Adaptive Wavelet Methods for Elliptic Operator Equations – Convergence Rates*

Albert Cohen, Wolfgang Dahmen, Ronald DeVore

October 6, 1999

Abstract

This paper is concerned with the construction and analysis of wavelet-based adaptive algorithms for the numerical solution of elliptic equations. These algorithms approximate the solution u of the equation by a linear combination of N wavelets. Therefore, a benchmark for their performance is provided by the rate of best approximation to u by an arbitrary linear combination of N wavelets (so called N-term approximation), which would be obtained by keeping the N largest wavelet coefficients of the real solution (which of course is unknown). The main result of the paper is the construction of an adaptive scheme which produces an approximation to u with error $O(N^{-s})$ in the energy norm, whenever such a rate is possible by N-term approximation. The range of $s > 0$ for which this holds is only limited by the approximation properties of the wavelets together with their ability to compress the elliptic operator. Moreover, it is shown that the number of arithmetic operations needed to compute the approximate solution stays proportional to N. The adaptive algorithm applies to a wide class of elliptic problems and wavelet bases. The analysis in this paper puts forward new techniques for treating elliptic problems as well as the linear systems of equations that arise from the wavelet discretization.

AMS subject classification: 41A25, 41A46, 65F99, 65N12, 65N55.
Key Words: Elliptic operator equations, quasi sparse matrices and vectors, best N-term approximation, fast matrix vector multiplication, thresholding, adaptive space refinement, convergence rates.

1 Introduction

1.1 Background

Adaptive methods, such as adaptive Finite Elements Methods (FEM), are frequently used to numerically solve elliptic equations when the solution is known to have singularities.

*This work has been supported in part by the Deutsche Forschungsgemeinschaft grants Da 117/8–2, the Office of Naval Research Contract N0014-91-J1343 and the TMR network "Wavelets in Numerical Simulation"

A typical algorithm uses information gained during a given stage of the computation to produce a new mesh for the next iteration. Thus, the adaptive procedure depends on the current numerical resolution of u. Accordingly, these methods produce a form of *nonlinear* approximation of the solution, in contrast with *linear* methods in which the numerical procedure is set in advance and does not depend on the solution to be resolved.

The motivation for adaptive methods is that they provide flexibility to use finer resolution near singularities of the solution and thereby improve on the approximation efficiency. Since the startling papers [2, 3] the understanding and practical realization of adaptive refinement schemes in a finite element context has been documented in numerous publications [3, 4, 5, 10, 31]. A key ingredient in most adaptive algorithms are *a-posteriori error estimators* or *indicators* derived from the current residual or the solution of local problems. They consist of local quantities such as jumps of derivatives across the interface between adjacent triangles or simplices. One often succeeds in bounding the (global) error of the current solution with respect to the energy norm, say, by sums of these quantities from *below and above*. Thus refining the mesh where these local quantities are large is *hoped* to reduce the bounds and hence the error in the next computation. Computational experience frequently confirms the success of such techniques for elliptic boundary value problems in the sense that adaptively generated highly nonuniform meshes indeed give rise to an accuracy that would require the solution of much larger systems of equations based on uniform refinements. However, on a rigorous level the quantitative gain of adaptive techniques is usually not clear. The central question is whether the mesh refinements *actually* result, at each step, in some *fixed* error reduction. To our knowledge only in [30] convergence of an adaptive scheme has been established for a rather special case namely a piecewise linear finite element discretization of the classical Dirichlet problem for Laplace's equation. There is usually no *rigorous* proof of the overall *convergence* of such schemes unless one assumes some quantitative information such as the *saturation property* about the unknown solution [10]. Saturation properties are *assumed but not proven* to hold.

Moreover, the derivation of error indicators in conventional discretizations hinges on the *locality* of differential operators. Additional difficulties are therefore encountered when considering elliptic operators with *nonlocal* Schwartz kernel arising, for instance, in connection with boundary integral equations.

In summary there seem to be at least two reasons for this state of affairs: (i) There is an inherent difficulty even for local operators in utilizing the information available at a given stage in the adaptive computation to *guarantee* that a suitable reduction will occur in the residual error during the next adaptive step. (ii) Finite element analysis is traditionally based on *Sobolev regularity* (see e.g. [11] or [12]) which is known to govern the performance of linear methods. Only recent developments in the understanding of nonlinear methods have revealed that *Besov regularity* is a decidedly different and more appropriate smoothness scale for the analysis of adaptive schemes, see e.g. [26].

In view of the significant computational overhead and severe complications caused by handling appropriate data structures for adaptive schemes, not only *guaranteeing convergence* but above all knowing its *speed* is of paramount importance for deciding whether or under which circumstances adaptive techniques actually pay off. To our knowledge

nothing is known so far about the actual *rate* of convergence of adaptive FEM solvers by which we mean the relation between the accuracy of the approximate solution and the involved degrees of freedom, or better yet the number of arithmetic operations.

1.2 Wavelet methods

An alternative to FEM are wavelet based methods. Similarily to mesh refinement in FEM, these methods offer the possibility to compress smooth functions with isolated singularities, into high-order adaptive approximations involving a small number of basis functions. In addition, it has been recognized for some time [8] that for a large class of operators (including integral operators) wavelet bases give rise to matrix representations that are quasi-sparse (see §§2-3 for a definition of quasi-sparse) and admit simple diagonal preconditioners in the case of elliptic operators. Therefore, it is natural to develop adaptive strategies based on wavelet discretizations in order to solve numerically elliptic operator equations.

The state of wavelet-based solvers is still in its infancy, and certain inherent impediments to their numerical use remain. These are mainly due to the difficulty of dealing with realistic domain geometries. Nevertheless, these solvers show great promise, especially for adaptive approximation (see e.g.[1, 9, 13, 15, 20]) . Most adaptive strategies exploit the fact that wavelet coefficients convey detailed information on the local regularity of a function and thereby allow the detection of its singularities. The rule of thumb is that wherever wavelet coefficients of the currently computed solution are large in modulus additional refinements are necessary. In some sense, this amounts to using the size of the computed coefficients as local a-posteriori error indicators. Note that here *refinement* has a somewhat different meaning than in the finite element setting. There the adapted spaces result from refining a *mesh*. The mesh is the primary controlling device and may create its own problems (of geometric nature) that have nothing to do with the underlying analytic task. In the wavelet context refinement means to *add* suitably selected further basis functions to those that are used to approximate the current solution. We refer to this as *space refinement*.

