Adaptive Finite Element Methods with Convergence Rates *

Peter Binev, Wolfgang Dahmen, Ron DeVore

June 14, 2002

Abstract

Adaptive Finite Element Methods for numerically solving elliptic equations are used often in practice. Only recently [10], [14] have these methods been shown to converge. However, this convergence analysis says nothing about the rates of convergence of these methods and therefore does not yet, in principle, describe yet any numerical advantages of adaptive strategies versus non-adaptive strategies. The present paper modifies the adaptive method of Morin, Nochetto, and Siebert [14] for solving the Laplace equation with piecewise linear elements on domains in $I\!R^2$ by adding a coarsening step and proves that this new method has certain optimal convergence rates in the energy norm (which is equivalent to the H^1 norm). Namely, it is shown that whenever $s > 0$ and the solution u is such that for each $n \geq 1$, it can be approximated to accuracy $O(n^{-s})$ in the energy norm by a continuous, piecewise linear function on a triangulation with n cells (using complete knowledge of u), then the adaptive algorithm constructs an approximation of the same type with the same asymptotic accuracy while using only information gained during the computational process. Moreover, the number of arithmetic computations in the proposed method is also of order $O(n)$ for each $n \geq 1$. The construction and analysis of this adaptive method relies on the theory of nonlinear approximation.

Key Words: elliptic equations, finite element methods, adaptive refinements, rates of convergence, nonlinear approximation.

1 Introduction

Adaptive methods are frequently used to numerically compute solutions to elliptic equations. While these methods have been shown to be very successful computationally, the theory describing the advantages of such methods over their non-adaptive counterparts is still not complete. For example, only recently [10], [14] have there even been proofs of

*This work has been supported by the Office of Naval Research Contract Nr. N0014-91-J1343 , the Army Research Office Contract Nr. DAAD 19-02-1-0028, the National Science Foundation Grants DMS0221642, the Deutsche Forschungsgemeinschaft grant SFB 401, the TMR Network "Wavelets in Numerical Simulation" funded by the European Commission and the Humboldt Foundation.

convergence of such methods. These proofs of convergence still do not prove any guaranteed advantage of these adaptive methods since there is no analysis of their *rate of convergence* in terms of the number of degrees of freedom or the number of computations.

Recently, an analysis of rates of convergence for wavelet based adaptive methods was given in [5],[6]. These papers derive an adaptive wavelet based algorithm for solving elliptic problems and show that this algorithm has optimal efficiency in the sense that if the solution u can be approximated (using complete knowledge of u) in the energy norm by an n-term wavelet expansion to accuracy $O(n^{-s})$, $n \to \infty$, then the adaptive method will do the same using *only* knowledge of u gained through the adaptive iteration. Wavelet methods vary from their FEM counterparts in that they can be viewed as solving linear systems that are finite sections of one fixed infinite dimensional matrix problem whose solution gives the wavelet coefficients of u.

The theoretical foundation of Adaptive Finite Element Methods (AFEM) is less satisfying. There is no known algorithm with a proven rate of convergence save for the univariate case [3]. The purpose of the present paper is to give an AFEM and prove convergence rates for this method which are the analogue of the wavelet case. Our algorithm is not much different from existing adaptive methods based on bulk chasing of a posteriori error estimators. The one main difference is the utilization of a coarsening strategy. We should mention that coarsening also played an important role in the analysis of adaptive wavelet methods. However, in the practical implementation of the adaptive wavelet methods, for many problems, coarsening is not needed. The same may be the case for AFEM.

We primarily view the present paper as a contribution to the theoretical analysis of AFEM rather than the construction of an adaptive method that outperforms other adaptive methods in practice. In particular, we wish to clarify whether a-posteriori information can lead to an adaptive algorithm that exhibits asymptotically optimal performance. In spite of the theoretical emphasis it should not be excluded that some of the ideas of the present paper may be useful in practice. One of these tools is the theory of nonlinear approximation by piecewise polynomials. Since adaptive methods are a form of nonlinear approximation, this theory will on the one hand help us to provide a benchmark for measuring the success of adaptive methods, and on the other hand, provide an effective implementation for the coarsening (see §4.5).

Adaptive Finite Element Methods have several complications that make their analysis more cumbersome. These include the need for graded meshes, the problem of hanging nodes, and the analysis of a-posteriori error estimators. If not for these complications, the analysis in this paper would be considerably simplified. In order to present the ideas of this paper in their simplest form, we shall try to minimize these obstacles. In particular, we shall restrict ourselves to the Poisson problem

$$-\Delta u = f \quad \text{in} \quad \Omega, \quad u = 0 \quad \text{on} \quad \partial\Omega, \tag{1.1}$$

where Ω is a polygonal domain in $\mathbb{R}^2$ and $\partial\Omega$ is its boundary. We shall also consider only approximations of the solution u by piecewise linear elements using a very specific adaptive refinement strategy (called newest vertex bisection) well-known in the FEM literature. In this way, the essentials of our arguments will be clear and we can also call

on several known results concerning a-posteriori error estimates that can be found in the literature.

We conclude this introduction by briefly describing the structure of this paper. In §2 we discuss the general form of adaptive Finite Element Methods which is marking, subdivision, and completion (to remove hanging nodes). We then consider in detail the structure of subdivision using newest vertex bisection. We shall introduce a simple labelling for edges that will facilitate the analysis of this type of subdivision. The main result of this section is Theorem 2.4 which bounds the number of cells in the completion process by the number of marked cells. This bound is vital in proving optimal convergence rates. §3 recalls Galerkin approximations.

In §4 we study adaptive approximation by piecewise linear functions on adaptively generated triangulations. The spirit of this section is to understand how to construct good adaptive approximations to a *known* function w. In particular, we introduce the algorithm of [2] which will be used heavily in our adaptive Finite Element Algorithm for solving (1.1). Namely, it is used to approximate the right hand side f (see §4.4) and for our coarsening strategy (see §4.5). We also discuss in this section what are optimal approximation rates to a known function. This will provide a benchmark for our analysis of adaptive Finite Element Algorithms.

In §5, we recall the adaptive FEM method in [14] and record some of the proven facts about this method that will be used in the present paper. This includes their introduction of local a-posteriori error estimators and their analysis of how these estimators can be used to bound global errors. In §6, we make some minor modifications of the **MNS** algorithm.

In §7, we describe the main ingredients of our new adaptive algorithm. We show its optimal rates of convergence in §8. We conclude the paper with an appendix which discusses the smoothness conditions that govern rates of convergence by adaptive methods. These results are not important for the analysis in the present paper but may be of interest to the reader.

2 Newest vertex bisection and completion

This section has three purposes. The first is to set forward some of the notation we shall use in this paper. The second is to introduce the main form of adaptive Finite Element Methods which is marking, subdividing, and completing. The third is to introduce and analyze the particular form of subdivision we shall use in this paper, the so called *newest vertex bisection* method.

Let Ω be a polygonal domain in $\mathbb{R}^2$. We shall use P to denote a partition of Ω into triangular cells Δ. This means that $\Omega = \cup_{\Delta\in P}\Delta$ and any two $\Delta, \Delta' \in P$ satisfy $\text{meas}(\Delta\cap\Delta') = 0$ where here and later in this paper meas denotes the Euclidean measure in $\mathbb{R}^2$. Given such a partition, we let $\mathcal{S}_P$ denote the space of continuous, piecewise linear functions subordinate to P which vanish on $\partial\Omega$. A function S is in $\mathcal{S}_P$ if and only if S is a linear function on each $\Delta \in P$, S is continuous on Ω, and S vanishes on $\partial\Omega$.

We denote by $\mathcal{E}_P$ the set of edges of P and by $\dot{\mathcal{E}}_P$ the set of *interior edges*. Thus, $E \in \dot{\mathcal{E}}_P$ means that E is an edge of some $\Delta \in P$ and that the interior of E is in the

interior of Ω. All other edges are called *boundary edges*. We also denote by $\mathcal{V}_P$ the set of all vertices v of P and by $\dot{\mathcal{V}}_P$ the set of *interior vertices*. Thus, $v \in \dot{\mathcal{V}}_P$ means that v is a vertex of one of the $\Delta \in P$ and v is in the interior of Ω. All other vertices are called *boundary vertices*.

There are two special conditions that we shall impose on a partition P that are important in Finite Element constructions. First, we say that a partition satisfies a *minimal angle condition* if for each $\Delta \in P$ all of its angles are $\geq a_0$ for some positive number a_0. Second, we shall require a partition P to be *conforming* which means that the intersection of any two cells is either empty or a common edge or a common vertex. A *family* of partitions whose elements are all conforming and which satisfy a minimal angle condition with respect to a common constant $a_0 > 0$, are called *admissible*.

The uniform minimal angle condition implies that the ratio of the radii of the smallest circumscribed and the largest inscribed circle for each cell in any partition P from an admissible family $\mathcal{P}$ are uniformly bounded. This is sometimes referred to as *local quasi-uniformity* or *shape regularity*. In particular, this implies the existence of a constant $\hat{C} = \hat{C}(\mathcal{P})$ such that

$$1 \leq \operatorname{diam}(\Delta)^2/|\Delta| \leq \hat{C} \quad \text{for all} \quad \Delta \in P,\ P \in \mathcal{P}, \tag{2.1}$$

where $|\Delta| = \operatorname{meas}(\Delta)$ denotes the Lebesgue measure of Δ. Moreover, there exists a constant $G_0 = G_0(\mathcal{P})$ such that for any two cells $\Delta, \Delta' \in P$, for which $\Delta \cap \Delta' \neq \emptyset$, we have

$$\operatorname{diam}(\Delta) \leq G_0 \operatorname{diam}(\Delta'). \tag{2.2}$$

Locally quasi-uniform partitions allow one to control the global error in approximating a function by local errors. Typical estimates in FEM depend on a_0 and deteriorate if a_0 is small.

Conformity favors common finite element data structures by conveniently relating local and global stiffness matrices since global basis functions are composed of the local shape functions on each element in a simple way. In our specific context the global basis functions will be the Courant "hat functions" (nodal functions) ϕ_v, $v \in \mathcal{V}_P$. The function ϕ_v is the unique element in $\mathcal{S}_P$ which is one at v and is zero at all other vertices in $\mathcal{V}_P$. The nodal basis functions are locally supported on the union of all triangles which share v as a vertex. Moreover, one can construct locally supported dual functionals with the same supports consisting of (discontinuous) piecewise linear functions thereby giving rise to local linear projectors on $\mathcal{S}_P$ that are bounded in $H^s(\Omega)$ for $s \leq 1$ (in fact even beyond 1). Again, it is important for estimating errors that for an partition P from an admissible family, each basis functions is overlapped by a uniformly bounded number of other basis functions since the *valence* of the vertices, i.e. the number of edges emanating from a given vertex, remains uniformly bounded.

The adaptive procedures we shall consider in this paper will generate a family of partitions which is admissible.

A typical AFEM generates a sequence of partition $P_0, P_1, \ldots, P_n$ by using rules for subdividing triangles. Given the partition P_k, the algorithm marks certain of the triangular cells $\Delta \in P_k$ for subdivision. We shall denote by $\mathcal{M}_k$ the collection of marked cells. These marked cells are subdivided using certain subdivision rules. This process, however,

creates *hanging nodes*. We say that $v \in \mathcal{V}_P$ is a hanging node for $\Delta \in P$ if v appears in the interior of one of the sides of Δ. Since hanging nodes obviously violate conformity, in a second step a certain collection $\mathcal{M}'_k$ of additional cells are subdivided in order to guarantee that the resulting partition P_{k+1} is admissible. The partition P_n is the final admissible partition associated with this application of the adaptive algorithm.

2.1 Newest vertex bisection

We shall restrict ourselves in this paper to a very specific method of subdivision known as *newest vertex bisection*. We shall call on certain properties of this method of subdivision in what follows. We could not find some of these properties (or proofs of these properties) in the FEM literature and therefore our discussion and development of newest vertex insertion will be somewhat lengthy. The book of Verfürth [17] and the research article of Mitchell [13] describe this subdivision method and give some of its properties.

Given an initial partition P_0 of Ω, to each $\Delta \in P_0$, we assign exactly one of its vertices $v(\Delta)$ and call it the *newest vertex* for that cell. This initial labeling can be made in an arbitrary way. The edge in Δ opposite to $v(\Delta)$ will be denoted by $E(\Delta)$. Each triangular cell that arises in the adaptive process will also have exactly one of its vertices designated as a newest vertex. If this cell is to be subdivided then the subdivision is a simple bisection of the newest vertex and the side $E(\Delta)$ opposite. Thus the cell produces two new cells and their newest vertex (assigned to each new triangular cell) is by definition the midpoint of $E(\Delta)$.

The partitions which arise when using newest vertex insertion satisfy a uniform minimal angle condition. This is established by showing that all triangles that arise in newest vertex insertion can be classified into a set of similarity classes depending only on the initial partition P_0 (see Mitchell [13]). Also note that if a partition P is created by a sequence of newest vertex insertions and if P has no hanging nodes, then it is conforming. Thus, it is admissible. We shall show in the next subsection how any given partition generated by newest vertex insertion can be completed to a partition with no hanging nodes by subdividing certain other triangular cells. This process is called *completion*. Furthermore, we shall bound the number of additional subdivisions necessary to remove hanging nodes. But first we want to examine another important property of newest vertex insertion which is its tree structure.

We can represent newest vertex bisection subdivision by an infinite binary tree T_* (which we call the *master tree*). The master tree T_* consists of all triangular cells which can be obtained by a sequence of subdivisions. The roots of the master tree are the triangular cells in P_0. When a cell Δ is subdivided, it produces two new cells which are called the children of Δ and Δ is their parent. It is very important to note that, no matter how a cell arises in a subdivision process, its associated newest vertex is unique and only depends on the initial assignment of newest vertices in P_0. This means that the children of Δ are uniquely determined and do not depend on how Δ arose in the subdivision process, i.e., it does not depend on the preceding sequence of subdivisions. The reason for this is that any subdivision only assigns newest vertices for the new triangular cells produced by the subdivision and does not alter any previous assignment. It follows that T_* is unique and does not depend at all on the order of subdivisions.

The *generation* of a triangular cell Δ is the number $g(\Delta)$ of ancestors it has in the master tree. Thus cells in P_0 have generation 0, their children have generation 1 and so on. The generation of a cell is also the number of subdivisions necessary to create this cell from its corresponding root cell in P_0.

