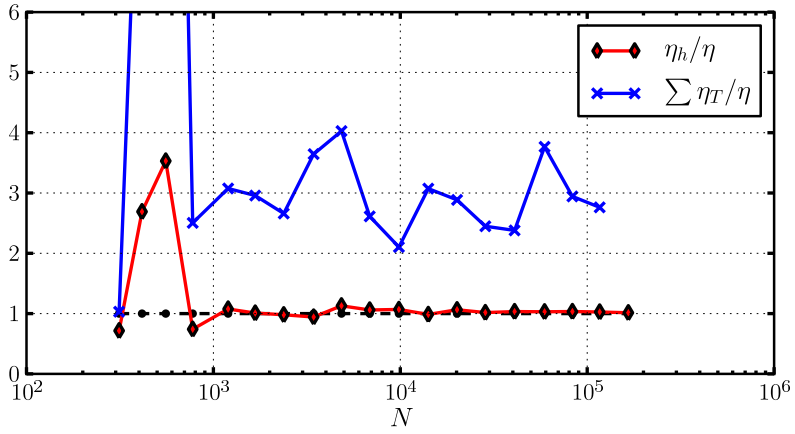
$$\mathcal{M}(u, p) = \int_{x=4} u \cdot n\, \mathrm{d}s \approx 0.40863917.$$

The system is discretized using a Taylor–Hood elements; that is, the velocity space is discretized using continuous piecewise quadratic vector fields and the pressure space is discretized using continuous piecewise linears. The nonlinear system is solved using a standard Newton iteration.

The results for this case are shown in Figure 8.3. As seen in this figure, the error estimate is not as accurate as for the two previous test cases. The efficiency index oscillates in the range $0.2-1.0$. This is not surprising, considering that (i) a linearization error is introduced when linearizing the dual problem around the computed approximate solution u_h, rather than computing the average (6.3), and (ii) both the primal and dual problems exhibit singularities at the reentrant corners making the higher-order extrapolation procedure suboptimal for approximating the exact dual solution. Still, we obtain reasonably good error estimates and error indicators. Furthermore, the adaptive algorithm performs very well when comparing the convergence obtained with the adaptively refined sequence of meshes to that of uniform refinement, cf. Figure 8.4. The final mesh is shown in Figure 8.5, and we note that it is heavily refined in the vicinity of the reentrant corners.