In spite of promising numerical performances, the problem remains (as in the finite element context) to *quantify* these strategies, that is, to decide *which* and *how many* additional wavelets need to be added in a refinement step in order to guarantee a *fixed* error reduction rate at the next resolution step. An adaptive wavelet scheme based on a-posteriori error estimators has been recently developed in [17], which ensures this fixed error reduction for a wide class of elliptic operators including those of negative order. This shows that making use of the characteristic features of wavelet expansions, such as the sparsification and preconditioning of elliptic operators, allows one to go beyond what is typically known in the conventional framework of adaptive FEM. However, similar to FEM, there are so far *no* results about the rate of convergence of adaptive wavelet based solvers, i.e., the dependence of the error on the number of degrees of freedom.

1.3 The objectives

The purpose of the present paper is twofold. Firstly, we provide analytical tools that can be utilized in studying the theoretical performance of adaptive algorithms. Secondly, we show how these tools can be used to construct and analyze wavelet based adaptive algorithms which display optimal approximation and complexity properties in the sense that we describe below.

The adaptive methods we analyze in this paper take the following form. We assume that we have in hand a wavelet basis $\{\psi_\lambda\}_{\lambda\in\nabla}$ to be used for numerically resolving the elliptic equation. Our adaptive scheme will iteratively produce finite sets $\Lambda_j \subset \nabla$, $j = 1, 2, \ldots$, and the Galerkin approximation u_{Λ_j} to u from the space $S_{\Lambda_j} := \mathrm{span}(\{\psi_\lambda\}_{\lambda\in\Lambda_j})$. The function u_{Λ_j} is a linear combination of $N_j := \#\Lambda_j$ wavelets. Thus the adaptive method can be viewed as a particular form of nonlinear N-term wavelet approximation and a benchmark for the performance of such an adaptive method is provided by comparison with *best N-term approximation* (in the energy norm) when full knowledge of u is available.

Much is known about N-term approximation. In particular, there is a characterization of the functions v that can be approximated in the energy norm with accuracy $O(N^{-s})$ by using linear combinations of N wavelets. As we already mentioned this class B^s is typically a Besov space, which is substantially larger than the corresponding Sobolev space W^s which ensures $O(N^{-s})$ accuracy for *uniform* discretization with N parameters. In several instances of the elliptic problems, e.g. when the right hand side f has singularities, or when the boundary of Ω has corners, the Besov regularity of the solution will exceed its Sobolev regularity (see [16] and [18]). So these solutions can be approximated better by best N-term approximation than by uniformly refined spaces and the use of adaptive methods is suggested. Another important feature of N-term approximation is that a near best approximation is produced by *thresholding*, i.e., simply keeping the N largest contributions (measured in the same metric as the approximation error) of the wavelet expansion of v.

Of course, since best N-term approximation requires complete information on the approximated function it cannot be applied directly to the unknown solution. It is certainly not clear beforehand whether at all a concrete numerical scheme can produce at least asymptotically the same convergence rate. Thus ideally an *optimal* adaptive wavelet algorithm should produce a similar result as thresholding the exact solution. In more quantitative terms this means whenever the solution u is in B^s, the approximations u_{Λ_j} should satisfy

$$\|u - u_{\Lambda_j}\| \leq C\|u\|_{B^s} N_j^{-s}, \quad N_j := \#\Lambda_j, \tag{1.1}$$

where $\|\cdot\|$ is the energy norm and $\|\cdot\|_{B^s}$ is the norm for B^s. Since in practice one is mostly interested in controlling a prescribed accuracy with a minimal number of parameters, we shall rather say that the adaptive algorithm is of *optimal order* $s > 0$ if whenever the solution u is in B^s, then for all $\epsilon > 0$, there exists $j(\epsilon)$ such that

$$\|u - u_{\Lambda_j}\| \leq \epsilon, \quad j \geq j(\epsilon), \tag{1.2}$$

and such that

$$\#(\Lambda_{j(\epsilon)}) \leq C\|u\|_{B^s}^{1/s}\epsilon^{-1/s}. \tag{1.3}$$

Such a property ensures an optimal *memory size* for the description of the approximate solution.

Another crucial aspect of the adaptive algorithms is their computational complexity: we shall say that the adaptive algorithm is *computationally optimal* if, in addition to ((1.2)-(1.3)), the number of arithmetic operation needed to derive u_{Λ_j} is proportional to $\#\Lambda_j$. Note that an instance of computational optimality in the context of linear methods is provided by the *full multigrid* algorithm when N represents the number of unknowns necessary to achieve a given accuracy on a uniform grid. We are thus interested in algorithms that exhibit the same type of computational optimality with respect to an optimal adaptive grid which is not known in advance and should itself be generated by the algorithm.

The main accomplishment of this paper is the development of an adaptive numerical scheme which for a wide class of operator equations (including those of negative order) is optimal with regard to best N-term approximation and also computationally optimal in the above sense.

1.4 Organization of the paper

In §2, we introduce the general setting of elliptic operator equations where our results apply. In this context, after applying a diagonal preconditioner, wavelet discretizations allow us to view the equation as a discrete *well conditioned* ℓ_2 linear system.

In § 3, we review certain aspects of nonlinear approximation, quasi-sparse matrices and fast multiplication using such matrices. The main result of this section is an algorithm for the fast computation of the application of a quasi-sparse matrix to a vector.

In § 4, we analyze the rate of convergence of the refinement procedure introduced earlier in [17]. We will refer to this scheme here as Algorithm I. We show that this algorithm is optimal for a small range of $s > 0$. However, the *full range* of optimality should be limited only by the properties of the wavelet basis (smoothness and vanishing moments) and the operator which is not the case for Algorithm I. The analysis in § 4 identifies however the barrier that keeps Algorithm I from being optimal in the full range of s.

In § 5, we introduce a second strategy – Algorithm II – for adaptively generating the sets Λ_j that is shown to provide optimal approximation of order $s > 0$ for the *full range* of s. The new ingredient that distinguishes Algorithm II from Algorithm I is the addition of thresholding steps which delete some indices from Λ_j. This would be the analogue of coarsening the mesh in FEM.

Although we have qualified so far both procedures in § 4 and § 5 as "algorithms", we have actually ignored any issue concerning practical realization. They are idealized in the sense that the exact assessment of residuals and Galerkin solutions is assumed. This was done in order to identify clearly the essential analytical tasks. Practical realizations require truncations and approximations of these quantities. § 6 is devoted to developing

the ingredients of a realistic numerical scheme. This includes quantitative thresholding procedures, approximate matrix/vector multiplication, approximate Galerkin solvers and the approximate evaluation of residuals.

In § 7 we employ these ingredients to formulate a computable version of Algorithm II which is shown to be computationally optimal for the full range of s. Recall that this means that it realizes for this range the order of best N-term approximation at the expense of a number of arithmetic operations that stays proportional to the number N of significant coefficients. Computational optimality hinges to a great extent on the fast approximate matrix/vector multiplication from § 3.