A *subtree* $T \subset T_*$ is a collection of triangular cells $\Delta \in T_*$ with the following two properties: (i) whenever $\Delta \in T$ then its sibling is also in the tree; (ii) when $\Delta \subset \Delta'$ are both in the tree then each triangular cell $\bar{\Delta} \in T_*$ with $\Delta \subset \bar{\Delta} \subset \Delta'$ is also in T. The roots of T are all the cells $\Delta \in T$ whose parents are not in T. We say that T is *proper* if it has the same roots as T_*, i.e., it contains all $\Delta \in P_0$.

If $T \subset T_*$ is a finite subtree, we say $\Delta \in T$ is a *leaf* of T if T contains none of the children of Δ. We denote by $\mathcal{L}(T)$ the collection of all leaves of T.

For a proper subtree T, we define $N(T)$ to be the number of subdivisions made to produce T.

Any partition $P = P_n$ which is obtained by the application of an adaptive procedure based on newest vertex insertion (such as the algorithms we consider in this paper) can be associated to a proper subtree $T = T(P)$ of T_* consisting of all triangular cells that were created during the algorithm, i.e. all of the cells in $P_0, \ldots, P_n$. The set of leaves $\mathcal{L}(T)$ form the final partition $P = P_n$.

We shall say that $T = T(P)$ is admissible if P is admissible. We denote the class of all proper trees by $\mathcal{T}$ and all admissible trees by $\mathcal{T}^a$. We also let $\mathcal{T}_n$ be the set of all proper trees T with $N(T) = n$ and by $\mathcal{T}_n^a$ the corresponding class of admissible trees from $\mathcal{T}_n$. We denote by $\mathcal{P}$ the class of all partitions P that can be generated by newest vertex insertion and by $\mathcal{P}^a$ the set of all admissible partitions. Similarly, $\mathcal{P}_n$ and $\mathcal{P}_n^a$ are the subclasses of those partitions that have at most n cells. There is a precise identification between $\mathcal{P}_n$ and $\mathcal{T}_n$. Any $P \in \mathcal{P}_n$ can be given by a tree, i.e. $P = P(T)$ for some $T \in \mathcal{T}_n$. Conversely any $T \in \mathcal{T}_n$ determines a $P = P(T_n)$ in $\mathcal{P}_n$. The same can be said about admissible partitions and trees.

2.2 Completion of subdivision

The adaptive algorithms we consider in this paper will be of the following type. We begin with P_0 and mark certain cells in P_0 for subdivision. After doing these subdivisions we arrive at the partition P_1'. This partition is not necessarily admissible and so we shall make some additional subdivisions which will complete P_1' to an admissible partition P_1. We then repeat this process of marking and completing. It will be important for us to see that the completion process does not seriously inflate the number of triangular cells in P_n. We have not found any result in the literature saying that the overall number of triangles created through completion stays always proportional to the number of marked cells throughout the refinement process. To establish this will be a bit technical and will be the subject of this subsection.

Suppose that P is an admissible partition with $\#(P) > 2$ (the case $\#(P) = 2$ is trivial in what follows). To each $\Delta \in P$, we associate a triangular cell $F(\Delta) \in P$ as follows. Let $v(\Delta)$ be the newest vertex of Δ and $E(\Delta)$ the edge of Δ opposite to $v(\Delta)$. If $E(\Delta)$ is a boundary edge then we define $F(\Delta) = \emptyset$. Otherwise, there is a unique triangular cell $\Delta' \neq \Delta$ which has $E(\Delta)$ as one of its edges and we define $F(\Delta) = \Delta'$. One can visualize

the mapping $\Delta \to F(\Delta)$ as a flow determined by the vector which serves to bisect Δ in the subdivision process.

By a *chain* $C(\Delta)$ (with starting cell Δ) in P, we mean a sequence $\Delta, F(\Delta), \ldots, F^m(\Delta)$ with no repetition of the cells in this chain and with $F^{m+1}(\Delta) = F^k(\Delta)$, for some $k \in \{0, \ldots, m-1\}$ or $F^{m+1}(\Delta) = \emptyset$.

Remark: *We shall see below that by starting with a particular assignment of newest vertices in P_0, for any of the subsequent partitions $P = P_k$, the only way $F^{m+1}(\Delta) = F^k(\Delta)$ is for k to be equal to $m-1$. We shall therefore assume this property in going further.*

The completion of a chain $C(\Delta)$ is a collection $\bar{C}(\Delta)$ of cells produced by two sets of subdivisions. In the first set, each cell $\Delta' = F^k(\Delta)$ in this chain is subdivided using the newest vertex subdivision (i.e. the insertion of the diagonal connecting $v(\Delta')$ to the midpoint on $E(\Delta')$). This subdivision of Δ' produces two new cells (the children of Δ').

After this first set of subdivisions has been completed, there will generally be cells with hanging nodes. The second part of the subdivision process is to subdivide each of the children that have a hanging node. Hanging nodes occur inside a cell $\Delta' = F^k(\Delta)$, when $E(F^{k-1}(\Delta)) \neq E(F^k(\Delta))$. In this case the new edge we need to add in $F^k(\Delta)$ is the one connecting the midpoints of these two edges. This part of the subdivision process removes all remaining hanging nodes. By the above remarks, this has the effect of subdividing for $k = 1, \ldots, m'$ that child of $F^k(\Delta)$ into two grandchildren which has $E(F^{k-1}(\Delta))$ as an edge. Here $m' = m-1$, when $F^{m+1}(\Delta) = F^{m-1}(\Delta)$, and $m' = m$, when $F^{m+1}(\Delta) = \emptyset$.

We shall make some further observations about the structure of $\bar{C}(\Delta)$ and the resulting flow structure. For this purpose, we shall introduce a way of labelling all edges that arise in the subdivision process.

We shall label the edges in the partitions $P_0, P_1, \ldots, P_n$ by nonnegative integers. This labelling will give us a simple way to keep track of the subdivision and completion process. Given any triangular cell Δ in one of these partitions, the sides of Δ will be labelled by $(i+1, i+1, i)$ where $i = g(\Delta)$ is the nonnegative integer that represents the generation of Δ (i.e. how many subdivisions of a cell in P_0 were needed to create Δ). The labelling will be such that the lowest labelled side will be $E(\Delta)$, i.e. the side opposite the newest vertex of Δ. At the outset, it may appear that the labelling of a side will depend on the triangle Δ and so a side will get two labels depending on which triangle we view it to be in. However, as we shall see, for admissible partitions the labelling of an edge can be *independent* of the triangle to which it belongs provided we start with a suitable labelling of P_0.

To start the labelling process, we describe how to label the edges in P_0. We begin with the following lemma.

Lemma 2.1 *For any initial partition P_0 there is a labelling of the edges in P_0 such that each edge is given a label of either 0 or 1 and whenever a triangle $\Delta \in P_0$ then exactly two of its edges are labeled with a 1 and the other edge is labeled with a 0.*

Proof: For any collection A of cells we denote by Ω_A their union. Suppose we could find a set $Q \subseteq P_0$ with the following properties:
(i) All triangles in $P_0 \setminus Q$ have at least one edge on the boundary of Ω. In brief, $P_0 \setminus Q$ consists only of boundary triangles.

(ii) The domain Ω_Q can be decomposed into an essentially disjoint union of quadrilaterals formed by pairs of adjacent triangles from Q.

Now, given any collection $Q \subseteq P_0$ with the above properties (i), (ii), we can assign the following labels to the edges. Given a pair of triangles from Q whose union forms one of the quadrilaterals, we assign 0 to their common edge and 1 to all other edges. By (i) we have missed at most edges on the boundary Γ of Ω. If such an edge E belongs to a triangle with two interior edges, they must have the label 1, so we assign the label 0 to E. If the edge belongs to a triangle with two edges on the boundary we label one by 0 and the other one by 1 to obtain the type of labelling asserted by the lemma.

Thus the question is, can one always find collections Q in P_0 satisfying (i), (ii). This question can be transformed into a question about so called matchings in cubic graphs. Recall that a cubic graph is one in which every vertex has exactly three edges. Consider the *dual graph* of P_0 (whose vertices are the triangles and whose edges between two vertices are the edges shared by the corresponding triangles). This graph is not yet a cubic graph but it can be completed to a cubic graph as follows. We map the triangulation P_0 to the upper hemisphere of S^3 (the unit sphere in $\mathbb{R}^3$) and then complete this to a triangulation of all of S^3. The dual graph of this triangulation is a cubic graph (meaning that exactly three edges are incident at every vertex). Any set of edges in a graph where no two of them share the same vertex is called a matching. The matching is perfect if all vertices are covered. Thus our problem becomes the question of the existence of perfect matchings in cubic graphs which indeed has been established by Petersen's Theorem from [15], see also Theorem 3.4.1 on page 110 in [12].

In order to keep the present exposition selfcontained we sketch a proof of the existence of suitable collections Q (with the properties (i) and (ii) above) which, in particular, indicates how to generate a suitable labelling in practice.

To this end, extend the triangulation P_0 to a triangulation $\hat{P}$ of all of $\mathbb{R}^2$ in such a way that the side lengths of all edges and all angles stay bounded away from zero. It suffices to find a collection $\hat{Q}$ of quadrilaterals $\diamond$, formed by adjacent pairs of triangles in $\hat{P}$ such that $\Omega \subseteq \Omega_{\hat{Q}}$. In fact, then the set $Q := \{\Delta \in P_0 : \Delta \subset \diamond,\ \diamond \in \hat{Q},\ \diamond \subset \Omega\}$ satisfies (i) and (ii).

To construct $\hat{Q}$ as above, fix any vertex v_0 of P_0 and enumerate all vertices that are connected with v_0 by an edge as $v_1, \ldots, v_m$, in such a way that the closed polygon connecting $v_1, \ldots, v_m$ and back to v_1 is traversed in a clockwise fashion. Then, when $m = 2k$ is even, form the set Q_1 of quadrilaterals $\diamond_i = [v_0, v_{2i+1}, v_{2i+2}, v_{2i+3}]$, $i = 0, \ldots, k-1$. By construction, Ω_{Q_1} contains all triangles sharing v_0 as a vertex and hence is a simply connected polygonal domain. If $m = 2k+1$ is odd, we form the quadrilaterals $\diamond_i$ as above but complete the remaining triangle $[v_0, v_{2k+1}, v_1]$ by pairing it with another triangle $\Delta' = [v_{2k+1}, v_1, w]$ to a quadrilateral $\diamond_k$, label w as v_{m+1} and set this time $Q_1 := \{\diamond_i : i = 0, \ldots, k\}$. Again Ω_{Q_1} is a simply connected polygonal domain which is a disjoint union of quadrilaterals and whose boundary Γ_1 consists of the edges connecting the vertices $v_1, \ldots, v_{m_1}, v_1$, where $m_1 = m$ when m is even and $m_1 = m+1$ when m is odd.

Now suppose we are given a simply connected polygonal domain $\Omega_l = \Omega_{Q_l}$ which is decomposed into the quadrilaterals from Q_l with boundary polygon Γ_l connecting the vertices $v_{m_{l-1}+1}, \ldots, v_{m_l}$ numbered clockwise. It suffices to show how to grow Ω_l to a polygonal domain Ω_{l+1} with exactly the same properties. We set $\tilde{\Omega}_{l+1} = \bigcup\{\Delta \in \hat{P} :$

$\Delta \cap \Gamma_l \neq \emptyset\} \cup \Omega_l$. Thus, in a first step, we expand Ω_l by a layer L_{l+1} of triangles sharing at least one vertex with Γ_l. We order vertices on the new boundary $\partial\tilde{\Omega}_{l+1}$ as $v_{m_l+1}, \ldots, v_{m_l+r}$ in a clockwise fashion. We also order the triangles in L_{l+1} in a clockwise fashion. Every triangle in L_{l+1} has either one edge on Γ_l and one vertex on $\partial\tilde{\Omega}_{l+1}$ or vice versa. Thus Δ' is on the right of Δ if the vertices of Δ' are on the right of those of Δ with respect to Γ_l and $\partial\tilde{\Omega}_{l+1}$. When $\#(L_{l+1})$ is even we set $\Omega_{l+1} := \tilde{\Omega}_{l+1}$, $\Gamma_{l+1} := \partial\tilde{\Omega}_{l+1}$, $m_{l+1} := m_l + r$. We can now decompose the layer L_{l+1} into quadrilaterals formed by pairs of adjacent triangles by simply grouping two successive ones together. If $\#(L_{l+1})$ is odd we start with a triangle $\Delta_{1,1} \in L_{l+1}$ whose left neighbor $\tilde{\Delta}$ has only one vertex in common with Γ_l. There must exist such a triangle. When forming quadrilaterals from the triangles in L_{l+1} as before by grouping two successive ones together, we miss exactly the triangle $\tilde{\Delta}$, which has only one vertex on Γ_l. Therefore it has an edge E on $\partial\tilde{\Omega}_{l+1}$. Let $\tilde{\Delta}'$ be the triangle in $\hat{P}$ which shares E with $\tilde{\Delta}$. Since $\tilde{\Delta}'$ can have no other edge on $\partial\tilde{\Omega}_{l+1}$, the set $\Omega_{l+1} := \tilde{\Omega}_{l+1} \cup \tilde{\Delta}'$ is still simply connected and can be decomposed into quadrilaterals formed by pairs of adjacent triangles contained in Ω_{l+1}. This advances the induction so that we can continue expanding as above forming a sequence of domains Ω_n. Since the elements of P_0 (and hence of $\hat{P}$ have diameters bounded away from zero, the domains Ω_n contain circles about v_0 of strictly increasing diameter. Thus after finitely many steps n_0 we have $\Omega = \Omega_{P_0} \subseteq \Omega_{n_0}$ which finishes the proof. $\ominus$

Given the labelling of sides in P_0 by Lemma 2.1, we define the newest vertex of a triangular cell $\Delta \in P_0$ to be the vertex opposite the side which is labeled by 0.

In going further, we shall always assume that the initial labelling of newest vertices in P_0 has been done in accordance with Lemma 2.1. Notice that this means that any chain in P_0 has at most two cells and that the subdivision of these cells gives an admissible partition (i.e. there is no need to go to the second subdivisions which generated grandchildren).