(a) Errors

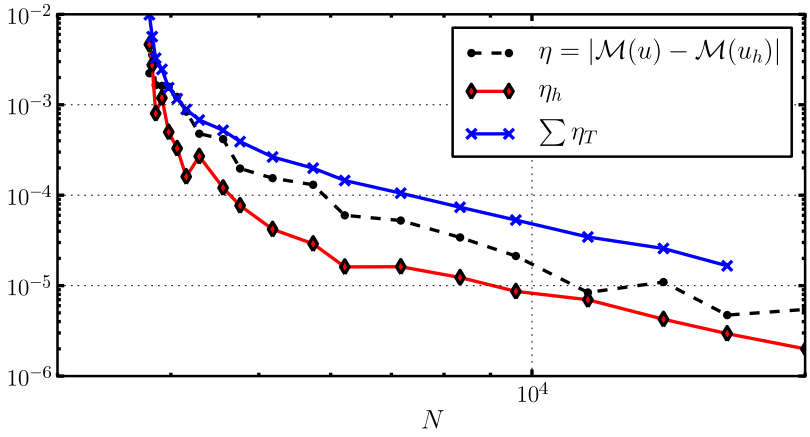


(b) Efficiency indices

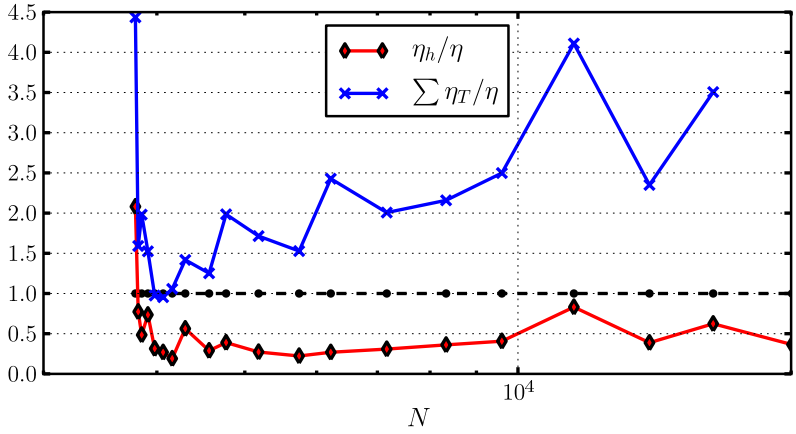
FIG. 8.2. *Errors, error estimates, and summed error indicators (top) and efficiency indices (bottom) versus the number of degrees of freedom N for adaptively refined meshes for the mixed elasticity problem. Note the excellent agreement between the error η (dashed black curve) and the error estimate η_h (solid red curve), as well as the convergence of the efficiency index η_h/η towards* 1.

9. Conclusions. We have demonstrated a new strategy for automated, adaptive solution of finite element variational problems. The strategy is implemented and freely available as part of the DOLFIN finite element library [29]; accessible both through the Python and C++ interfaces.

The strategy and its implementation are currently limited to stationary nonlinear variational problems. Another limitation is the restriction to conforming finite element discretizations. These are both issues that we plan to consider in future extensions of this work. Additionally, we have assumed that the dual problem is well-posed. This assumption may fail in cases where the primal problem has been stabilized by the introduction of additional terms; the adjoint (linearized) dual problem is then not



(a) Errors



(b) Efficiency indices

FIG. 8.3. *Errors, error estimates, and summed error indicators (top) and efficiency indices (bottom) versus the number of degrees of freedom N for adaptively refined meshes for the Navier–Stokes problem. This is a detail of the full convergence plot shown in Figure 8.4, where the convergence of the adaptive algorithm is contrasted to the convergence obtained with uniform refinement.*

necessarily well-posed. Automated error control for such formulations is an interesting topic for further research, but is beyond the scope of the present work.

Although the implementation has been tested on a number of model problems with convincing results, the effect of the linearization error (approximating $u \approx u_h$ in (6.3)) is unknown. As a consequence, the computed error estimates typically underestimate the error for nonlinear problems. The effect of the linearization error and its proper treatment remains an open (and fundamental) question. Also, the extrapolation algorithm proposed and numerically tested here should be examined from a theoretical viewpoint.

We remark that the techniques described in this paper could also be used for

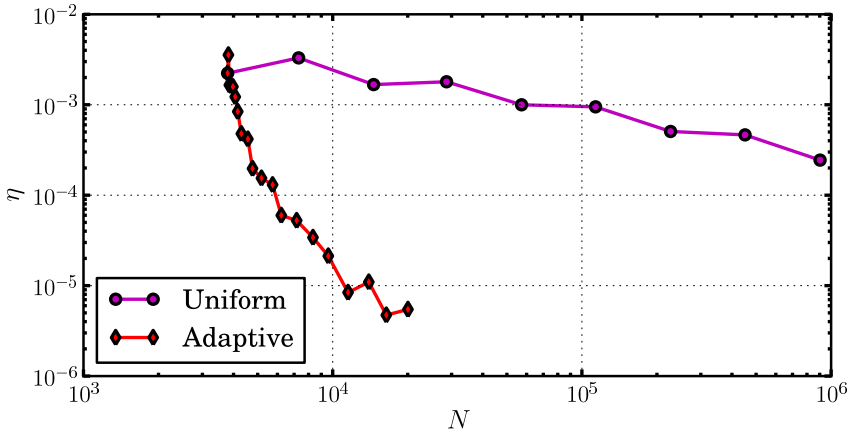


FIG. 8.4. *Convergence for adaptively and uniformly refined meshes for the Navier–Stokes problem. Adaptive refinement outperforms uniform refinement by 1–2 orders of magnitude.*