It should be noted however that an additional cost in our wavelet adaptive algorithm is incurred by sorting the coefficients in the currently computed solution. This cost at stage j is of order $N \log N$ where $N = \#\Lambda_j$, thus slightly larger than the cost in arithmetic operations. It should be stressed that the complexity of the algorithm is analysed under the assumption that the solution exhibits a certain rate of best N-term approximation which is, for instance, implied by a certain Besov regularity. The algorithm itself does *not* require any a-priori assumption of that sort.

We have decided to carry out the (admittedly more technical) analysis of the numerical ingredients in some detail in order to substantiate our claim that the optimality analysis is not based on any hidden assumptions (beyond those hypotheses that are explicitly stated) such as accessing infinitely many data. Nevertheless the main message of this paper can be read in § 4 and § 5: optimal adaptive approximations of elliptic equations can be computed by iterative wavelet refinements using a-posteriori error estimators, provided that the computed solution is regularly updated by appropriate thresholding procedures. This fact was already suggested by numerical experiments in [15] that show similar behavior between the numerical error generated by such adaptive algorithms and by thresholding the exact solution.

2 The Setting

In this section, we shall introduce the setting in which our results apply. In essence, our analysis applies whenever the elliptic operator equation takes place on a manifold or domain which admits a biorthogonal wavelet basis.

2.1 Ellipticity Assumptions

This subsection gives the assumptions we make on the operator equation to be numerically solved. These assumptions are quite mild and apply in great generality.

Let Ω denote a bounded open domain in the Euclidean space $\mathbb{R}^d$ with Lipschitz boundary or, more generally, a Lipschitz manifold of dimension d. In particular, Ω could be a closed surface which arises as a domain for a boundary integral equation. The space $L_2(\Omega)$ consists of all square integrable functions on Ω with respect to the (canonically

induced) Lebesgue measure. The corresponding inner product is denoted by

$$\langle \cdot, \cdot \rangle_{L_2(\Omega)}. \tag{2.1}$$

Let A be a linear operator mapping a Hilbert space H into H^* (its dual relative to the pairing $\langle \cdot, \cdot \rangle_{L_2(\Omega)}$) where H is a space with the property that either H or its dual H^* is embedded in $L_2(\Omega)$. The operator A induces the bilinear form a defined on $H \times H$ by

$$a(u, v) := \langle Au, v \rangle, \tag{2.2}$$

where $\langle \cdot, \cdot \rangle$ denotes the (H^*, H) duality product.

(A1): We assume that the bilinear form a is symmetric positive definite and *elliptic* in the sense that

$$a(v, v) \sim \|v\|_H^2, \quad v \in H. \tag{2.3}$$

Here, and throughout this paper, $\sim$ means that both quantities can be uniformly bounded by constant multiples of each other. Likewise $\lesssim$ indicates inequalities up to constant factors.

It follows that H is also a Hilbert space with respect to the inner product a and that this inner product induces an equivalent norm (called the energy norm) on H by

$$\| \cdot \|_a^2 := a(\cdot, \cdot). \tag{2.4}$$

By duality, A thus defines an isomorphism from H onto H^*. We shall study the equation

$$Au = f \tag{2.5}$$

with $f \in H^*$. From our assumptions, it follows that for any $f \in H^*$, this equation has a unique solution in H, which will always be denoted by u. This is also the unique solution of the variational equation

$$a(u, v) = \langle f, v \rangle, \ \text{ for all } \ v \in H. \tag{2.6}$$

The typical examples included in the above assumptions are Poisson's, Helmholtz or the biharmonic equations on bounded domains in $\mathbb{R}^d$; single or double layer potentials and hypersingular operators on closed surfaces arising in the context of boundary integral equations. In these examples H is a Sobolev space, e.g. $H = H_0^1(\Omega)$, $H_0^2(\Omega)$, or $H = H^{-1/2}(\Omega)$; see [19, 17, 36] for examples.

2.2 Wavelet Assumptions

By now wavelet bases are available for various types of domains that are relevant for the formulation of operator equations. This covers, for instance, polyhedral surfaces of dimension two and three [24] as well as manifolds or domains that can be represented as

a disjoint union of smooth regular parametric images of a simple parameter domain such as the unit cube [22].

There are many excellent accounts of wavelets on $\mathbb{R}^d$ (see e.g. [33] or [25]). For the construction and description of wavelet bases on domains and manifolds, we refer the reader to the survey paper [19] and the references therein. This survey also sets forth the notation we shall employ below for indexing the elements in a wavelet basis. To understand this notation, it may be useful for the reader to keep in mind the case of wavelet bases on $\mathbb{R}^d$. In this setting, a typical biorthogonal wavelet basis of compactly supported functions is given by the shifted dilates of a set Γ of $2^d - 1$ functions. Namely, the collection of functions

$$2^{jd/2}\gamma(2^j \cdot -k), \quad j \in \mathbb{Z},\ k \in \mathbb{Z}^d,\ \gamma \in \Gamma, \tag{2.7}$$

form a Riesz basis for $L_2(\mathbb{R}^d)$. The dual basis is given by

$$2^{jd/2}\tilde{\gamma}(2^j \cdot -k), \quad j \in \mathbb{Z},\ k \in \mathbb{Z}^d,\ \tilde{\gamma} \in \tilde{\Gamma}, \tag{2.8}$$

with $\tilde{\Gamma}$ again a set of $2^d - 1$ functions. The integer j gives the dyadic level (2^j the frequency) of the wavelet. The multiinteger k gives the position ($2^{-j}k$) of the wavelet. Namely, the wavelet has support contained in a cube of diameter $\lesssim 2^{-j}$ centered at the point $2^{-j}k$. Note that there are $2^d - 1$ functions with the same dyadic level j and position $2^{-j}k$.

Another way to construct a wavelet basis for $\mathbb{R}^d$ is to start the multiscale decomposition at a finite dyadic level j_0. In this case, the basis consists of the functions of (2.7) with $j \geq j_0$, together with a family of functions

$$2^{j_0 d/2}\phi(2^{j_0} \cdot -k), \quad k \in \mathbb{Z}^d, \tag{2.9}$$

with ϕ a fixed (scaling) function. Wavelet bases for domains take a similar form except that some alterations are necessary near the boundary.

We shall denote wavelet bases by $\{\psi_\lambda\}_{\lambda \in \nabla}$. In the particular case above, this notation incorporates the three parameters j, k, γ into the one λ. We use $|\lambda| := j$ to denote the dyadic level of the wavelet. We let $\Psi_j = \{\psi_\lambda : \lambda \in \nabla_j\}$, $\nabla_j := \{\lambda \in \nabla : |\lambda| = j\}$, consist of the wavelets at level j.