We now give a rule to label any edges that arise from the subdivision-completion process. There will be two main properties of this labelling. The first is that each triangular cell will have sides with labels $(i, i, i-1)$ for some positive integer i. The second is that the newest vertex for this cell will be the vertex opposite the side with lowest label. Certainly the edges in P_0 have such a labelling as we have just shown.

Suppose that we have such a labelling for the edges in P_k and let us describe how to label the edges in P_{k+1}. Suppose that a triangular cell Δ has sides which have been labelled $(i, i, i-1)$ and the newest vertex for this cell is the one opposite the side labelled $i-1$. When this cell is subdivided (using newest vertex bisection) the side labelled $i-1$ is bisected and we label each of the two new sides $i+1$. We also label the *bisector* by $i+1$, i.e. the new edge connecting the newest vertex of Δ with the midpoint of the edge $E(\Delta)$ labelled by $i-1$. Thus each new triangle now has sides labelled $(i, i+1, i+1)$ with the newest vertex opposite the side with the lowest label. We note the important fact that if a cell has label $(i+1, i+1, i)$ then it is of generation i (i.e. it has been obtained from a cell in P_0 by i subdivisions). Therefore, specifying that the generation of the cell is i is the same as specifying that its label is $(i+1, i+1, i)$.

Lemma 2.2 *Suppose P_0 is an arbitrary partition and its edges and newest vertices are labeled in accordance with Lemma 2.1. Suppose that $P_1, \ldots, P_n$ are partitions which are*

generated from P_0 using the marking, subdivision, and completion process. We label edges in $P_1, \ldots, P_n$ as described above. Then there holds for each $k = 0, 1, \ldots, n$:
(i) each edge in P_k has a unique label independent of the two triangles which share this edge.
(ii) If Δ is a triangular cell in P_k of generation $g(\Delta) = i$, i.e. the edge with label i is the side shared by Δ and $F(\Delta)$, then $g(F(\Delta)) \in \{i, i-1\}$. If $g(F(\Delta)) = i$ the flow ends at $F(\Delta)$.
(iii) for any $\Delta \in P_k$ of generation $g(\Delta) = i$, the cells in its chain $C(\Delta) = \{\Delta, F(\Delta), \ldots, F^m(\Delta)\}$ have the property that $g(F^j(\Delta)) = i - j$, $j = 0, \ldots, m-1$, and the terminating cell $F^m(\Delta)$ for this chain is either of generation $i - m + 1$ or it is a boundary cell with lowest labelled edge an edge of the boundary.

Proof: This is proved by induction on k. All three assertions are clear for $k = 0$ by the construction of the labelling of P_0 given by Lemma 2.1. Suppose that we have proven the lemma for P_{k-1} and consider P_k.
Proof of (i): Note that the above rule leave the labels of all those edges unchanged that are not affected by subdivisions. Any edge E in P_k which was not in P_{k-1} was obtained in one of two ways. The first is that it is a new edge which was added as a bisector in the subdivision-completion process. In this case there is nothing to prove about its labelling being unique. The second possibility is that the new edge E was obtained by bisecting an edge E', say with label i, from P_{k-1}. Let Δ and Δ' be the two triangular cells in P_{k-1} which shared E'. For one of these triangular cells, which we can assume is Δ', we have $E' = E(\Delta')$. So Δ' had label $(i+1, i+1, i)$ and therefore viewed from Δ', E is assigned the label $i + 2$. By induction assumption, as an edge of Δ, E' is also labelled by i, so that Δ has either the label $(i, i+1, i+1)$ or $(i, i, i-1)$. In the first case the situation is symmetric to Δ' since $E' = E(\Delta)$ is bisected and E as one of the halves is labelled $i + 2$. In the second case E' is still an edge of one of the children of Δ obtained by bisecting the edge with label $i - 1$. Thus the midpoint of E' is still a hanging node in that child which is now labelled $(i+1, i+1, i)$. This is the situation described by the first case which finishes the proof of (i).
Proof of (ii): Since Δ has generation i, $E(\Delta)$ has label i. Since this edge is shared with $F(\Delta)$, we conclude that the sides of $F(\Delta)$ have either labels $(i, i, i-1)$ or $(i+1, i+1, i)$. Also, in this latter case $F^2(\Delta) = \Delta$ and the flow ends at $F(\Delta)$.
Proof of (iii) This follows from (ii). $\ominus$

Note that all admissible partitions, generated by newest vertex bisection based on an initial labelling according to Lemma 2.1, are *graded* in the sense that any two cells sharing an edge differ in generation by at most one.

We shall next give a bound for the number of cells in P_n. In preparation for this, let us note that there are constants c_1, C_1 depending only on P_0 such that for each Δ of generation i, we have $c_1 2^{-i} \leq |\Delta| \leq C_1 2^{-i}$ where $|\Delta| = \text{meas}(\Delta)$ is the area of Δ. Indeed, each subdivision of a cell gives two cells with each having half the area of the original cell. By adjusting the constants if necessary, we also infer from (2.1) that

$$c_1 2^{-i/2} \leq \text{diam}(\Delta) \leq C_1 2^{-i/2}, \quad g(\Delta) = i, \tag{2.3}$$

because P_n belongs to an admissible family with parameters $\hat{C}, G_0, a_0$ depending only on the initial partition P_0.[1]

Let $\mathcal{M} := \cup_{j=0}^{n-1} \mathcal{M}_j$ be the collection of all cells that were marked in going from P_0 to P_n. Here is the way to view the following argument that will bound the number of cells in P_n. We give each cell $\Delta' \in \mathcal{M}$ a fixed number $C > 0$ of dollars to spend. These cells will spend these dollars such that each new cell that was created in going from P_0 to P_n will get at least $c > 0$ dollars where c is an absolute constant. This means, we can bound the number of new cells created by the number of marked cells.

We now describe how a cell $\Delta' \in \mathcal{M}$ will spend its money. We define

$$A := C_1 \sum_{j=-1}^{\infty} 2^{-j/4} \tag{2.4}$$

where C_1 is the constant in (2.3). We define now a function $\lambda : P_n \times \mathcal{M} \to I\!R$ as follows:

$$\lambda(\Delta, \Delta') := \begin{cases} (j-k+2)^{-2}, & \text{if } g(\Delta') = j,\ g(\Delta) = k,\ \operatorname{dist}(\Delta, \Delta') \le A2^{-k/2} \\ & \text{and } k \le j+1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

The quantity $\lambda(\Delta, \Delta')$ is the portion of money which is spent by the marked cell Δ' on nearby cells $\Delta \in P_n$ of generation at most $g(\Delta') + 1$. Given $\Delta' \in \mathcal{M}$ with $g(\Delta') = j$, there are for any $k \le j+1$ at most C' cells $\Delta \in P_n$ of generation $g(\Delta) = k$ which satisfy $\operatorname{dist}(\Delta, \Delta') \le A2^{-k/2}$ (see (2.3)), where C' is an absolute constant. Hence for any such $\Delta' \in \mathcal{M}$ one has

$$\sum_{\Delta \in P_n} \lambda(\Delta, \Delta') \le C' \sum_{\nu=1}^{\infty} \nu^{-2} = C \tag{2.6}$$

with C an absolute constant. It follows then from (2.6) that

$$\sum_{\Delta' \in \mathcal{M}} \sum_{\Delta \in P_n} \lambda(\Delta, \Delta') \le C\#(\mathcal{M}), \tag{2.7}$$

i.e. the total amount of money spent by all the marked cells is proportional to their number. Conversely, each cell in P_n receives at a least minimum share bounded away from zero as will be shown next.

Lemma 2.3 *For any $\Delta \in P_n \setminus P_0$ we have*

$$\sum_{\Delta' \in \mathcal{M}} \lambda(\Delta, \Delta') \ge c \tag{2.8}$$

where $c > 0$ is an absolute constant.

[1] Generic constants whose value may vary on each occurrence will be denoted by C. Whenever the specific value of a constant matters we shall use subscripts. Using the same subscript for different constants indicates that they occur in the same type of estimates. It is then understood that they take the maximum value of the constants arising in this common context.

Proof: We fix Δ and let $k := g(\Delta) \geq 1$ be its generation. We are going to define a sequence of marked cells $\Delta_1, \ldots, \Delta_m \in \mathcal{M}$ associated to Δ with each Δ_ν of generation $\geq k-1$. Δ_1 is the marked cell such that $\Delta \in \bar{C}(\Delta_1)$, i.e. Δ arose by subdividing the chain associated to Δ_1. Given that Δ_j has been defined, we let $\Delta_{j+1} \in \mathcal{M}$ be the marked cell such that $\Delta_j \in \bar{C}(\Delta_{j+1})$. We let s be the smallest integer such that $g(\Delta_s) = k-1$. Note that there must be such an integer because subdividing a chain can only increase the highest generation in the chain by one.

For each $i \geq k-1$, we keep a running count $m(i,j)$ of the number of cells Δ_ν with $g(\Delta_\nu) = j+k$, $\nu \leq i$.

CASE 1: There is a $j \in \{-1, 0, \ldots\}$ such that $m(i,j) \geq 2^{j/4}$ for some $1 \leq i \leq s$. In this case, we choose j^* as the integer which has the smallest i (defined to be $i^* := i$) with this property. In other words, j^* is the integer j whose count first exceeds $2^{j/4}$ and i^* is the smallest i for which $m(i,j^*) \geq 2^{j^*/4}$. It follows from (2.3) that

$$\text{dist}(\Delta, \Delta_\nu) \leq C_1 2^{-k/2} \sum_{j=-1}^{\infty} 2^{-j/4} = A 2^{-k/2}, \quad \nu < i^*. \tag{2.9}$$

Thus, for each Δ_ν, $\nu < i^*$, with $g(\Delta_\nu) = j^* + k$, we have $\lambda(\Delta, \Delta_\nu) = (j^*+2)^{-2}$. Since there are at least $2^{j^*/4}$ such values of ν, we obtain

$$\sum_{\Delta' \in \mathcal{M}} \lambda(\Delta, \Delta') \geq (j^*+2)^{-2} 2^{j^*/4} \geq c \tag{2.10}$$

with $c := \min_{\nu \geq -1} (\nu+2)^{-2} 2^{\nu/4}$ an absolute constant. This is (2.8) in this case.

CASE 2: In this case, for all $j \geq -1$ we have $m(s,j) \leq 2^{j/4}$. Therefore, as in (2.9), we have

$$\text{dist}(\Delta, \Delta_s) \leq C_1 2^{-k/2} \sum_{j=-1}^{\infty} 2^{-j/4} = A 2^{-k/2}. \tag{2.11}$$

This means $\lambda(\Delta, \Delta_s) = 1$ and again we have (2.8). $\ominus$

Theorem 2.4 *Suppose that $P_0, \ldots, P_n$ is a sequence of partitions generated as described above. Then, there is a constant $C_2 > 0$ depending only on P_0 such that*

$$\#(P_n) \leq \#(P_0) + C_2(\#(M_0) + \ldots + \#(M_{n-1})). \tag{2.12}$$

Proof: It follows from Lemma 2.3 and (2.7) that the number of new cells added in the subdivision and completion process does not exceed $C(\#(M_0) + \ldots + \#(M_{n-1}))/c$ and so (2.12) holds with $C_2 := C/c$. $\ominus$

It is important to note that the constant C_2 depends only on the constants $\hat{C}, G_0, a_0$. Thus we have the following consequence of Theorem 2.4.

Corollary 2.5 *If $P \in \mathcal{P}^a$ is any admissible partition (not necessarily P_0) and $P_1, \ldots, P_n$ are obtained from P by a sequence of markings and completions, then one has*

$$\#(P_n) \leq \#(P) + C_2(\#(M_0) + \ldots + \#(M_{n-1})). \tag{2.13}$$

We can also think of the completion process being applied to any proper tree $T \subset T_*$.

Theorem 2.6 *Let $T' \subset T_*$ be any proper finite subtree of T_* which is a refinement of the admissible tree T. Then there is a proper subtree T'' of T_* which is an admissible refinement of T' and*

$$N(T'') - N(T) \leq C_2(N(T') - N(T)) \tag{2.14}$$

with C_2 an absolute constant.

Proof: The tree T' is generated from T by a sequence of subdivisions. Let $\mathcal{M}_0$ be all of the leaves of T which are subdivided in creating T', i.e., that have children in T'. We apply our subdivision and completion process to $P(T)$ with the marked cells M_0. This produces the new partition P_1 and its new tree $T_1 := T(P_1)$. Now, let $\mathcal{M}_1$ denote all the cells which are leaves of T_1 and are interior nodes of T'. We apply our subdivision and completion to P_1 with marked cells $\mathcal{M}_1$. If we continue in this way we will finally arrive at an admissible partition P_n whose tree $T'' := T(P_n)$ contains T. Since

$$\#(M_0) + \ldots + \#(M_{n-1}) \leq (N(T') - N(T)) \tag{2.15}$$

the theorem follows from Theorem 2.4. ⊙

3 Galerkin approximations

Numerical methods based on Galerkin approximations to (1.1) begin with the weak formulation of (1.1) which is to find $u \in H_0^1(\Omega)$ such that

$$a(u, v) = (f, v), \quad v \in H_0^1(\Omega), \tag{3.1}$$

where $a(v, w) := (\nabla v, \nabla w)$, $(v, w) = (v, w)_\Omega := \int_\Omega vw dx$.

Given an admissible partition P, we shall denote the Galerkin solution to (3.1) by u_P throughout this paper. Thus u_P is the unique element in $\mathcal{S}_P$ which satisfies

$$a(u_p, \phi_v) = (f, \phi_v), \quad v \in \mathcal{V}_P. \tag{3.2}$$

From a practical point of view it would be better in (3.2) to replace f by an approximation and consider the solution of this modified variational problem as u_P. But we want to conform exactly to the algorithm in [14] so that we do not have to detour through a new development of a posteriori error analysis in this modified setting.

4 Adaptive approximation

In this section, we shall discuss adaptive approximation of a function v which is known to us and for which we can compute local polynomial approximants. These results do not apply directly to u since it is unknown but they serve to tell us what is the best we can expect in terms of approximating u. We shall also use the approximation methods we develop in this section in parts of our adaptive algorithm, namely to numerically approximate the right hand side f and to execute our coarsening step.

We limit ourselves to adaptive methods based on subdivision using the newest vertex subdivision rule starting with an initial partition P_0 and a labelling of vertices as given in Lemma 2.1. We recall that any adaptively generated partition P can be associated to a tree $T(P)$ which is a proper subtree of T_*. The leaves of $T(P)$ give the partition P. Conversely, any finite proper subtree T gives a partition P consisting of the leaves of T.