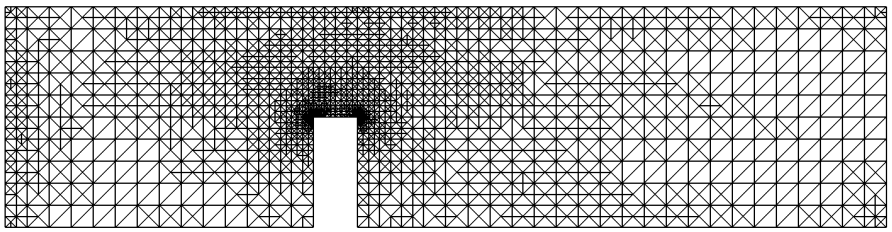


FIG. 8.5. *Final mesh for the Navier–Stokes problem.*

norm-based error estimation. *A posteriori* error estimates for energy or other Sobolev norms typically rely on computing appropriately weighted norms of cell and averaged facet residuals. Hence, the strategy described here provides a starting-point for the automatic generation of norm-based error estimators.

REFERENCES

[1] M. AINSWORTH AND J. T. ODEN, *A Posteriori Error Estimation in Finite Element Analysis*, Wiley and Sons, New York, 2000.

[2] M. S. ALNÆS, *A Compiler Framework for Automatic Linearization and Efficient Discretization of Nonlinear Partial Differential Equations*, PhD thesis, University of Oslo, Unipub, Oslo, Norway, 2009. ISSN 1501-7710, No. 884.

[3] I. BABUSKA AND W. C. RHEINBOLDT, *Error estimates for adaptive finite element computations*, SIAM Journal on Numerical Analysis, (1978), pp. 736–754.

[4] W. BANGERTH AND R. RANNACHER, *Adaptive finite element techniques for the acoustic wave equation*, Journal of Computational Acoustics, 9 (2001), pp. 575–592.

[5] R. E. BANK AND R. K. SMITH, *A posteriori error estimates based on hierarchical bases*, SIAM Journal on Numerical Analysis, (1993), pp. 921–935.

[6] R. E. BANK AND A. WEISER, *Some a posteriori error estimators for elliptic partial differential equations*, Mathematics of Computation, (1985), p. 283301.

[7] R. BECK, R. HIPTMAIR, R. H. W. HOPPE, AND B. WOHLMUTH, *Residual based a posteriori error estimators for eddy current computation*, M2AN Math. Model. Numer. Anal., 34 (2000), pp. 159–182.