In all classical constructions of compactly supported wavelets, there exists fixed constants C and M such that $\mathrm{diam}(\mathrm{supp}(\psi_\lambda)) \leq C2^{-|\lambda|}$ and such that for all $\lambda \in \nabla_j$ there are at most M indices $\mu \in \nabla_j$ such that $\mathrm{meas}(\mathrm{supp}(\psi_\lambda) \cap \mathrm{supp}(\psi_\mu)) \neq 0$.

Since we shall consider only bounded domains in this paper, the wavelet decomposition will begin at some fixed level j_0. For notational convenience only, we assume $j_0 = 1$. We define Ψ_0 to be the set of scaling functions in the wavelet basis. We shall assume that Ω is a domain or manifold which admits two sets of functions:

$$\Psi = \{\psi_\lambda : \lambda \in \nabla\} \subset L_2(\Omega), \quad \tilde{\Psi} = \{\tilde{\psi}_\lambda : \lambda \in \nabla\} \subset L_2(\Omega) \tag{2.10}$$

that form a *biorthogonal* wavelet bases on Ω: writing $\langle \Theta, \Phi \rangle := (\langle \theta, \phi \rangle_{L_2(\Omega)})_{\theta \in \Theta, \phi \in \Phi}$ for any two collections Θ, Φ of functions in $L_2(\Omega)$, one has

$$\langle \Psi, \tilde{\Psi} \rangle = \mathbf{I}, \tag{2.11}$$

where $\mathbf{I}$ is the identity matrix.

A typical feature in the theory of biorthogonal bases is that the sequences $\Psi, \tilde{\Psi}$ are Riesz-bases. That is, using the shorthand notation $\mathbf{d}^T\Psi := \sum_{\lambda\in\nabla} d_\lambda \psi_\lambda$, one has

$$\|\mathbf{d}\|_{\ell_2(\nabla)} \sim \|\mathbf{d}^T\Psi\|_{L_2(\Omega)} \sim \|\mathbf{d}^T\tilde{\Psi}\|_{L_2(\Omega)}. \tag{2.12}$$

This property means that the wavelet bases characterize $L_2(\Omega)$. In the present context of elliptic equations, we shall not need (2.12) but rather that these bases provide a characterization of H and H^* in terms of wavelet coefficients. This is expressed by the following specific assumption.

(A2): *Let the energy space H be equipped with the norm $\|\cdot\|_H$ and its dual space H^* be equipped with the norm $\|v\|_{H^*} := \sup_{\|w\|_H=1} |\langle v, w\rangle|$. We assume that the wavelets in Ψ are in H, whereas those in $\tilde{\Psi}$ are in H^* (in this context, we can assume that (2.11) simply holds in the sense of the duality (H, H^*)). We assume that each $v \in H$ has a wavelet expansion $v = \mathbf{d}^T\Psi$ (with coordinates $d_\lambda = \langle v, \tilde{\psi}_\lambda\rangle$) and that*

$$\|\mathbf{D}^{-1}\mathbf{d}\|_{\ell_2(\nabla)} \sim \|\mathbf{d}^T\Psi\|_H. \tag{2.13}$$

with $\mathbf{D}$ a fixed positive diagonal matrix.

Observe that (2.13) implies that $\mathbf{D}_{\lambda,\lambda} \sim \|\psi_\lambda\|_H^{-1}$, and that Ψ (resp. $D^{-1}\Psi$) is an unconditional (resp. Riesz) basis for H. By duality, one easily obtains that each $v \in H^*$ has a wavelet expansion $v = \mathbf{d}^T\tilde{\Psi}$ (with coordinates $d_\lambda = \langle v, \psi_\lambda\rangle$) that satisfies

$$\|\mathbf{D}\mathbf{d}\|_{\ell_2(\nabla)} \sim \|\mathbf{d}^T\tilde{\Psi}\|_{H^*}. \tag{2.14}$$

One should keep in mind though that $\tilde{\Psi}$ is only needed for analysis purposes. The Galerkin schemes to be considered below only involve Ψ while $\tilde{\Psi}$ never enters any computation and need not even be known explicitly.

It is well known (see e.g. [22]) that wavelet bases provide such characterizations for a large variety of spaces (in particular the Sobolev and Besov spaces for a certain parameter range which depends on the smoothness of the wavelets). In the context of elliptic equations, H is typically some Sobolev space H^t. In this case **(A2)** is satisfied whenever the wavelets are sufficiently smooth, with $\mathbf{D}_{\lambda,\lambda} = 2^{-|\lambda|t}$. For instance, when $A = -\Delta$, one has $t = 1$.

2.3 Discretization and preconditioning of the elliptic equation

Using wavelets, we can rewrite (2.5) as an infinite system of linear equations. We take wavelet bases Ψ and $\tilde{\Psi}$ satisfying **(A2)** and write the unknown solution $u = \mathbf{d}^T\Psi$ and the given right hand side f in terms of the basis $\tilde{\Psi}$. This gives the system of equations

$$\langle A\Psi, \Psi\rangle^T\mathbf{d} = \langle f, \Psi\rangle^T. \tag{2.15}$$

The solution $\mathbf{d}$ to (2.15) gives the wavelet coefficients of the solution u to (2.5).

An advantage of wavelet bases is that they allow for *trivial preconditiong of the linear system* (2.15). This preconditioning is given by the matrix $\mathbf{D}$ of **(A2)** and results in the system of equations:

$$\mathbf{D}\langle A\Psi, \Psi\rangle^T \mathbf{D}\mathbf{D}^{-1}\mathbf{d} = \mathbf{D}\langle f, \Psi\rangle^T, \tag{2.16}$$

or more compactly,

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \tag{2.17}$$

where

$$\mathbf{A} := \mathbf{D}\langle A\Psi, \Psi\rangle^T \mathbf{D}, \quad \mathbf{u} := \mathbf{D}^{-1}\mathbf{d}, \quad \mathbf{f} := \mathbf{D}\langle f, \Psi\rangle^T \in \ell_2(\nabla). \tag{2.18}$$

Let us briefly explain the effect of the above diagonal scaling with regard to preconditioning. To this end, note that by **(A1)**, the matrix $\mathbf{A}$ is symmetric positive definite. We define its associated bilinear form $\mathbf{a}$ by

$$\mathbf{a}(\mathbf{v}, \mathbf{w}) := \langle \mathbf{A}\mathbf{v}, \mathbf{w}\rangle_{\ell_2(\nabla)}, \tag{2.19}$$

where $\langle \cdot, \cdot \rangle_{\ell_2(\nabla)}$ is the standard inner product in $\ell_2(\nabla)$, and denote the norm associated with this bilinear form by $\|\cdot\|$. In other words,

$$\|\mathbf{v}\|^2 := \mathbf{a}(\mathbf{v}, \mathbf{v}), \quad \mathbf{v} \in \ell_2(\nabla). \tag{2.20}$$