Recall that $\mathcal{P}_n$ denotes the set of all adaptively generated partitions which are obtained from P_0 by applying at most n elementary subdivisions. $\mathcal{P}_0$ consists of the single partition P_0. These correspond to trees $T = T(P)$ which satisfy $N(T) = n$. Similarly, we denote by $\mathcal{P}_n^a$ all the partitions in $\mathcal{P}_n$ which are admissible. We also recall our notation $\mathcal{S}_P$ of continuous piecewise linear functions subordinate to P.

4.1 Adaptive approximation in energy norm

The energy norm corresponding to the Poisson problem is defined by $\|\!|v|\!\|^2 := \|v\|^2_{H^1(\Omega)} = \|v\|^2_{L_2(\Omega)} + \|\nabla v\|^2_{L_2(\Omega)}$. Given a function $v \in H^1(\Omega)$ and a partition P, we have

$$E(v, \mathcal{S}_P)_{H^1(\Omega)} := \inf_{S \in \mathcal{S}_P} \|\!|v - S|\!\|, \tag{4.1}$$

which is the smallest error we can achieve by approximating v in the energy norm by elements of $\mathcal{S}_P$. In the case that $v = u$ is our solution to (1.1) and P is admissible then

$$E(u, \mathcal{S}_P)_{H^1(\Omega)} = \|\!|u - u_P|\!\| \tag{4.2}$$

where u_P is the Galerkin approximation associated to P.

Returning to the general case of a $v \in H^1(\Omega)$, we enter a competition over all partitions $P \in \mathcal{P}_n$ and introduce the error

$$\sigma_n(v) := \inf_{P \in \mathcal{P}_n} E(v, \mathcal{S}_P)_{H^1(\Omega)} \tag{4.3}$$

of best adaptive approximation.

It is unreasonable to expect any adaptive algorithm to perform exactly the same as $\sigma_n(v)$. However, we may expect the same *asymptotic* behavior. To quantify this, we introduce for any $s > 0$, the class $\mathcal{A}^s := \mathcal{A}^s(H^1(\Omega))$ of functions $v \in H^1(\Omega)$ such that

$$\sigma_n(v) \leq M n^{-s}, \quad n = 1, 2, \ldots. \tag{4.4}$$

The smallest M for which (4.4) is satisfied is the norm in $\mathcal{A}^s$:

$$\|v\|_{\mathcal{A}^s} := \sup_{n \geq 0} n^s \sigma_n(v). \tag{4.5}$$

We have a similar measure of approximation when we restrict ourselves to admissible partitions. Namely,

$$\sigma_n(v)^a := \inf_{P \in \mathcal{P}_n^a} E(v, \mathcal{S}_P)_{H^1(\Omega)} \tag{4.6}$$

now measures the best nonlinear approximation error obtained from admissible partitions and $\dot{\mathcal{A}}^s := \dot{\mathcal{A}}^s(H^1(\Omega))$ consists of all v which satisfy

$$\sigma_n^a(v) \leq M n^{-s}, \quad n = 1, 2, \ldots. \tag{4.7}$$

The smallest M for which (4.7) holds serves to define the norm $\|v\|_{\dot{\mathcal{A}}^s}$.

Note that in view of Theorems 2.4,2.6 we have $\mathcal{A}^s(H^1(\Omega)) = \dot{\mathcal{A}}^s(H^1(\Omega))$ with equivalent norms.

The reader should be interested to know what functions are in $\mathcal{A}^s$. It turns out that these classes are related to certain Besov spaces. Since we do not use or need this information in the construction of the algorithm, we postpone this discussion to Appendix 8.

4.2 Adaptive approximation in $H^{-1}(\Omega)$

We shall also need approximation by piecewise constants which will be our vehicle for resolving the right hand side f in our numerical algorithm for (1.1). The approximation will take place in the $H^{-1}(\Omega)$ norm which is defined for a tempered distribution g by duality:

$$\|g\|_{H^{-1}(\Omega)} := \sup_{\phi \in H_0^1(\Omega)} \frac{\langle g, \phi \rangle}{\|\phi\|}, \tag{4.8}$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing induced by the standard L_2-inner product. Given a partition P, we let $\mathcal{S}_P^0$ denote the class of piecewise constant functions subordinate to P. For a function $g \in H^{-1}(\Omega)$, we have

$$E(g, \mathcal{S}_P^0)_{H^{-1}(\Omega)} := \inf_{S \in \mathcal{S}_P^0} \|g - S\|_{H^{-1}(\Omega)}, \tag{4.9}$$

which is the best error we can achieve by approximating f in the $H^{-1}(\Omega)$-norm by elements of $\mathcal{S}_P^0$. Analogously, we have

$$\sigma_n(g)_{H^{-1}(\Omega)} := \inf_{P \in \mathcal{P}_n} E(g, \mathcal{S}_P^0)_{H^{-1}(\Omega)} \tag{4.10}$$

the error of best nonlinear approximation to g by piecewise constants.

As in the case of piecewise linear approximation, we introduce for any $s > 0$, the approximation class $\mathcal{A}^s(H^{-1}(\Omega))$ and its norm exactly as in (4.4) and (4.5) except that we use $\sigma_n(g)_{H^{-1}(\Omega)}$ in place of $\sigma_n(v)$. We have a similar measure of approximation when we restrict ourselves to admissible partitions. We denote the approximation class in this case by $\dot{\mathcal{A}}^s(H^{-1}(\Omega))$

Suppose now that $g \in L_2(\Omega)$. Then $g \in H^{-1}(\Omega)$. If P is any partition of Ω and $\Delta \in P$, we define

$$g_\Delta := \frac{1}{|\Delta|} \int_\Delta g \tag{4.11}$$

which is the average of g over Δ. Also, g_Δ is the best approximation to g in $L_2(\Delta)$ by constant functions. Therefore,

$$S_P^0(g) := \sum_{\Delta \in P} g_\Delta \chi_\Delta \tag{4.12}$$

is the best $L_2(\Omega)$ approximation to g by piecewise constants subordinate to P.

We can use $S_P^0(g)$ also to approximate g in the $H^{-1}(\Omega)$-norm. In fact, for any admissible partition P, we have the following bound for the approximation error

$$E(g, \mathcal{S}_P^0)_{H^{-1}(\Omega)}^2 \leq \|g - S_P^0(g)\|_{H^{-1}(\Omega)}^2 \leq C_0 \bar{E}(g, P) \tag{4.13}$$

where

$$\bar{E}(g, P) := \sum_{\Delta \in P} |\Delta| \|g - g_\Delta\|_{L_2(\Delta)}^2. \tag{4.14}$$

Indeed, to see that this is true, let ϕ be any function in $H_0^1(\Omega)$ of unit norm. Then for any $\phi_\Delta \in \mathcal{S}_P^0$

$$\langle g - S_P^0(g), \phi \rangle = \sum_{\Delta \in P} \int_\Delta (g - g_\Delta)(\phi - \phi_\Delta), \tag{4.15}$$

because $g - g_\Delta$ has integral zero over Δ. We use the Cauchy-Schwartz inequality on each of the terms in the last sum and then the Poincaré inequality to bound such a term by

$$\|g - g_\Delta\|_{L_2(\Delta)} \|\phi - \phi_\Delta\|_{L_2(\Delta)} \leq C |\Delta|^{1/2} \|\nabla \phi\|_{L_2(\Delta)} \|g - g_\Delta\|_{L_2(\Delta)}.$$

Here C is an absolute constant because all the triangular cells Δ are uniformly shape regular (2.1). Using this in (4.15) and again applying Cauchy-Schwarz, we arrive at (4.14) by taking a supremum over all ϕ of norm one in $H_0^1(\Omega)$.

We can use $\bar{E}(g, P)$ to define another nonlinear approximation error:

$$\bar{\sigma}_n^2(g)_{H^{-1}(\Omega)} := \inf_{P \in \mathcal{P}_n} \bar{E}(g, P). \tag{4.16}$$

Analogous to the approximation classes defined above, we define $\bar{\mathcal{A}}^s(H^{-1}(\Omega))$ using $\bar{\sigma}_n(g)_{H^{-1}(\Omega)}$ and the norm for this class as before. When dealing with admissible partitions we shall denote this class by $\dot{\bar{\mathcal{A}}}^s(H^{-1}(\Omega))$. As before $\dot{\bar{\mathcal{A}}}^s(H^{-1}(\Omega)) = \bar{\mathcal{A}}^s(H^{-1}(\Omega))$ with equivalent norms.

4.3 An algorithm for adaptive approximation of a given function

In this section, we wish to describe some of the results of [2] which give adaptive approximation algorithms for approximating a given target function v defined on Ω in a specified norm. These algorithms are different from an AFEM since they assume that the target function v is fully known (whereas our solution u to (1.1) is not). We shall use these algorithms in two different settings which we shall describe in the following subsections.

Although the algorithms in [2] apply in a more general setting, we shall limit our discussion to the case of newest vertex bisection and its associated master tree T_*. The starting assumption is that there is a functional e which associates to each triangular cell Δ in the master tree a nonnegative real number $e(\Delta)$. In applications, $e(\Delta)$ is usually some local approximation error (or a bound for this local error) associated to Δ.

There are two algorithms, called **Second Algorithm** and **Modified Second Algorithm** in [2], that we shall utilize.

The **Second Algorithm** in [2] can be applied under the assumption that e satisfies the following property: for each $\Delta \in T_*$ and each subtree T which contains Δ as its only root, we have

$$\sum_{\Delta' \in \mathcal{L}(T)} e(\Delta') \leq C\, e(\Delta) \tag{4.17}$$

where C is an absolute constant and $\mathcal{L}(T)$ is the set of leaves of T. This algorithm can be employed to resolve the right hand side f (see §4.4). Note that (4.17) follows, in particular, when e is *subadditive*, i.e.,

$$e(\Delta_1) + e(\Delta_2) \leq e(\Delta). \tag{4.18}$$

holds for any $\Delta \in T_*$ and its children Δ_1 and Δ_2. This will be seen to be the case when approximating f.

The **Modified Second Algorithm** in [2] is built to overcome certain technical difficulties that can arise in AFEM applications caused by the overlapping supports of nodal basis functions. It also begins with the property (4.17) but it is enforced only for the *admissible* subtrees T of T_*.

We next discuss the **Second Algorithm** in a little more detail. For any proper subtree $T \subset T_*$, we define

$$E(T) := \sum_{\Delta \in \mathcal{L}(T)} e(\Delta) \tag{4.19}$$

as the error associated to T. In applications to adaptive partitioning, $E(T)$ would correspond to the square of the error associated to the partition given by the leaves of T. We enter the following competition among all proper trees in $\mathcal{T}_n$:

$$E_n := \inf_{T \in \mathcal{T}_n} E(T). \tag{4.20}$$

So E_n is the smallest error we could achieve using trees from $\mathcal{T}_n$. Although the problem is finite, it is numerically too intensive to find a best tree which achieves the minimum error E_n because $\#(\mathcal{T}_n)$ is exponential in n.

Suppose that given any Δ, we are able to compute $e(\Delta)$. We would like to find a tree which performs almost as well as the best tree from $\mathcal{T}_n$ and to do so while only computing $O(n)$ values $e(\Delta)$. The main result of [2] is to show that this is possible. Namely, that paper gives a numerically realizable algorithms which yield *near best* trees T from $\mathcal{T}_n$ by which we mean

$$E(T) \leq C_0 E_{c_0 n} \tag{4.21}$$

where $C_0, c_0 > 0$ are absolute constants.

The algorithm creates a sequence of trees $T = T_j$, $j = 1, 2, \ldots$ starting from some initial tree T_0. Roughly speaking, at any given stage in the algorithm, it computes $e(\Delta)$ for all leaves in the current tree and then subdivides the ones with the largest value. However, to avoid a lengthy sequence of subdivisions with insufficient reduction in the error, the algorithm in actuality uses a modified functional $\tilde{e}$. What is important in the context of the present paper is that to create the tree in (4.21) requires the computation of at most Cn values of e so that the algorithm is computationally of order n.

We shall use this algorithm in the following setting described in [2] as **Thresholding Second Algorithm**. Given a tolerance $\mu > 0$, and an admissible partition P and its corresponding tree $T(P)$, the algorithm produces a tree T_μ which is a refinement of $T_0 = T(P)$ with the following properties:

P1: T_μ satisfies

$$E(T_\mu) \leq \mu. \tag{4.22}$$

P2: For absolute constants c_1 and C_1, it holds that, whenever $\tilde{T}$ is a refinement of $T(P)$ that satisfies

$$E(\tilde{T}) \leq c_1 \mu, \tag{4.23}$$

then it follows that

$$N(T_\mu) \leq C_1 N(\tilde{T}). \tag{4.24}$$

Applying this algorithm directly as described in [2] does not give an admissible tree. However, by virtue of Theorem 2.6, the output tree of the algorithm can be completed to an admissible tree by using at most $C_2(N(T_\mu) - N(T(P)))$ additional refinements. The subadditivity property (4.18) guarantees that the error for the admissible tree can be compared with that of T_μ. Putting all of this together we have an algorithm with the following properties.

Approximation Algorithm (AA): *Given an initial partition P, an error functional e satisfying (4.18), and an error tolerance $\mu > 0$,* **AA** *produces as output an admissible partition $P' = \mathbf{AA}(P, \mu)$ which is a refinement of P and satisfies*

$$E(T(P')) \leq \mu. \tag{4.25}$$

Moreover, there are absolute constants $c_1, C_1 > 0$ such that whenever $\tilde{P}$ is any refinement of P which satisfies $E(T(\tilde{P})) \leq c_1 \mu$, then we have

$$\#P' - \#P \leq C_1(\#\tilde{P} - \#P). \tag{4.26}$$

The number of evaluations $N(\mathbf{AA}, P, \mu)$ of e to compute the output P' to **AA** *satisfies*

$$N(\mathbf{AA}, P, \mu) \leq C_3(\#(P')). \tag{4.27}$$

Let us now discuss the **Modified Second Algorithm** of [2] which will be used in our coarsening step. This algorithm uses a new subdivision rule that consists of five consecutive subdivisions for each marked cell. After all marked cells of a current tree T_j are subdivided, a completion process is applied to the result to obtain the new *admissible* tree T_{j+1}.