[8] R. Becker, V. Heuveline, and R. Rannacher, *An optimal control approach to adaptivity in computational fluid mechanics*, International Journal for Numerical Methods in Fluids, 40 (2002), pp. 105–120.
[9] R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica, 10 (2001), pp. 1–102.
[10] D. Braess and R. Verfürth, *A posteriori error estimators for the Raviart-Thomas element*, SIAM Journal on Numerical Analysis, (1996), pp. 2431–2444.
[11] W. Dörfler, *A convergent adaptive algorithm for Poissons equation*, SIAM Journal on Numerical Analysis, 33 (1996), pp. 1106–1124.
[12] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numerica, 4 (1995), pp. 105–158.
[13] D. Estep and D. French, *Global error control for the continuous Galerkin finite element method for ordinary differential equations*, RAIRO-M2AN Modelisation Math et Analyse Numerique-Mathem Modell Numerical Analysis, 28 (1994), pp. 815–852.
[14] D. J. Estep, M. G. Larson, and R. D. Williams, *Estimating the error of numerical solutions of systems of reaction-diffusion equations*, Amer Mathematical Society, 2000.
[15] R. Falk, *Finite element methods for linear elasticity*, in Mixed Finite Elements, Compatibility conditions and Applications, Springer, 2008.
[16] M. Farhloul and M. Fortin, *Dual hybrid methods for the elasticity and the Stokes problems: a unified approach*, Numer. Math., 76 (1997), pp. 419–440.
[17] V. Heuveline and R. Rannacher, *A posteriori error control for finite element approximations of elliptic eigenvalue problems*, Advances in Computational Mathematics, 15 (2001), pp. 107–138.
[18] J. Hoffman, *On duality based a posteriori error estimation in various norms and linear functionals for LES*, SIAM J. Sci. Comput, 26 (2004), pp. 178–195.
[19] R. C. Kirby, *FIAT: a new paradigm for computing finite element basis functions*, ACM Trans. Math. Software, 30 (2004), p. 502516.
[20] R. C. Kirby and A. Logg, *A compiler for variational forms*, ACM Transactions on Mathematical Software, 32 (2006), pp. 417–444.
[21] ———, *Efficient compilation of a class of variational forms*, ACM Transactions on Mathematical Software, 33 (2007).
[22] M. G. Larson and T. J. Barth, *A posteriori error estimation for discontinuous Galerkin approximations of hyperbolic systems*, Lecture Notes in Comput. Sci. Engrg, 11 (1999), pp. 363–368.
[23] M. G. Larson and F. Bengzon, *Adaptive finite element approximation of multiphysics problems*, Communications in Numerical Methods in Engineering, 24 (2008), pp. 505–521.
[24] M. G. Larson and A. Målqvist, *Adaptive variational multiscale methods based on a posteriori error estimation: duality techniques for elliptic problems*, Multiscale Methods in Science and Engineering, (2005), pp. 181–193.
[25] F. Larsson, P. Hansbo, and K. Runesson, *Strategies for computing goal-oriented a posteriori error measures in non-linear elasticity*, International Journal for Numerical Methods in Engineering, 55 (2002), pp. 879–894.
[26] A. Logg, *Automating the finite element method*, Arch. Comput. Methods Eng., 14 (2007), p. 93138.
[27] A. Logg, K.-A. Mardal, G. N. Wells, et al., *Automated Solution of Differential Equations by the Finite Element Method*, Springer, 2012.
[28] A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells, *FFC: the FEniCS Form Compiler*, Springer, 2012, ch. 11.
[29] A. Logg and G. N. Wells, *DOLFIN: Automated finite element computing*, ACM Transactions on Mathematical Software, 37 (2010), pp. 20:1–20:28.
[30] M. Lonsing and R. Verfürth, *A posteriori error estimators for mixed finite element methods in linear elasticity*, Numerische Mathematik, 97 (2004), pp. 757–778.
[31] R. H. Nochetto, A. Veeser, and M. Verani, *A safeguarded dual weighted residual method*, IMA Journal of Numerical Analysis, 29 (2008), pp. 126–140.
[32] R. Rannacher and F. T. Suttmeier, *A posteriori error control in finite element methods via duality techniques: Application to perfect plasticity*, Computational mechanics, 21 (1998), pp. 123–133.
[33] S. Richling, E. Meinköhn, N. Kryzhevoi, and G. Kanschat, *Radiative transfer with finite elements*, Astronomy and Astrophysics, 380 (2001), pp. 776–788.
[34] M. E. Rognes and A. Logg, *Efficient implementation of automated goal-oriented error control for stationary variational problems*, (2012). In preparation.
[35] R. Sandboge, *Adaptive finite element methods for systems of reaction-diffusion equations*,

Computer Methods in Applied Mechanics and Engineering, 166 (1998), pp. 309–328.
[36] ———, *Adaptive finite element methods for reactive compressible flow*, Mathematical Models and Methods in Applied Sciences, 9 (1999), pp. 211–242.
[37] R. Verfürth, *A posteriori error estimators for the Stokes equations*, Numerische Mathematik, 55 (1989), pp. 309–325.
[38] ———, *A review of a posteriori error estimation techniques for elasticity problems*, Computer Methods in Applied Mechanics and Engineering, 176 (1999), p. 419440.
[39] A. Wahlberg, *Evaluation and comparison of Duality-Based a posteriori error estimates*, Master's thesis, Lund University, Technical Faculty, 2009.