Combining the ellipticity assumption **(A1)** together with the wavelet characterization of H **(A2)**, we obtain that $\|\cdot\|$ and $\|\cdot\|_{\ell_2(\nabla)}$ are equivalent norms, i.e., there exist constants $c_1, c_2 > 0$ such that

$$c_1\|\mathbf{v}\|^2_{\ell_2(\nabla)} \leq \|\mathbf{v}\|^2 \leq c_2\|\mathbf{v}\|^2_{\ell_2(\nabla)}. \tag{2.21}$$

It is immediate that these constants are also such that

$$c_1\|\mathbf{v}\|_{\ell_2(\nabla)} \leq \|\mathbf{A}\mathbf{v}\|_{\ell_2(\nabla)} \leq c_2\|\mathbf{v}\|_{\ell_2(\nabla)}, \tag{2.22}$$

and

$$c_2^{-1}\|\mathbf{v}\|_{\ell_2(\nabla)} \leq \|\mathbf{A}^{-1}\mathbf{v}\|_{\ell_2(\nabla)} \leq c_1^{-1}\|\mathbf{v}\|_{\ell_2(\nabla)}. \tag{2.23}$$

In particular, the condition number $\kappa := \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ of $\mathbf{A}$ satisfies

$$\kappa \leq c_2 c_1^{-1}. \tag{2.24}$$

The fact that the diagonal scaling turns the original operator into an isomorphism on $\ell_2(\nabla)$ will be a cornerstone of the subsequent development. Denoting by $a_{\lambda,\lambda'}$ the entries of $\mathbf{A}$ and by $\mathbf{A}_\Lambda = (a_{\lambda,\lambda'})_{\lambda,\lambda'\in\Lambda}$ the section of $\mathbf{A}$ restricted to the set Λ, it follows from the positive definiteness of $\mathbf{A}$ that

$$\|\mathbf{A}_\Lambda\| \leq \|\mathbf{A}\|, \quad \|\mathbf{A}_\Lambda^{-1}\| \leq \|\mathbf{A}^{-1}\|, \tag{2.25}$$

and that the condition numbers of the submatrices remain for *any* subset $\Lambda \subset \nabla$ uniformly bounded, i.e.,

$$\kappa(\mathbf{A}_\Lambda) := \|\mathbf{A}_\Lambda\|\|\mathbf{A}_\Lambda^{-1}\| \leq \kappa. \tag{2.26}$$

Finally, it is easy to check that the constants c_1 and c_2 also provide the equivalence

$$c_1^{1/2}\|\mathbf{v}\| \leq \|\mathbf{A}\mathbf{v}\|_{\ell_2(\nabla)} \leq c_2^{1/2}\|\mathbf{v}\|. \tag{2.27}$$

Here and later, we adopt the following rule about denoting constants. We shall denote constants which appear later in our analysis by $c_1, c_2, \cdots$. Other constants, whose value is not so important for us, will be denoted by C or incorporated into the $\lesssim$, $\sim$ notation.

A typical instance of the above setting involves Sobolev spaces $H = H^t$ in which case the entries of the diagonal matrix $\mathbf{D}$ can be chosen as $2^{-t|\lambda|}\delta_{\lambda,\lambda'}$. Of course, the constants in (2.24) will then depend on the relation between the energy norm (2.20) and the Sobolev norm. In some cases such a detour through a Sobolev space is not necessary and (2.13) can be arranged to hold for a suitable $\mathbf{D}$ when $\|\cdot\|_H$ already coincides with the energy norm. A simple example is $Au = -\epsilon\Delta u + u$ where $(\mathbf{D})_{\lambda,\lambda'} := \max\{1, \sqrt{\epsilon}2^{|\lambda|}\}\delta_{\lambda,\lambda'}$ is an appropriate choice. In fact, (2.13) will then hold *independently* of ϵ.

2.4 Quasi-sparsity assumptions on the stiffness matrix

Another advantage of the wavelet basis is that for a large class of elliptic operators, the resulting preconditioned matrix $\mathbf{A}$ exhibits fast decay away from the diagonal. This will later be crucial with regard to storage economy and efficiency of (approximate) matrix/vector multiplication.

Consider for example the (typical) case when H is the Sobolev space H^t of order t or its subspace H_0^t. Then, for a large class of elliptic operators, we have

$$2^{-(|\lambda'|+|\lambda|)t}|\langle A\psi_{\lambda'}, \psi_\lambda\rangle| \lesssim 2^{-||\lambda|-|\lambda'||\sigma}(1 + d(\lambda,\lambda'))^{-\beta}, \tag{2.28}$$

with $\sigma > d/2$ and $\beta > d$ and

$$d(\lambda,\lambda') := 2^{\min(|\lambda|,|\lambda'|)}\,\mathrm{dist}(\mathrm{supp}(\psi_\lambda), \mathrm{supp}(\psi_{\lambda'})). \tag{2.29}$$

The validity of (2.28) has been established in numerous settings (see e.g. [19, 8, 35, 37]). Decay estimates of the form (2.28) were initially introduced in [32] in the context of Littlewood-Paley analysis. The constant σ depends on the smoothness of the wavelets whereas β is related to the approximation order of the dual multiresolution (resp. the vanishing moments of the wavelets) and the order of the operator A. Estimates of the type (2.28) are known to hold for a wide range of cases including classical pseudo-differential operators and Calderón-Zygmund operators (see e.g. [21, 36]). In particular, the single and double layer potential operators fall into this category. We refer the reader to [19] for a full discussion of settings where (2.28) is valid.

We introduce the class $\mathcal{A}_{\sigma,\beta}$ of all matrices $\mathbf{B} = (b_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$ which satisfy

$$|b_{\lambda,\lambda'}| \leq c_{\mathbf{B}}2^{-||\lambda|-|\lambda'||\sigma}(1 + d(\lambda,\lambda'))^{-\beta}, \tag{2.30}$$

with $d(\lambda,\lambda')$ defined by (2.29). We say that a matrix $\mathbf{B}$ is *quasi-sparse* if it is in the class $\mathcal{A}_{\sigma,\beta}$ for some $\sigma > d/2$ and $\beta > d$. Properties of quasi-sparse matrices will be discussed in §3.

(A3): We assume that, for some $\sigma > d/2$, $\beta > d$, the matrix $\mathbf{A}$ of (2.17) is in the class $\mathcal{A}_{\sigma,\beta}$.

Let us note that in the case $H = H^t$ discussed earlier, we obtain (2.30) from (2.28) because $\mathbf{D} = (2^{-t|\lambda|}\delta_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$.