It enters the competition

$$E_n := \inf_{T \in \mathcal{T}_n^a} E(T) \tag{4.28}$$

among all admissible trees. The analogue of the algorithm **AA** is an algorithm with the following properties given in §7 of [2].

Second Approximation Algorithm (SAA): *Given an initial admissible partition $P \in \mathcal{P}^a$, an error functional e satisfying (4.17), and an error tolerance $\mu > 0$,* **SAA** *produces as output an admissible partition $P' = \mathbf{SAA}(P, \mu)$ which is a refinement of P and satisfies*

$$E(T(P')) \leq \mu. \tag{4.29}$$

Moreover, there are absolute constants $c_1, C_1 > 0$ such that whenever $\tilde{P}$ is any admissible refinement of P which satisfies $E(T(\tilde{P})) \leq c_1 \mu$, then we have

$$\#P' - \#P \leq C_1(\#\tilde{P} - \#P). \tag{4.30}$$

The number of evaluations $N(\mathbf{SAA}, P, \mu)$ of e to compute the output P' to **SAA** *satisfies*

$$N(\mathbf{SAA}, P, \mu) \leq C_3(\#(P')). \tag{4.31}$$

4.4 Approximation of the right hand side

We shall describe in this section an algorithm for approximating f which will be a part of our adaptive algorithm for solving (1.1). We shall approximate f by piecewise constants on admissible partitions that are adaptively generated using newest vertex bisection. To achieve this approximation we shall use the first adaptive approximation algorithm (namely **AA**) of §4.3. Let P_0 be any fixed initial partition and let T_* denote the master tree for newest vertex bisection. As before, we let $\mathcal{P}^a$ denote the class of all admissible partitions that can be obtained using newest vertex bisection.

For each $\Delta \in T_*$, we define

$$e(f, \Delta) = e(\Delta) := |\Delta| \|f - f_\Delta\|^2_{L_2(\Delta))}. \tag{4.32}$$

The property (4.18) follows easily from the fact that f_Δ is the best $L_2(\Delta)$ approximation to f.

The following statements describe our algorithm for approximating f and its properties:

APPROX $(f, P, \mu) \to P'$: *The input of this algorithm is the function $f \in L_2(\Omega)$, an admissible partition $P \in \mathcal{P}^a$, and the error tolerance $\mu > 0$. The algorithm uses the inputs f, P and μ^2 in algorithm* **AA** *with the functional e defined by* (4.32) *and receives the output partition $P' =$* **APPROX**$(f, P, \mu) =$ **AA** (P, μ^2) *which, by the properties of* **AA**, *is admissible and satisfies:*

$$\bar{E}(f, P') = E(T(P')) \leq \mu^2. \tag{4.33}$$

From the properties of **(AA)** we immediately have the following facts.

Proposition 4.1 *If $\tilde{P}$ is any partition which is a refinement of P and satisfies $\bar{E}(f, \tilde{P}) \leq c_1\mu^2$, then*

$$\#(P') - \#(P) \leq C_1(\#(\tilde{P}) - \#(P)), \tag{4.34}$$

where c_1, C_1 are the constants associated to **AA**. *The number of evaluations $N(\mathbf{APROX}, P, \mu)$ of e to compute the output P' to* **APPROX** *satisfies*

$$N(\mathbf{APPROX}, P, \mu) \leq C(\#(P')). \tag{4.35}$$

The following lemma describes how we shall use **APPROX**.

Lemma 4.2 *If we apply* **APPROX** *to an input function* $f \in \bar{\mathcal{A}}^s = \bar{\mathcal{A}}^s(H^{-1}(\Omega))$, $s > 0$, *an input partition* P, *and an input tolerance* $\mu > 0$, *then the output partition* $P' =$ **APPROX**(f, P, μ) *satisfies*

$$\#(P') \leq \#(P) + C_2(s)\|f\|_{\bar{\mathcal{A}}^s}^{1/s}\mu^{-1/s} \tag{4.36}$$

with $C_2(s)$ *a constant depending only on* s.

Proof: Let P^* be the smallest partition in $\mathcal{P}$ which satisfies

$$\bar{E}(f, P^*) \leq c_1\mu^2. \tag{4.37}$$

where c_1 is the constant appearing in algorithm **AA**. Then, by the definition of the class $\bar{\mathcal{A}}^s$, we have

$$\#(P^*) \leq \|f\|_{\bar{\mathcal{A}}^s}^{1/s}(\sqrt{c_1}\mu)^{-1/s}. \tag{4.38}$$

We generate an admissible partition $\bar{P}$ which is a common refinement of P and P^* as follows. Starting with P, we define $\mathcal{M}_0$ to be the set of all triangular cells which are leaves of P and whose children are in $T(P^*)$. We refine P using the marked cells $\mathcal{M}_0$ and then complete the resulting partition to obtain P_1 which is admissible. We let $\mathcal{M}_1$ be the set of all triangular cells which are leaves of P_1 and whose children are in $T(P^*)$. We refine P_1 using the marked cells $\mathcal{M}_1$ and then complete the resulting partition to obtain P_2. We continue in this way getting $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_k$ with $\mathcal{M}_{k+1} = \emptyset$ and $\bar{P} := P_{k+1}$. We have shown in Corollary 2.5 that we can bound the number of cells in $\bar{P}$ by

$$\#(\bar{P}) - \#(P) \leq C_2 \sum_{j=0}^{k} \#\mathcal{M}_j \leq C_2\#(P^*) \leq C_2\|f\|_{\bar{\mathcal{A}}^s}^{1/s}(\sqrt{c_1}\mu)^{-1/s}. \tag{4.39}$$

By (4.18)

$$\bar{E}(f, \bar{P}) \leq \bar{E}(f, P^*) \leq c_1\mu^2. \tag{4.40}$$

It follows from (4.26) that

$$\#(P') - \#(P) \leq C_1(\#(\bar{P}) - \#(P)) \leq C_1C_2\|f\|_{\bar{\mathcal{A}}^s}^{1/s}(\sqrt{c_1}\mu)^{-1/s}, \tag{4.41}$$

which is the desired inequality. ⊙

4.5 Coarsening

In this section, we describe our second application of the adaptive approximation algorithm. The setting is the following. Suppose that P is any admissible partition and u_P is the Galerkin solutions to (1.1) on this partition. We assume that we have a computable function $\Phi(P)$ which provides an upper bound for the Galerkin error

$$\|u - u_P\|^2 \leq \Phi(P). \tag{4.42}$$

Such an upper bound Φ is given in §5 (see (5.2)).

The algorithm we shall construct in this subsection starts with the input of two admissible partitions P, P' with P' a refinement of P, and a tolerance μ for which we know the bounds

$$\|u - u_P\|^2 \leq \Phi(P) \leq \mu^2 \tag{4.43}$$

and

$$\|u - u_{P'}\|^2 \leq \Phi(P') \leq \alpha\mu^2. \tag{4.44}$$

At this stage, the constant $0 < \alpha < 1$ is arbitrary but fixed. We shall specify the value of α, and the two constants β and γ that follow, at the end of §7. For now the reader should think of these as arbitrary but fixed constants satisfying the relations we specify below.

With these inputs, we shall generate an admissible partition $\bar{P}$ which is a refinement of P and a coarsening of P' such that a certain further refinement $r(\bar{P})$ satisfies

$$\|u - u_{r(\bar{P})}\|^2 \leq \beta\mu^2, \tag{4.45}$$

where $\alpha < \beta < 1$. The main property (not held by P') we will gain in the construction of $\bar{P}$ is that we will be able to bound its number of elements.

To find $\bar{P}$, we are going to use algorithm **SAA** with input $u_{P'}$. We need to define the error functional e for that algorithm. To do this, we introduce the *minimal ring* associated to a triangular cell $\Delta \in T_*$. Given any admissible partition P from $\mathcal{P}^a$ and $\Delta \in P$, we define

$$R(\Delta, P) := \bigcup_{\Delta' \in P, \Delta \cap \Delta' \neq \emptyset} \Delta' \tag{4.46}$$

which is the first ring about Δ. This ring depends on P. However, we can find a minimal ring about Δ which does not depend on P. Namely, we define

$$R_-(\Delta) := \bigcap_{P \in \mathcal{P}^a, \Delta \in P} R(\Delta, P) = \bigcup_{\Delta' \in P_-(\Delta)} \Delta' \tag{4.47}$$

where $P_-(\Delta)$ is the collection of cells from T_* which touch Δ and make up $R_-(\Delta)$.

Before returning to the definition of e that we will use, we take this opportunity to develop a little further the properties of the set $P_-(\Delta)$ because we will need these properties later in this section. Let m_0 be the maximal valence that can occur for any vertex v in any admissible partition. Because of the uniform boundedness from below of the angles in triangular cells for admissible partitions, it follows that m_0 is a fixed finite constant.

Lemma 4.3 *Suppose P is an admissible partition and $\Delta \in P$ has the label $(k+1, k+1, k)$. Any cell $\Delta' \in P$ such that $\Delta' \cap \Delta \neq \emptyset$ has label $(k+\nu, k+\nu, k+\nu-1)$ where $-\lfloor \frac{m_0-3}{2} \rfloor \leq \nu \leq \lceil \frac{m_0}{2} \rceil$.*

Proof: Any such Δ' shares a vertex with Δ. We consider the vertex v which is common to the two edges labelled $k+1$, $k+1$. The argument for the other vertices is similar. We claim that the maximum label of any of the edges containing v is $k+\ell$ where $\ell \leq \lceil \frac{m_0}{2} \rceil$. To see this note first that the maximum label of edges in two adjacent cells can differ by at most 1. Now suppose that the number of edges incident at v is r. To go from the label $k+1$ (the label of one of the edges of Δ which contains v) to $k+\ell$ and then from $k+\ell$

back down to $k+1$ (the label of the other edge of Δ containing v) requires $r = 2\ell$ edges when r is even and $2\ell - 1$ edges when r is odd. Thus for the largest label $k+\ell$ we have $\ell = \lceil \frac{r}{2} \rceil$. This confirms the upper bound for the case that v belongs to two edges with label $k+1$. An analogous argument applies when one of the edges has label k. To obtain a lower bound, suppose that a vertex v is common to an edge with label k and one with label $k+1$ and that again r edges meet at v. Since it takes at least $\ell + 2$ steps to go down to the label $k - \ell$ and another $\ell + 1$, respectively ℓ steps to go up to label k again when r is odd, respectively even, the smallest possible label is $k - \ell$ for $\ell = \frac{r-3}{2}$ when r is odd and $\ell = \frac{r-2}{2}$ when r is even. When r is odd this lowest possible label has to appear twice on consecutive incident edges. When r is even, this label can only be attained once by an incident edge and is therefore the generation of the two adjacent triangles. Hence the are labelled $(k - \frac{r-2}{2} + 1, k - \frac{r-2}{2} + 1, k - \frac{r-2}{2})$. Thus the lowest possible label of a triangle sharing the above v is $(k - \lfloor \frac{r-3}{2} \rfloor, k - \lfloor \frac{r-3}{2} \rfloor, k - \lfloor \frac{r-3}{2} \rfloor - 1)$, which confirms the lower bound. $\ominus$

For the formulation of the next lemma, we introduce the notion of a *full refinement*. Given an admissible partition P, by a full refinement of P we mean the subdivision of each cell in P into six cells using newest vertex bisection. These new cells consist of two grandchildren of Δ and four great grandchildren of Δ and result in the bisection of each original edge of Δ as well as the bisection of the new edge added when Δ is subdivided using newest vertex bisection.

Lemma 4.4 *Assume that P is any admissible partition and P' is the partition obtained from P by subjecting each cell in P to $\lceil \frac{m_0}{2} \rceil$ full refinements. Then P' is admissible and any ring $R(\Delta', P')$ of a cell $\Delta' \in P'$ is contained in the minimal ring $R_-(\Delta)$ of the ancestor $\Delta \in P$ of Δ'.*

Proof: The admissibility of P' follows from the fact that one full refinement leaves no hanging nodes. For the second property note that a full refinement to a triangular cell with label $(k+1, k+1, k)$ produces cells whose new edges carry labels that are at least $k+2$. Thus ℓ successive full refinements produce triangles with generations at least $k + 2\ell$. Next note that the minimal ring of Δ consists of triangles touching Δ and having the highest possible generation admitted by T_*. We have shown in Lemma 4.3 that the minimal generation of cells in $R(P, \Delta)$ is $k - \ell - 1$ where $\ell = \lfloor \frac{m_0 - 3}{2} \rfloor$. Thus, applying r full refinements the smallest possible generation appearing in $R(P, \Delta)$ will increase to $g(r) := k - \lfloor \frac{m_0-3}{2} \rfloor - 1 + 2r$. By Lemma 4.3, the highest possible generation appearing in $R(P, \Delta)$ is $G := k + \lceil \frac{m_0}{2} \rceil - 1$. It can be checked that $g(r) \geq G$, provided that $r \geq m_0/2$. This completes the proof. $\ominus$

We now return to our problem of coarsening a partition P. Given a function $w \in H^1(\Omega)$ (later we shall take $w = u_{P'}$), we define

$$e(w, \Delta) = e(\Delta) := \inf_S \|w - S\|^2_{H^1(R_-(\Delta))} = \|w - S_\Delta\|^2_{H^1(R_-(\Delta))}, \tag{4.48}$$

where the infimum is taken over all continuous piecewise linear functions S defined on $R_-(\Delta)$ which are subordinate to $P_-(\Delta)$.

Proposition 4.5 *The local error functional e, defined by (4.48), satisfies the weak subadditivity property (4.17). Moreover, for $E = E(w, \cdot)$ be defined by (4.19) for this choice of e, and any admissible partition $P \in \mathcal{P}^a$ one has*

$$E(w, T(P)) \leq C \inf_{S \in \mathcal{S}_P} |||w - S|||^2, \tag{4.49}$$

where the constant C depends only on the initial partition P_0 which consists of the roots of T_*.

Proof: Let T be a finite admissible tree with single root Δ. The leaves $\mathcal{L}(T)$ of T form an admissible partition of Δ. For each $\Delta' \in \mathcal{L}(T)$, we have that $P_-(\Delta')$ is a refinement of $P_-(\Delta)$ on the set $R_-(\Delta')$, and

$$\|w - S_{\Delta'}\|_{H^1(R_-(\Delta'))} \leq \|w - S_\Delta\|_{H^1(R_-(\Delta'))}, \tag{4.50}$$

by the very definition of $S_{\Delta'}$.

Let us observe that a given point $x \in R_-(\Delta)$ appears in at most C of the sets $R_-(\Delta')$, $\Delta' \in \mathcal{L}(T)$, with C an absolute constant. Indeed, if $x \in \Delta^*$ with $\Delta^* \in \mathcal{L}(T)$, then, because of the minimality of the rings, x will not appear in any $R_-(\Delta')$ unless Δ' touches $R_-(\Delta^*)$. Since the partition $\mathcal{L}(T)$ is admissible, there are at most C cells which touch $R_-(\Delta^*)$ with C an absolute constant. We use this property and the set subadditivity of $\|\cdot\|^2_{H^1}$ to find,

$$\sum_{\Delta' \in \mathcal{L}(T)} \|w - S_{\Delta'}\|^2_{H^1(R_-(\Delta'))} \leq \sum_{\Delta' \in \mathcal{L}(T)} \|w - S_\Delta\|^2_{H^1(R_-(\Delta'))} \leq C\|w - S_\Delta\|^2_{H^1(R_-(\Delta))} \tag{4.51}$$

where the last inequality uses the set subadditivity and the fact that each $R_-(\Delta') \subset R_-(\Delta)$. This inequality verifies property (4.17) for this choice of e.

As for (4.49), let w_P be the minimizer of the right hand side of (4.49) and note that only a finite uniformly bounded number of the minimal rings $R_-(\Delta)$, $\Delta \in P$ overlap at any given point in Ω. Hence

$$\sum_{P \in P} \|w - S_\Delta\|^2_{H^1(R_-(\Delta))} \leq \sum_{P \in P} \|w - w_P\|^2_{H^1(R_-(\Delta))} \leq C|||w - w_P|||^2,$$

as claimed. $\ominus$

We cannot expect that the inequality (4.49) can be reversed in general. However, we shall see next that for certain partitions the error in the energy norm can be bounded by E. To this end, let us denote for any admissible partition $P \in \mathcal{P}^a$ by $r(P)$ the partition which is obtained by applying $\lceil \frac{m_0}{2} \rceil$ full refinements to each cell in P. It is obvious that $r(P)$ is again admissible.

Proposition 4.6 *There exists a constant C_4 depending only on the initial partition P_0 such that for any admissible refinement $P \in \mathcal{P}^a$ of P_0, one has*

$$\inf_{S \in \mathcal{S}_{r(P)}} |||w - S|||^2 \leq C_4 E(w, T(P)). \tag{4.52}$$

Proof: We shall employ quasi-interpolants for bounding the best approximation in the energy norm. To this end, recall that the Courant elements ϕ_v, $v \in \mathcal{V}_{r(P)}$, form a basis for $\mathcal{S}_{r(P)}$. Thus each $S \in \mathcal{S}_{r(P)}$ has the unique representation

$$S = \sum_{v \in \mathcal{V}_{r(P)}} S(v)\phi_v = \sum_{v \in \mathcal{V}_{r(P)}} \lambda_v(S)\tilde{\phi}_v, \tag{4.53}$$

where the $\tilde{\phi}_v$ are normalized to have norm one in $H^1(\Omega)$, i.e. $\|\tilde{\phi}_v\|_{H^1(\Omega)} = 1$, and where the λ_v are dual functionals, i.e. $\lambda_v(\tilde{\phi}_{v'}) = \delta_{v,v'}$. This means that the λ_v also have norm one as linear functionals on $\mathcal{S}_{r(P)}$ when this space is equipped with the $H^1(\Omega)$ norm. There is a norm preserving extension λ_v (which we continue to denote by λ_v) to all of $H^1(\Omega)$ and we can require that this extension is given as the integral with an L_2 function Λ_v:

$$\lambda_v(g) = \int_{B_v} g\Lambda_v, \tag{4.54}$$

with

$$B_v := \mathrm{supp}(\Lambda_v) \subset \bigcup\{\Delta : v \in \Delta\}, \tag{4.55}$$

the *star* of v. It follows that

$$|\lambda_v(g)| \leq \|g\|_{H^1(B_v)}. \tag{4.56}$$

Clearly, $Q_{r(P)}(w) := \sum_{v \in \mathcal{V}_{r(P)}} \lambda_v(w)\tilde{\phi}_v$ defines a projector from $L_2(\Omega)$ into $\mathcal{S}_{r(P)}$. Let us now bound $\|w - Q_{r(P)}(w)\|_{H^1(\Omega)}$. Given any triangular cell $\Delta \in P$, we have

$$\begin{aligned} \|w - Q_{r(P)}(w)\|^2_{H^1(\Delta)} &\leq 2\|w - S_\Delta\|^2_{H^1(\Delta)} + 2\|Q_{r(P)}(w - S_\Delta)\|^2_{H^1(\Delta)} \\ &\leq 2e(\Delta) + 2\|Q_{r(P)}(w - S_\Delta)\|^2_{H^1(\Delta)}, \end{aligned} \tag{4.57}$$

because $Q_{r(P)}(S_\Delta) = S_\Delta$ on Δ. We need to bound the second term in (4.57). Let us note that on Δ only finitely many terms $\lambda_v(w - S_\Delta)\tilde{\phi}_v$ are nonzero. These correspond to the vertices from $r(P)$ that are in Δ and, in view of the fact that each full refinement introduces four new vertices on each triangle, there are at most $C = 3 + 4^{\lceil \frac{m_0}{2} \rceil}$ vertices from $r(P)$ in Δ. Let v be one of these vertices, then

$$\|\lambda_v(w - S_\Delta)\tilde{\phi}_v\|^2_{H^1(\Delta)} \leq \|w - S_\Delta\|^2_{H^1(B_v)} \leq \|w - S_\Delta\|^2_{H^1(R_-(\Delta))}\,, \tag{4.58}$$

where we have used that, by Lemma 4.4, $B_v \subset R_-(\Delta)$, $v \in \Delta \cap \mathcal{V}_{r(P)}$. Hence, summing over all vertices $v \in \mathcal{V}_{r(P)}$ in Δ, we obtain

$$\|Q_{r(P)}(w - S_\Delta)\|^2_{H^1(\Delta)} \leq Ce(\Delta). \tag{4.59}$$

Using this back in (4.57) and summing over all $\Delta \in P$, gives

$$\|w - Q_{r(P)}(w)\|^2_{H^1(\Omega)} \leq C_4\ E(w, T(P)). \tag{4.60}$$

This proves the assertion. $\ominus$

Throughout the remainder of this section let E be defined by (4.19) for e, defined by (4.48). We return now to our problem of generating the set $\bar{P}$. We have the following

algorithm.

COARSE $(P, P', \mu) \to \bar{P}$
Given as inputs the two admissible partitions P, P', the Galerkin approximation $u_{P'}$ and the tolerance μ such that (4.43) and (4.44) hold. The algorithm **COARSE** *outputs the admissible partition $\bar{P}$, which is a refinement of P, as follows:*

> *Apply* **SAA** $(P, \gamma\mu^2) \to \bar{P}$ *with respect to the error functional e defined by (4.48) and $w = u_{P'}$ where γ is specified below.*

The rest of this subsection will be devoted to deriving the important properties of the output $\bar{P}$ of **COARSE**. Let us first observe that given P', the number of computations $N(P', \mu)$ necessary to invoke the algorithm, is bounded by

$$N(P', \mu) \leq C_3 \#(P') \tag{4.61}$$

with C_3 an absolute constant. Indeed, we find S_Δ, $\Delta \in P'$, from the least squares problem by using a fixed number of computations because $u_{P'}$ is continuous and piecewise linear and the number of cells in $P_-(\Delta)$ has an absolute bound (see our discussion of the structure of $P_-(\Delta)$ given above).

Next, we want to bound $\|u - u_{r(\bar{P})}\|$ for the Galerkin solution $u_{r(\bar{P})}$. To do this, note that, by Proposition 4.6,

$$\|u - u_{r(\bar{P})}\|^2 \leq 2\|u - u_{P'}\|^2 + 2C_4\gamma\mu^2 \leq 2(\alpha + C_4\gamma)\mu^2 \leq \beta\mu^2 \tag{4.62}$$

provided α and γ are chosen so as to satisfy

$$\alpha \leq \beta/4 \quad \text{and} \quad C_4\gamma \leq \beta/4. \tag{4.63}$$

We impose these requirements on α, β and γ in what follows.

We now summarize the properties of the output $\bar{P}$ of algorithm **COARSE**.

Theorem 4.7 *Given the inputs $P \subset P'$ of two admissible partitions and an input tolerance μ, the output $\bar{P} =$***COARSE**(P, P', μ) *satisfies:*
(i) The Galerkin solution $u_{r(\bar{P})}$ on $r(\bar{P})$ satisfies:

$$\|u - u_{r(\bar{P})}\|^2 \leq \beta\mu^2. \tag{4.64}$$

*(ii) The number of flops $N($***COARSE**$, P, \mu)$ *used to compute $\bar{P}$ satisfies*

$$N(\textbf{COARSE}, P, P', \mu) \leq C_3(\#(P')) \tag{4.65}$$

with C_3 an absolute constant.

(iii) If $u \in \dot{\mathcal{A}}^s = \dot{\mathcal{A}}^s(H^1(\Omega))$, then

$$\#(\bar{P}) \leq \#(P) + C_2(s)\|u\|^{1/s}_{\dot{\mathcal{A}}^s(H^1(\Omega))}\mu^{-1/s} \tag{4.66}$$

with $C_2(s)$ a constant depending only on s.

Proof: Statement (i) is just (4.62). Statement (ii) follows because of (4.31) and the bound (4.61) we have already mentioned for computing $u_{P'}$. The proof of (iii) is similar to the proof in Lemma 4.2. Let P^* be the smallest admissible partition that satisfies

$$\|\!|u - u_{P^*}|\!\|^2 \leq c_3\mu^2 \tag{4.67}$$

where c_3 is a constant which is specified in a few lines. From the fact that $u \in \dot{\mathcal{A}}^s$, we know

$$\#(P^*) \leq \|u\|^{1/s}_{\dot{\mathcal{A}}^s}(\sqrt{c_3}\mu)^{-1/s}. \tag{4.68}$$

Now let $\widetilde{P}$ be a common refinement of P and P^* which is admissible and satisfies

$$\#(\widetilde{P}) \leq \#(P) + C(\#(P^*) \leq \#(P) + C\|u\|^{1/s}_{\dot{\mathcal{A}}^s}(\sqrt{c_3}\mu)^{-1/s}. \tag{4.69}$$

This refinement is constructed exactly as in the proof of Lemma 4.2. It follows that

$$\|\!|u - u_{\widetilde{P}}|\!\|^2 \leq c_3\mu^2 \tag{4.70}$$

Hence,

$$\|\!|u_{P'} - u_{\widetilde{P}}|\!\|^2 \leq 2\|\!|u - u_{\widetilde{P}}|\!\|^2 + 2\|\!|u - u_{P'}|\!\|^2 \leq 2(c_3 + \alpha)\mu^2, \tag{4.71}$$

where we have used (4.44) and (4.70) in the last step.

Recalling that $w = u_{P'}$ in **COARSE**, it follows from Proposition 4.5 that

$$E(u_{P'}, T(P'')) \leq C_2\|u_{P'} - u_{P''}\|^2_{H^1(\Omega)} \leq 2C_2(c_3 + \alpha)\mu \leq c_1\mu, \tag{4.72}$$

where c_1 is the constant in Algorithm **SAA**, provided that the constants α and c_3 are chosen sufficiently small which puts another condition on their value, see (4.63). It follows from (4.30) that

$$\#\bar{P} - \#P \leq C_1\left(\#(\widetilde{P}) - \#(P)\right). \tag{4.73}$$

When this is used together with (4.69) we arrive at (4.66). ⊙

5 The adaptive algorithm of Morin, Nochetto, and Siebert

In this section, we shall recall the AFEM of Morin, Nochetto, and Siebert [14] which is based on subdividing (using the newest vertex rule) certain collections of marked cells. We shall later modify this algorithm to arrive at our new algorithm. Our notation is slightly different from theirs in places.

The strategy in [14] for marking cells for subdivision and the guaranteed error reduction that results is based on two criteria. The first of these is a local error indicator that we now describe. Suppose that $P \in \mathcal{P}^a$ is an admissible partition. Given an edge $E \in \mathcal{E}_P$, we let Ω_E denote the union of the two triangles that share E and let ℓ_E denote the length of E. We also denote by $J_E(P)$ the jump of ∇u_P across E. The quantity

$$\eta_E := \eta_E(P) := \|\ell_E^{1/2}J_E(u_P)\|^2_{L_2(E)} + \|\mathrm{diam}(\Omega_E)f\|^2_{L_2(\Omega_E)}, \tag{5.1}$$

is an indicator of the local error in $u - u_P$. One can show that their sum bounds the global error from above. More precisely, there exists an absolute constant A_3 such that

$$\|u - u_P\|^2 \leq A_3 \sum_{E \in \mathcal{E}_P} \eta_E.$$

We therefore define the global quantity

$$\Phi(P) := A_3 \sum_{E \in \mathcal{E}_P} \eta_E. \tag{5.2}$$

The second ingredient that we shall need is a bound for the global H^{-1} error in approximating f by piecewise constants in terms of a sum of local quantities. We have already introduced these quantities in §4.4. We recall the local error functional

$$e(\Delta) := |\Delta| \|f - f_\Delta\|^2_{L_2(\Delta)}, \tag{5.3}$$

where f_Δ is the average of f on Δ. We also have the global error

$$\bar{E}(f, P) := \sum_{\Delta \in P} e(\Delta), \tag{5.4}$$

which was used in algorithm **AA**.

The local error indicators from (5.1) together with (5.4) can be used to prove the following bounds for the Galerkin error (see [14] equations (2.7) and (2.8)):

$$A_1 \Phi(P) - A_2 \bar{E}(f, P) \leq \|u - u_P\|^2 \leq \Phi(P) \tag{5.5}$$

where $A_1, A_2 > 0$ are absolute constants.