2.5 Wavelet Galerkin methods

A wavelet based Galerkin method for solving (2.5) takes the following form. We choose a finite set Λ of wavelet indices and use the space $S_\Lambda := \text{span}\{\psi_\lambda : \lambda \in \Lambda\}$ as our trial and analysis space. The approximate Galerkin solution u_Λ from S_Λ is defined by the conditions

$$a(u_\Lambda, v) = \langle f, v\rangle_{L_2(\Omega)}, \quad v \in S_\Lambda. \tag{2.31}$$

We introduce some notation which will help embed the finite dimensional problem (2.31) into the infinite dimensional space $\ell_2(\nabla)$. For any set $\Lambda \subset \nabla$, we let

$$\ell_2(\Lambda) := \{\mathbf{v} = (v_\lambda)_{\lambda\in\nabla} \in \ell_2(\nabla) : v_\lambda = 0,\ \lambda \notin \Lambda\}.$$

Thus, we will for convenience identify a vector with finitely many components with the sequence obtained by setting all components outside its support to zero. Moreover, let $\mathbf{P}_\Lambda$ denote the orthogonal projector from $\ell_2(\nabla)$ onto $\ell_2(\Lambda)$, that is, $\mathbf{P}_\Lambda\mathbf{v}$ is simply obtained from $\mathbf{v}$ by setting all coordinates outside Λ to zero.

Using the preconditioning matrix $\mathbf{D}$, (2.31) is equivalent to the finite linear system

$$\mathbf{P}_\Lambda\mathbf{A}\mathbf{u}_\Lambda = \mathbf{P}_\Lambda\mathbf{f}, \tag{2.32}$$

with unknown vector $\mathbf{u}_\Lambda \in \ell_2(\Lambda)$ and where $\mathbf{A}$ and $\mathbf{f}$ refer to the preconditioned system given in (2.18). The solution $\mathbf{u}_\Lambda$ to (2.32) determines the wavelet coefficients of u_Λ. Namely,

$$u_\Lambda = (\mathbf{D}\mathbf{u}_\Lambda)^T\Psi. \tag{2.33}$$

Of course, coefficients corresponding to $\lambda \notin \Lambda$ are zero.

We shall work almost exclusively in the remainder of this paper with the preconditioned discrete system (2.17). Note that the solution $\mathbf{u}_\Lambda$ to (2.32) can be viewed as its Galerkin approximation. In turn, it has the property that

$$\|\mathbf{u} - \mathbf{u}_\Lambda\| = \inf_{\mathbf{v}\in\ell_2(\Lambda)} \|\mathbf{u} - \mathbf{v}\|. \tag{2.34}$$

Our problem then is to find a good set of indices Λ such that the Galerkin solution $\mathbf{u}_\Lambda \in \ell_2(\Lambda)$ is a good approximation to $\mathbf{u}$. In view of the equivalences (see (2.21),(2.3), (2.20))

$$\|u - u_\Lambda\|_H \sim \|u - u_\Lambda\|_a \sim \|\mathbf{u} - \mathbf{u}_\Lambda\|_{\ell_2(\nabla)} \sim \|\mathbf{u} - \mathbf{u}_\Lambda\|, \tag{2.35}$$

any estimate for the error $\|\mathbf{u} - \mathbf{u}_\Lambda\|$ translates into an estimate for how well the Galerkin solution u_Λ from the wavelet space S_Λ approximates u.

3 N-term Approximation and Quasi-Sparse Matrices

We have seen in the previous section how the problem of finding Galerkin solutions to u from the wavelet space S_Λ is equivalent to finding Galerkin approximations to $\mathbf{u}$ from the sequence spaces $\ell_2(\Lambda)$. This leads us to understand first what properties of a vector $\mathbf{v} \in \ell_2(\nabla)$ determine its approximability from the spaces $\ell_2(\Lambda)$. It turns out that this is a simple and well understood problem in approximation theory which we now review.

3.1 N-term Approximation

In this subsection, we want to understand the properties of $\mathbf{u}$ that determine its approximability by a $\mathbf{u}_\Lambda$ with Λ of small cardinality. This is a special case of what is called N-term approximation which is completely understood in our setting. We shall recall the simple results in this subject that are pertinent to our analysis.

For each $N = 1, 2, \ldots$, let $\Sigma_N := \cup\{\ell_2(\Lambda) : \#\Lambda \leq N\}$. Thus, Σ_N is the (nonlinear) subspace of $\ell_2(\nabla)$ consisting of all vectors with at most N nonzero coordinates. Given $\mathbf{v} \in \ell_2(\nabla)$, $\mathbf{v} = (v_\lambda)_{\lambda\in\nabla}$, we introduce the error of approximation

$$\sigma_N(\mathbf{v}) := \inf_{\mathbf{w}\in\Sigma_N} \|\mathbf{v} - \mathbf{w}\|_{\ell_2(\nabla)}. \tag{3.1}$$

A best approximation to $\mathbf{v}$ from Σ_N is obtained by taking a set Λ with $\#\Lambda \leq N$ on which $|v_\lambda|$ takes its N largest values. The set Λ is not unique but all such sets yield best approximations from Σ_N. Indeed, given such a set Λ, we let $\mathbf{P}_\Lambda\mathbf{v}$ be the vector in Σ_N which agrees with $\mathbf{v}$ on Λ. Then

$$\sigma_N(\mathbf{v}) = \|\mathbf{v} - \mathbf{P}_\Lambda\mathbf{v}\|_{\ell_2(\nabla)}.$$

We next want to understand which vectors $\mathbf{v} \in \ell_2(\nabla)$ can be approximated efficiently by the elements of Σ_N. For each $s > 0$, we let $\mathcal{A}^s$ denote the set of all vectors $\mathbf{v} \in \ell_2(\nabla)$ such that

$$\|\mathbf{v}\|_{\mathcal{A}^s} := \sup_{N\geq 0}(N+1)^s\sigma_N(\mathbf{v}) \tag{3.2}$$

is finite, where $\sigma_0(\mathbf{v}) := \|v\|_{\ell_2(\nabla)}$. Thus $\mathcal{A}^s$ consists of all vectors which can be approximated with order $O(N^{-s})$ by the elements of Σ_N.