To describe the main result in [14], we denote for each $\Delta \in T_*$ by $G(\Delta)$ the set of six triangular cells obtained from Δ when using a full refinement as described in the paragraph preceding Lemma 4.4. The following theorem is Theorem 3.1 of [14]:

Theorem 5.1 *Given $0 < \theta < 1$, there exist constants $0 < \tau < 1$ and $A_4 > 0$ with the following property. Let P be any admissible partition in $\mathcal{P}^a$ and suppose that $\mathcal{E}_0$ is a collection of edges from $\mathcal{E}_P$ such that*

$$A_3 \sum_{E \in \mathcal{E}_0} \eta_E \geq \theta \Phi(P), \tag{5.6}$$

and

$$\bar{E}(f, P) \leq A_4 \mu^2. \tag{5.7}$$

If P' is any admissible refinement of P that contains $G(\Delta)$ for every Δ which has an edge in common with $\mathcal{E}_0$, then either $\|u - u_{P'}\|^2 \leq \mu^2$ or

$$\|u - u_{P'}\|^2 \leq \tau \|u - u_P\|^2. \tag{5.8}$$

In other words, if the error in approximating f is small enough then the refinement strategy (5.6) guarantees an error reduction until the desired tolerance μ is met. Using this result, Morino, Nochetto, and Siebert build an AFEM which they prove converges whenever $f \in L_2(\Omega)$. We will give a modification of their algorithm in the following section.

In going further in this paper, we shall consider the values of θ and τ as fixed once and for all. Given a partition P and a marked set $\mathcal{E}_0$ of edges satisfying (5.6), we shall say that a partition P' which is a refinement of P has the G-property if it is admissible and P' contains all triangular cells in $G(\Delta)$ whenever Δ has an edge in $\mathcal{E}_0$.

Remark 5.2 *Subsequent complexity estimates will frequently refer to steps involving Galerkin approximations with respect to the partitions under consideration. Referring to suitable preconditioning techniques combined with nested iteration, we will always assume without further explicit mentioning that the computational cost of determining u_P stays proportional to $\#(P)$.*

6 Modifications of the MNS Algorithm

We shall make some modifications of the **MNS** algorithm in this section in order to prepare for its use in our new algorithm of the next section. Let us first note some properties of $\bar{E}$ and Φ. First of all if P' is a refinement of P then

$$\bar{E}(f, P') \leq \bar{E}(f, P). \tag{6.1}$$

This follows from the definition of $\bar{E}$ (4.14). Secondly, if $\bar{E}(f, P) \leq (A_1/2A_2)\Phi(P)$, then

$$\Phi(P) \leq A_5 |\!|\!| u - u_P |\!|\!|^2 \tag{6.2}$$

with $A_5 = 2/A_1$. This follows from the lower inequality in (5.5).

Our first modification of **MNS** which we shall call **MMNS** is to recast it in the form of marking and completion and still have the property that the refined partition contains $G(\Delta)$ (we need this to be able to obtain the guaranteed reduction in Theorem 5.1). Here are the properties of

MMNS $(P, \mu) \rightarrow P'$: *This algorithm takes as input an admissible partition P and an error tolerance μ with P satisfying $\bar{E}(f, P) \leq A_4\mu^2$ and produces a new partition $P' =$* **MMNS**(P, μ) *which satisfies either $|\!|\!| u - u_{P'} |\!|\!| \leq \mu$ or $|\!|\!| u - u_{P'} |\!|\!|^2 \leq \tau |\!|\!| u - u_P |\!|\!|^2$.*

The algorithm **MMNS** proceeds as follows. It marks all triangular cells which have an edge from $\mathcal{E}_0$ where $\mathcal{E}_0$ are the marked edges from **MNS**. Call this marked set $\mathcal{M}_0$. It performs a refinement and then a completion for $\mathcal{M}_0$ and P giving the new partition P'_1. It next marks any cells Δ' in P'_1 which have a proper descendent which is one of the cells in $G(\Delta)$, $\Delta \in \mathcal{M}_0$. Call this set of marked cells $\mathcal{M}_1$. We do a refinement and completion for $\mathcal{M}_1$ and P'_1 giving the new partition P'_2. We repeat this process one more time ending

with the partition $P' = P'_3$. We note that the number of subdivision and computations $N(\mathbf{MMNS}, P, \mu)$ needed in **MMNS** satisfies

$$N(\mathbf{MMNS}, P, \mu) \leq C_3 \#(P), \tag{6.3}$$

with C_3 an absolute constant.

Our next modifications of **MNS** center around two issues. The first is that we cannot compute $\|u - u_P\|$ for a partition P, we can only estimate this norm by using $\Phi(P)$. Therefore, we want to build an algorithm which will reduce Φ (i.e. we want $\Phi(P') < \Phi(P)$ for P' the refinement of P). The second issue is that we shall want a sufficiently large reduction in Φ when going from P to P'. This will be accomplished by the following algorithm which we call **REDUCE**. In this algorithm, there appears a constant A^* which will be specified later. Its role is to make sure that the term $\bar{E}(f, P')$ appearing in (5.5) is negligible compared to $\Phi(P')$ for all of the partitions P' that are encountered in the execution of the Algorithm.

REDUCE $(P, \mu, \mu') \to P'$: *This algorithm takes as input any tolerances* $0 < \mu' < \mu$ *and any admissible partition* P *for which we know*

$$\Phi(P) \leq \mu^2, \quad \bar{E}(f, P) \leq A^* \mu^2 \tag{6.4}$$

and produces a new partition $P' = \mathbf{REDUCE}(P, \mu, \mu')$ *which is a refinement of* P *and satisfies*

$$\Phi(P') \leq (\mu')^2, \quad \bar{E}(f, P') \leq A^* (\mu')^2. \tag{6.5}$$

We describe now the steps used to produce P'. We take $\mu'' = \mu' A_5^{-1/2}$ and use **MMNS** with the inputs P and μ''. The result is a partition $\tilde{P}_1 = \mathbf{MMNS}(P, \mu'')$ which is a refinement of P. We reapply **MMNS** with inputs μ'' and $\tilde{P}_1$ to obtain the output $\tilde{P}_2$. we repeat this k times where k is chosen as the smallest integer so that $\tau^k \leq (A_5)^{-1}(\mu'/\mu)^2$. We now apply **APPROX** with the input $\tilde{P}_k$ and $\sqrt{A^*}\mu'$ to get the partition $P' = \mathbf{REDUCE}(P, \mu, \mu')$.

Let us now check that P' satisfies (6.5). First of all, we know that the second inequality in (6.5) is valid because of the **APPROX** application. So we need to verify only the first inequality. Our first requirement on A^* is that $A^* \leq A_4$ so that Theorem 5.1 is applicable to each iteration, see (5.7). So we have two possibilities. The one is that $\|u - u_{\tilde{P}_j}\| \leq \mu''$ for one of the values of $j \in \{1, \ldots, k\}$. Since P' is a refinement of P_j, we have

$$\Phi(P') \leq A_5 \|u - u_{P'}\|^2 \leq A_5 \|u - u_{\tilde{P}_j}\|^2 \leq A_5 (\mu'')^2 \leq (\mu')^2, \tag{6.6}$$

which is the desired inequality. The other case is that the error is reduced by the factor τ at each application of **MMNS** which gives

$$\Phi(P') \leq A_5 \|u - u_{P'}\|^2 \leq A_5 \|u - u_{\tilde{P}_k}\|^2 \leq A_5 \tau^k \|u - u_P\|^2 \leq A_5 \tau^k \Phi(P) \leq (\mu')^2 \tag{6.7}$$

as desired.

Proposition 6.1 *The number of elements* $N(\mathbf{REDUCE}, P, \mu, \mu')$ *in* P' *is bounded by*

$$\#(P') \leq C_3^k(\#(P)) + N(\mathbf{APPROX}, \tilde{P}_k, \sqrt{A^*}\mu'), \tag{6.8}$$

where C_3 *is the constant from (6.3),* $\tilde{P}_k$ *is the partition produced by the kth application of* **MMNS**, *as described above, and* $k := \frac{2}{|\log\tau|}\log\frac{\mu}{\mu'} + \left|\frac{\log A_5}{\log\tau}\right|$. *The right side also bounds the number of flops needed to produce* P'.

Proof: By the above description of **REDUCE** we have

$$\#(P') \leq \#(\tilde{P}_k) + N(\mathbf{APPROX}, \tilde{P}_k, \sqrt{A^*}\mu') \leq C_3^k\#(P) + N(\mathbf{APPROX}, \tilde{P}_k, \sqrt{A^*}\mu'),$$

where we have used (6.3) and the fact that **MMNS** is applied a fixed number k of times. $\ominus$

7 The Main Loop

We are now in a position to build the main iteration of our adaptive algorithm. This loop will use the **REDUCE** algorithm to generate a partition P' and then follow it with a coarsening step whose sole purpose is to give a control on the size of the output partition. The description of this algorithm will serve to set the value of the constants $\alpha, \beta, \gamma, A^*$ which have appeared earlier but were left unspecified. We call the algorithm for the main loop **MAIN**.

MAIN $(P, \mu) \to \tilde{P}$: *This algorithm takes as input a tolerance* μ *and an admissible partition* P *satisfying*

$$\Phi(r(P)) \leq \mu^2 \tag{7.1}$$

and

$$\bar{E}(f, P) \leq A^*\mu^2 \tag{7.2}$$

with A^* *the constant appearing in* **REDUCE**. *It outputs an admissible partition* $\tilde{P} :=$ **MAIN** (P, μ) *that satisfies*

$$\Phi(r(\tilde{P})) \leq \mu^2/2 \tag{7.3}$$

and

$$\bar{E}(f, \tilde{P}) \leq A^*\mu^2/2. \tag{7.4}$$

We now describe the steps in **MAIN**.

STEP1: *Apply* **REDUCE** *with inputs* $r(P)$, μ, *and* $\mu' = \sqrt{\alpha}\mu$ *with* α *the constant in* **COARSE**, *see (4.44). The output* $P' =$ **REDUCE** $(r(P), \mu, \sqrt{\alpha}\mu)$ *is an admissible partition which satisfies (6.5).*

STEP2: *Apply* **COARSE** *with inputs* P, P' *and* μ *and obtain the output* $\bar{P} =$ **COARSE**(P, P', μ) *which, on account of (4.64), satisfies*

$$\Phi(r(\bar{P})) \leq A_5\|u - u_{r(\bar{P})}\|^2 \leq A_5\beta\mu^2. \tag{7.5}$$

STEP 3: *Apply* **APPROX** *with input* $\bar{P}$ *and tolerance* $(A^*/2)^{1/2}\mu$ *and obtain as output the partition* $\tilde{P} = \mathbf{APPROX}(\bar{P}, (A^*/2)^{1/2}\mu)$ *which satisfies* $\bar{E}(f, \tilde{P}) \leq A^*\mu^2/2$.

Already to guarantee these steps we need to impose conditions on the constants. Namely, in **STEP 2** to obtain the first inequality in (7.5), we need that

$$A^* \leq (A_1/2A_2)\alpha \tag{7.6}$$

in order to guarantee that $\bar{E}$ is negligible compared to Φ for the partitions created in the algorithms **REDUCE** and **COARSE** (see (6.2)). We impose this condition on A^* which serves to fix its value once α is specified (as it will be in a moment).

Let us now check that the conclusions stated in **MAIN** are indeed valid. The condition (7.4) is an obvious consequence of **STEP 3** in **MAIN**. Since $\tilde{P}$ is an admissible refinement of $\bar{P}$, we infer from (7.5) and (6.2) that

$$\Phi(r(\tilde{P})) \leq A_5 \|\!| u - u_{r(\tilde{P})} |\!\|^2 \leq A_5 \|\!| u - u_{r(\bar{P})} |\!\|^2 \leq A_5 \beta \mu^2. \tag{7.7}$$

The estimate (7.3) follows from (7.7) provided $A_5^2\beta \leq 1/2$. We impose this condition and thereby specify the value of β as

$$\beta = 1/(2A_5^2). \tag{7.8}$$

This in turn specifies the value of α and γ when we impose our earlier requirement (4.63). Thus, recalling (7.6), admissible choices for α, A^* and γ are

$$\alpha := 1/(8A_5^2), \quad \gamma := 1/(8C_4A_5^2), \quad A^* = A_1/(16A_2A_5^2) \tag{7.9}$$

where C_4 is the constant from (4.52). So now all constants have been specified.

Proposition 7.1 *The number of flops* $N(\mathbf{MAIN}, P, \mu)$ *used in computing* $\tilde{P}$ *is bounded by*

$$N(\mathbf{MAIN}, P, \mu) \leq C(\#(P)) + N(\mathbf{APPROX}, P, \sqrt{A^*\alpha}\mu) + N(\mathbf{APPROX}, \bar{P}, \sqrt{A^*/2}\mu), \tag{7.10}$$

where $\bar{P}$ *is the output of* **STEP 2** *and* C *is an absolute constant. In the case that* $f \in \bar{\mathcal{A}}^s(H^{-1})$ *and* $u \in \dot{\mathcal{A}}^s$ *then we have the inequality*

$$\#(\tilde{P}) - \#(P) \leq C_2(s)(\|f\|_{\bar{\mathcal{A}}^s}^{1/s} + \|u\|_{\dot{\mathcal{A}}^s}^{1/s})\mu^{-1/s} \tag{7.11}$$

and

$$N(\mathbf{MAIN}, P, \mu) \leq C_5(s)\left\{\#(P) + (\|f\|_{\bar{\mathcal{A}}^s}^{1/s} + \|u\|_{\dot{\mathcal{A}}^s}^{1/s})\mu^{-1/s}\right\} \tag{7.12}$$

with $C_2(s)$ *and* $C_5(s)$ *constants depending only on* s.

Proof: The application of **REDUCE** in **STEP1** of **MAIN** has, by Proposition 6.1, (6.8), a computational complexity bounded by

$$N(\mathbf{REDUCE}, P, \mu, \sqrt{\alpha}\mu) \leq C(\#(P)) + N(\mathbf{APPROX}, P, \sqrt{A^*\alpha}\mu).$$

Note that C is an absolute constant since, once α being fixed, the number k of calls of **MMNS** in **REDUCED** remains uniformly bounded. Therefore, the output $\tilde{P}_k$ in the last call of **MMNS** satisfies $\#(\tilde{P}_k) \leq C(\#(P))$. By (4.65) the cost of **STEP2** remains proportional to the above bound. The estimate (7.10) follows now from the last call of **APPROX** in **STEP3**.

Now suppose that $f \in \bar{\mathcal{A}}^s(H^{-1})$ and $u \in \dot{\mathcal{A}}^s$. By (4.36) and the above observations the number of operations required in **STEP1** remains bounded by $C(\#(P) + \|f\|^{1/s}_{\bar{\mathcal{A}}^s}\mu^{-1/s})$ where C is an absolute constant. Moreover, note that (7.12) follows from (7.11)

Thus it remains to verify (7.11). From (4.66) we know that

$$\#(\bar{P}) \leq \#(P) + C_2(s)\|u\|^{1/s}_{\dot{\mathcal{A}}^s}\mu^{-1/s}, \tag{7.13}$$

while from (4.36) we obtain

$$\#(\tilde{P}) \leq \#(\bar{P}) + C_2(s)\|u\|^{1/s}_{\dot{\mathcal{A}}^s}(A^*/2)^{-1/2s}\mu^{-1/s}. \tag{7.14}$$

These two estimates combine to verify (7.11). $\ominus$

8 An adaptive algorithm with coarsening

In this section, we shall formulate our adaptive algorithm for solving (1.1) and prove the convergence properties of this algorithm. We shall call this algorithm **ALG**. We shall use the following notation consistently in this section.

ALG $(\varepsilon, P_0) \to (P, u_P)$: *This algorithm takes as input a desired tolerance $\epsilon > 0$ and an initial partition P_0 and outputs a partition P and the corresponding Galerkin approximation u_P which satisfies*

$$\Phi(P) \leq \epsilon, \tag{8.1}$$

(and hence, by (5.5), $\|u - u_P\|^2 \leq \varepsilon$), using the following steps:
(I): *(Initialization) For the initial partition P_0, compute $\Phi(P_0)$. If $\Phi(P_0) \leq \epsilon$ then stop the algorithm and output $P := P_0$. Otherwise, define $\epsilon_0 := A_5\Phi(P_0)$ and compute $P_0' :=$* **APPROX**$(f, P_0, \sqrt{A^*\varepsilon_0})$*, where A_5 is the constant from (6.2). The partition P_0' then satisfies*

$$\Phi(r(P_0')) \leq A_5\|u - u_{r(P_0')}\|^2 \leq A_5\|u - u_{P_0}\|^2 \leq A_5\Phi(P_0) = \epsilon_0, \tag{8.2}$$

and

$$\bar{E}(f, P_0') \leq A^*\epsilon_0. \tag{8.3}$$

Set $k = 0$ and proceed to Step **(II)**.
(II): *For the current value of k, apply* **MAIN** *with inputs P_k' and $\sqrt{\epsilon_k}$, satisfying $\Phi(r(P_k')) \leq \varepsilon_k$ and $\bar{E}(f, P_k') \leq A^*\varepsilon_k$, to obtain the partition $P_{k+1}' =$* **MAIN**$(P_k', \sqrt{\epsilon_k})$*. Compute $\Phi(r(P_{k+1}'))$ which by (7.3) in* **MAIN** *is $\leq \epsilon_k/2$.*
(III): *If $\Phi(r(P_{k+1}')) \leq \epsilon$ then stop and output $P = r(P_{k+1}')$. If not compute $\bar{E}(f, P_{k+1}')$ (which is $\leq A^*\varepsilon_k/2$) and define*

$$\varepsilon_{k+1} := \max\{\Phi(r(P_{k+1}')), \bar{E}(f, P_{k+1}')/A^*\}$$

which is $\leq \varepsilon_k/2$. *Set* $k+1$ *to* k *and go to* **(II)**.

The following is the main result of this paper.

Theorem 8.1 *For any function* $f \in H^{-1}(\Omega)$ *and any* $\epsilon > 0$, **ALG** *produces a partition* P *for which*

$$\Phi(P) \leq \epsilon. \tag{8.4}$$

If $s > 0$ *and* $u \in \dot{\mathcal{A}}^s$, *and* $f \in \dot{\bar{\mathcal{A}}}^s$, *then*

$$\#(P) \leq \#(P_0) + C(s)(\|f\|^{1/s}_{\dot{\bar{\mathcal{A}}}^s} + \|u\|^{1/s}_{\dot{\mathcal{A}}^s})\epsilon^{-1/s} \tag{8.5}$$

with $C(s) > 0$ *a constant depending only on* s *and the initial partition* P_0. *Moreover, The number of computations used in producing* P *does not exceed* $C(s)(\#(P_0) + \|f\|^{1/s}_{\dot{\bar{\mathcal{A}}}^s} + \|u\|^{1/s}_{\dot{\mathcal{A}}^s})\epsilon^{-1/s}$.

Proof: The conclusion (8.4) follows immediately from the stopping criteria.

To verify (8.5), suppose that $s > 0$ and $u \in \dot{\mathcal{A}}^s$, and $f \in \dot{\bar{\mathcal{A}}}^s$. To start out with, we know from (4.36) that

$$\#(P_0') \leq \#(P_0) + C_2(s)\|f\|^{1/s}_{\dot{\bar{\mathcal{A}}}^s}(\epsilon_0)^{-1/s}. \tag{8.6}$$

From (7.11) we know that

$$\#(P_{k+1}') \leq \#(P_k') + C_5(s)\left(\|u\|^{1/s}_{\dot{\mathcal{A}}^s}(\epsilon_k)^{-1/s} + \|f\|^{1/s}_{\dot{\bar{\mathcal{A}}}^s}(\epsilon_k)^{-1/s}\right). \tag{8.7}$$

Thus, iterating this inequality and taking the exponential decay of the ε_k into account, we arrive at

$$\begin{aligned} \#(P_n') &\leq \#(P_0) + C(s)(\|f\|^{1/s}_{\dot{\bar{\mathcal{A}}}^s} + \|u\|^{1/s}_{\dot{\mathcal{A}}^s}) \sum_{k=0}^{n-1} (\epsilon_k)^{-1/s} \\ &\leq \#(P_0) + C(s)(\|f\|^{-1/s}_{\dot{\bar{\mathcal{A}}}^s} + \|u\|^{-1/s}_{\dot{\mathcal{A}}^s})\epsilon^{-1/s}, \end{aligned} \tag{8.8}$$

since $\varepsilon_{n-1} > \varepsilon$. The bound on the computational complexity follows in the same fashion from (4.35), (4.36) and Proposition 7.1. ⊙

Appendix

Recall from (4.7) and (4.16) that the classes $\mathcal{A}^s(H^1(\Omega))$ and $\bar{\mathcal{A}}^s(H^{-1}(\Omega))$ are determined through the nonlinear approximation properties of their elements. For instance, the elements in $\mathcal{A}^s(H^1(\Omega))$ can be approximated in $H^1(\Omega)$ to accuracy ε on admissible partitions with the order of $\varepsilon^{-1/s}$ cells. Obviously not all of these elements can be approximated on uniform partitions with the same accuracy. In this section we wish to explain which properties make a function belong to $\mathcal{A}^s(H^1(\Omega))$, say. This amounts to relating the above approximation classes to *regularity*. The following results about the behavior of $\sigma_n(u)$, defined by (4.3), are in principle known. They invoke the Besov smoothness of functions.

We refer the reader to any of the standard treatments of Besov spaces (e.g.[16, 1, 7]) for the definition of the Besov spaces $B_q^s(L_\tau(\Omega))$ and only remark here that such a space is a smoothness space consisting of functions with smoothness order s (number of derivatives) measured in L_τ. For example $B_2^s(L_2)$ is identical with H^s and $B_\infty^s(L_\tau)$ is a Lipschitz space in L_τ whenever s is not an integer. The role of q is secondary and only serves to give a fine grading of the spaces important in many applications such as embedding theorems.

The results about σ_n, defined by (4.3), can be formulated as follows.

Theorem 8.2 *If $u \in B_\tau^{\alpha+1}(L_\tau(\Omega))$ with $0 \le \alpha \le 1$ and $1/\tau < (\alpha+1)/2$, then*

$$\sigma_n(u) = \inf_{P \in \mathcal{P}_n} \inf_{S \in \mathcal{S}_P} \|u - S\| \le C_0 n^{-\alpha/2} \|u\|_{B_\tau^{\alpha+1}(L_\tau))},$$

where the constant C_0 depends on the discrepancy $\delta := \frac{\alpha+1}{2} - \frac{1}{\tau}$ when δ tends to zero.

A few brief comments on the range of the involved parameters are in order.

Remark 8.3 *The restriction on α arises because we are approximating in H^1 using piecewise linears and so $\alpha + 1 \le 2$. Thus in two spatial dimensions $N^{-1/2}$ is the highest attainable order in the class $B_\tau^2(L_\tau))$ with $\tau^{-1} < 1$. The restriction on τ arises from the Sobolev embedding theorem. It guarantees that the Besov space is embedded compactly in H^1. When $\tau^{-1} > (\alpha+1)/2$, this Besov space is no longer embedded in H^1.*

The above regularity assumptions are only sufficient for u to belong to $\mathcal{A}^{\alpha/2}$, say. The following *inverse theorem* shows that, although this is not a complete characterization, it is sharp in the following sense.

Theorem 8.4 *If $u \in H^1(\Omega)$ satisfies $\sigma_n(u) \le Cn^{-\alpha/2}$ then $u \in B_\tau^{\alpha+1}(L_\tau)$ with $\tau^{-1} = (\alpha+1)/2$.*

The proofs of Theorems 8.2, 8.4 will be given elsewhere. For related results see [11].

Finally, as a consequence of known results on the metric entropy of unit balls of Besov classes, the order of Besov smoothness limits the approximation order in the following sense.

Theorem 8.5 *For each $0 \le \alpha \le 1$ and $\tau^{-1} < (\alpha+1)/2$ we have*

$$\sup_{f \in U(B_\tau^{\alpha+1}(L_\tau(\Omega)))} \sigma_n(u) \ge CN^{-\alpha/2} \tag{8.9}$$

with $C > 0$ an absolute constant. (Here $U(X)$ denotes the unit ball of a normed space X).

The above results should be read as follows. First note that for fixed β the smoothness measure given by the space $B_\tau^\beta(L_\tau))$ becomes weaker when τ decreases. Theorem 8.2 says that for fixed α the loss of regularity incurred by decreasing τ towards the critical value $2/(\alpha+1)$ can be compensated by *nonlinear approximation* so as to retain the order $N^{-\alpha/2}$ of approximation in H^1. Theorem 8.5 says that this order is best possible with respect to the full unit ball in the respective Besov class. To obtain the same approximation

order through spaces on (quasi-) uniform meshes is equivalent to u to belong to the much smaller space $B_\infty^{\alpha+1}(L_2)$ (which is close to $H^{\alpha+1}$).

Thus whenever the solution u has sufficient regularity measured in L_2 the best possible balance of accuracy versus the number of degrees of freedom can be obtained at least asymptotically by using quasi-uniform meshes. Nonlinear approximation provides asymptotically better rates whenever u has a higher regularity in L_τ for $\tau < 2$. Theorem 8.1 combined with Theorem 8.2 then say that this better rate is actually recovered by the above adaptive algorithm which is a special instance of a nonlinear process. It has recently be shown in [9, 8] that, depending on the smoothness of the domain Ω, the solution to Poisson's equation indeed has typically higher Besov than Sobolev regularity in the sense that

$$\alpha^* := \sup\{\alpha : u \in B_\tau^{\alpha+1}(L_\tau)), \tau^{-1} = (\alpha+1)/2\} > \beta^* := \sup\{\beta : u \in B_\infty^{\beta+1}(L_2))\}.$$

Hence in those cases the use of the adaptive scheme pays by a better asymptotic work/accuracy rate.

References

[1] J. Bergh, J. Löfström, *Interpolation Spaces, An Introduction*, Springer, 1976.

[2] P. Binev, and R. DeVore, Fast computation in tree approximation, preprint

[3] I. Babuska and M. Vogelius, *Feedback and adaptive finite element solution of one dimensional boundary value problems*, Numer. Math., 44 (1984), pp. 75-102.

[4] I. Babuška, A. Miller, *A Feedback Finite Element method with A Posteriori Error Estimations: Part I. The Finite Element method and Some Basic Properties of the A Posteriori Error Estimator* , Comput. Methods Appl. Mech. Engrg., 61(1987), pp. 1-40.

[5] A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods for elliptic operator equations - convergence rates*, Math. Comp. 70(2001), pp. 27-75.

[6] A. Cohen, W. Dahmen, and R. DeVore, *Adaptive wavelet methods II: least squares formulations* , Foundations of Comp. Math, 2002, to appear.

[7] R. DeVore, and R. Sharpley, Besov spaces on domains in $\mathbb{R}^d$, TAMS, preprint

[8] S. Dahlke: Besov regularity for elliptic boundary value problems on polygonal domains, Appl. Math. Lett., 12 (1999), 31–36.

[9] S. Dahlke, R. DeVore, Besov regularity for elliptic boundary value problems, Comm. Partial Differential Equations, 22 (1997), 1–16.

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

[11] B. Koraivanov, P. Petrushev, Nonlinear piecewise polynomial approximation beyond Besov spaces, IMI Preprint Series 2001:13, University of South Carolina, 2001.

[12] L. Lovász, M. D. Plummer, *Matching Theory*, Akadémiai Kiadó, Budapest, 1986.

[13] W.F. Mitchell, *A comparison of adaptive refinement techniques for elliptic problems*, ACM Transaction on Math. Software, **15**(1989), 326–347.

[14] P. Morin, R. Nochetto, K. Siebert, *Data Oscilation and Convergence of Adaptive FEM*, SIAM J. Numer. Anal., 38 (2000), pp. 466-488.

[15] J. Petersen, Die Theorie der regulären Graphen, Acta Math. 15(1891), 193–220.

[16] H. Triebel, *Interpolation Theory, Function Spaces, and Differential Operators*, North-Holland, Amsterdam, 1978.

[17] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley-Teubner, Chichester, 1996.

Peter G. Binev
Department of Mathematics
University of South Carolina
Columbia, SC 29208
U.S.A.
e–mail: binev@math.sc.edu
WWW: http://www.math.sc.edu/∼binev/
Tel: 803-576-6304, Fax: 803-777-6527

Wolfgang Dahmen
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
e–mail: dahmen@igpm.rwth-aachen.de
WWW: http://www.igpm.rwth-aachen.de/∼dahmen/
Tel: 49-241-8093950

Ronald A. DeVore
Department of Mathematics
University of South Carolina
Columbia, SC 29208
U.S.A.
e–mail: devore@math.sc.edu
WWW: http://www.math.sc.edu/∼devore/
Tel: 803-777-6323, Fax: 803-777-6527