It is easy to characterize $\mathcal{A}^s$ for any $s > 0$. For this we introduce the *decreasing rearrangement* $\mathbf{v}^*$ of $\mathbf{v}$. For each $n \geq 1$, let v_n^* be the n-th largest of the numbers $|v_\lambda|$ and let $\mathbf{v}^* := (v_n^*)_{n=1}^\infty$. For each $0 < \tau < 2$, we let $\ell_\tau^w(\nabla)$ denote the collection of all vectors $\mathbf{v} \in \ell_2(\nabla)$ for which

$$|\mathbf{v}|_{\ell_\tau^w(\nabla)} := \sup_{n\geq 1} n^{1/\tau}\mathbf{v}_n^* \tag{3.3}$$

is finite. The space $\ell_\tau^w(\nabla)$ is called *weak* ℓ_τ and is a special case of a *Lorentz sequence space*. The expression (3.3) defines its quasi-norm (it does not in general satisfy the triangle inequality). We shall only consider the case $\tau < 2$ in this paper. In this case $\ell_\tau^w(\nabla) \subset \ell_2(\nabla)$ and for certain notational convenience, we define

$$\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} := |\mathbf{v}|_{\ell_\tau^w(\nabla)} + \|\mathbf{v}\|_{\ell_2(\nabla)}. \tag{3.4}$$

If $\mathbf{v}$,$\mathbf{w}$ are two sequences, then

$$\|\mathbf{v}+\mathbf{w}\|_{\ell_\tau^w(\nabla)} \le C(\tau)\left(\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} + \|\mathbf{w}\|_{\ell_\tau^w(\nabla)}\right), \tag{3.5}$$

with $C(\tau)$ depending on τ when τ tends to zero.

We have $\mathbf{v} \in \ell_\tau^w(\nabla)$ if and only if $v_n^* \le cn^{-1/\tau}$, $n \ge 1$, and the smallest such c is equal to $\|\mathbf{v}\|_{\ell_\tau^w}$. In other words, the coordinates of $\mathbf{v}$ when rearranged in decreasing order are required to decay at the rate $O(n^{-1/\tau})$. Another description of this space is given by

$$\#\{\lambda : |\mathbf{v}_\lambda| \ge \epsilon\} \le c\epsilon^{-\tau} \tag{3.6}$$

and the smallest c which satisfies (3.6) is equivalent to $|\mathbf{v}|^\tau_{\ell_\tau^w(\nabla)}$.

Remark 3.1 *An alternative description of $\ell_\tau^w(\nabla)$ is*

$$\{\mathbf{v} \in \ell_2(\nabla) : \#\{\lambda : 2^{-j} \ge |v_\lambda| \ge 2^{-j-1}\} \le c2^{j\tau},\ j \in \mathbb{Z} \text{ for some } c < \infty\}.$$

Moreover the smallest such c is equivalent to $|\mathbf{v}|^\tau_{\ell_\tau^w(\nabla)}$.

We recall that $\ell_\tau^w(\nabla)$ contains $\ell_\tau(\nabla)$, and we trivially have $n(v_n^*)^\tau \le \sum_{n\ge1} |v_n|^\tau$ and therefore

$$|\mathbf{v}|_{\ell_\tau^w(\nabla)} \le \|\mathbf{v}\|_{\ell_\tau(\nabla)}, \tag{3.7}$$

i.e.,

$$\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} \le 2\left(\sum_{\lambda\in\nabla} |v_\lambda|^\tau\right)^{1/\tau}. \tag{3.8}$$

The following well known result characterizes $\mathcal{A}^s$.

Proposition 3.2 *Given $s > 0$, let τ be defined by*

$$\frac{1}{\tau} = s + \frac{1}{2}. \tag{3.9}$$

Then the sequence $\mathbf{v}$ belongs to $\mathcal{A}^s$ if and only if $\mathbf{v} \in \ell_\tau^w(\nabla)$ and

$$\|\mathbf{v}\|_{\mathcal{A}^s} \sim \|\mathbf{v}\|_{\ell_\tau^w(\nabla)} \tag{3.10}$$

with constants of equivalency depending only on τ when τ tends to zero (respectively, only on s when s tends to ∞). In particular, if $\mathbf{v} \in \ell_\tau^w(\nabla)$, then

$$\sigma_n(\mathbf{v}) \le C\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} n^{-s}, \quad n = 1, 2, \ldots, \tag{3.11}$$

with the constant C depending only on τ when τ tends to zero.

For the simple proof of this proposition, we refer the reader to [29] or the survey [26].

Conditions like $\mathbf{u} \in \ell_\tau(\nabla)$ or $\mathbf{u} \in \ell_\tau^w(\nabla)$, are equivalent to *smoothness conditions* on the function u. We describe a typical situation when $H = H^t$ and $\mathbf{D} = (2^{-t|\lambda|}\delta_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$. Then, the condition $\mathbf{u} \in \ell_\tau(\nabla)$ is equivalent to the requirement that the wavelet coefficients $\langle u, \tilde{\psi}_\lambda\rangle$, $\lambda \in \nabla$, satisfy

$$(2^{t|\lambda|}\langle u, \tilde{\psi}_\lambda\rangle)_{\lambda\in\nabla} \in \ell_\tau(\nabla). \tag{3.12}$$

For a certain range of s (and hence τ) depending on the smoothness and vanishing moments of the wavelet basis, the condition (3.12) describes membership in a certain Besov space. Namely for s and τ related by (3.9), we have

$$\mathbf{u} \in \ell_\tau \text{ iff } u \in B_\tau^{sd+t}(L_\tau(\Omega)) \tag{3.13}$$

with $B_p^r(L_p)$ the usual Besov space measuring "r orders of smoothness in L_p". The weaker condition $\mathbf{u} \in \ell_\tau^w(\nabla)$ gives a slightly larger space X_τ endowed with the (quasi) norm

$$\|u\|_{X_\tau} := \|\mathbf{u}\|_{\ell_\tau^w(\nabla)}. \tag{3.14}$$

In view of (2.35), the space X^τ consists exactly of those functions u whose best N-term wavelet approximation in the energy norm produces an error $O(N^{-s})$.

3.2 Quasi-Sparse Matrices

In this subsection, we shall consider some of the properties of the quasi-sparse matrices $\mathbf{A}$ that appear in the discrete reformulation (2.17) of the elliptic equation (2.5). We recall that such matrices $\mathbf{A}$ are in the class $\mathbf{A}_{\sigma,\beta}$ for some $\sigma > d/2$, $\beta > d$ and therefore they satisfy (2.30)

We begin by discussing the mapping properties of matrices $\mathbf{B} \in \mathcal{A}_{\sigma,\beta}$. We denote by $\|\mathbf{B}\|$ the spectral norm of $\mathbf{B}$. We shall use the following version of the Schur Lemma: if for the matrix $\mathbf{B} = (b_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$ there is a sequence $\omega_\lambda > 0$, $\lambda \in \nabla$, and a positive constant c such that

$$\sum_{\lambda'\in\nabla} |b_{\lambda,\lambda'}|\omega_{\lambda'} \le c\omega_\lambda, \quad \text{and} \quad \sum_{\lambda\in\nabla} |b_{\lambda,\lambda'}|\omega_\lambda \le c\omega_{\lambda'}, \quad \lambda, \lambda' \in \nabla, \tag{3.15}$$

then $\|\mathbf{B}\| \le c$. An instance of the application of this lemma to the classes $\mathcal{A}_{\sigma,\beta}$ is the following result (which can be found in [32]).

Proposition 3.3 *If $\sigma > d/2$ and $\beta > d$ then every $\mathbf{B} \in \mathcal{A}_{\sigma,\beta}$ defines a bounded operator on $\ell_2(\nabla)$.*

Proof: We apply Schur's lemma with the weights $\omega_\lambda = 2^{-|\lambda|d/2}$, $\lambda \in \nabla$. To establish the first inequality in (3.15), let $\lambda \in \nabla$ and let $|\lambda| = j$. Then, using the estimate

$\sum_{|\lambda'|=j'}(1+d(\lambda,\lambda'))^{-\beta} \lesssim 2^{d\max\{0,j'-j\}}$ for the summation in space, we obtain

$$\begin{aligned}\omega_\lambda^{-1}\sum_{\lambda'\in\nabla}\omega_{\lambda'}|b_{\lambda,\lambda'}| &\lesssim 2^{d|\lambda|/2}\sum_{j'\geq 0}2^{-dj'/2}2^{-\sigma|j-j'|}\sum_{|\lambda'|=j'}(1+d(\lambda,\lambda'))^{-\beta}\\ &\lesssim \sum_{j'\geq j}2^{-d(j'-j)/2}2^{-\sigma(j'-j)}2^{d(j'-j)}+\sum_{0\leq j'<j}2^{-d(j'-j)/2}2^{\sigma(j'-j)}\\ &\lesssim \sum_{l\geq 0}2^{-(\sigma-d/2)l}<\infty.\end{aligned}$$

A symmetric argument confirms the second estimate in (3.15) proving that $\mathbf{B}$ is bounded. □

While Proposition 3.3 is of general interest, it does not tell us any additional information when applied to the matrix $\mathbf{A}$ of (2.17) since our ellipticity assumptions **(A1)** already implies that $\mathbf{A}$ is bounded on $\ell_2(\nabla)$.

It is well-known that decay estimates of the type (2.30) form the basis of matrix compression [8, 21, 35, 36]. The following proposition employs a compression technique which is somewhat different from the results in these papers.

Proposition 3.4 *For each $\sigma > d/2$, $\beta > d$ let*

$$s^* := \min\left\{\frac{\sigma}{d}-\frac{1}{2},\frac{\beta}{d}-1\right\} \tag{3.16}$$

assume that $\mathbf{B}\in\mathcal{A}_{\sigma,\beta}$. Then, given any $s<s^*$, there exists for every $J\in\mathbb{N}$ a matrix $\mathbf{B}_J$ which contains at most 2^J nonzero entries in each row and column and provides the approximation efficiency

$$\|\mathbf{B}-\mathbf{B}_J\|\leq C2^{-Js},\quad J\in\mathbb{N}. \tag{3.17}$$

Moreover this result also holds for $s=s^*$ provided $\sigma-d/2\neq\beta-d$.

Proof: Let $\mathbf{B}=(b_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$ be in $\mathcal{A}_{\sigma,\beta}$. We fix $J>0$ and we first apply a truncation in scale, defining $\tilde{\mathbf{B}}_J:=(\tilde{b}_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$ where

$$\tilde{b}_{\lambda,\lambda'}:=\begin{cases} b_{\lambda,\lambda'}, & ||\lambda|-|\lambda'||\leq J/d,\\ 0, & \text{else.}\end{cases}$$

In order to estimate the spectral norm $\|\mathbf{B}-\tilde{\mathbf{B}}_J\|$, we can employ the Schur lemma with the same weights as in the proof of Proposition 3.3. As in that proof, we obtain, for any $\lambda\in\nabla$ and $|\lambda|=j$,

$$\begin{aligned}\omega_\lambda^{-1}\sum_{\lambda'}\omega_{\lambda'}|b_{\lambda,\lambda'}-\tilde{b}_{\lambda,\lambda'}| &= \omega_\lambda^{-1}\sum_{\{\lambda':|j-|\lambda'||>J/d\}}\omega_{\lambda'}|b_{\lambda,\lambda'}|\\ &\lesssim \sum_{l>J/d}2^{-(\sigma-d/2)l}\\ &\lesssim 2^{-(\sigma-d/2)J/d}\lesssim 2^{-Js}.\end{aligned}$$

It follows that

$$\|\mathbf{B} - \tilde{\mathbf{B}}_J\| \lesssim 2^{-Js}. \tag{3.18}$$

We also need a truncation in space provided by the new matrix $\mathbf{B}_J := (b'_{\lambda,\lambda'})_{\lambda,\lambda'\in\nabla}$ where

$$b'_{\lambda,\lambda'} := \begin{cases} \tilde{b}_{\lambda,\lambda'}, & d(\lambda,\lambda') \le 2^{J/d-||\lambda|-|\lambda'||}\gamma(||\lambda| - |\lambda'||), \\ 0, & \text{else}, \end{cases}$$

and where $\gamma(n)$ is a polynomially decreasing sequence such that $\sum_n \gamma(n)^d < \infty$. Specifically, we take $\gamma(n) := (1+n)^{-2/d}$.

We can then immediately estimate the maximal number N_J of non-zero entries in each row and column of $\mathbf{B}_J$ by

$$N_J \lesssim \sum_{l=0}^{[J/d]} [2^{J/d-l}\gamma(l)]^d 2^{ld} \lesssim 2^J.$$

In view of (3.18), it remains only to prove that $\|\mathbf{B}_J - \tilde{\mathbf{B}}_J\| \lesssim 2^{-Js}$. In order to estimate the spectral norm $\|\mathbf{B}_J - \tilde{\mathbf{B}}_J\|$, we again use the Schur lemma with the same weights. For each j' and $\lambda \in \Lambda$, we have

$$\sum_{\{\lambda' \,:\, d(\lambda,\lambda')>R\}} (1 + d(\lambda,\lambda'))^{-\beta} \lesssim R^{-\beta+d} 2^{d\max\{0,|\lambda'|-|\lambda|\}},$$

Therefore, for any $\lambda \in \nabla$,

$$\begin{aligned} \omega_\lambda^{-1} \sum_{\lambda'} \omega_{\lambda'} |b'_{\lambda,\lambda'} - \tilde{b}_{\lambda,\lambda'}| &\lesssim \sum_{l=0}^{[J/d]} 2^{-(\sigma-d/2)l} [2^{J/d-l}\gamma(l)]^{-(\beta-d)} \\ &= 2^{-sJ} [2^{-J(\beta-d-ds)/d} \sum_{l=0}^{J} 2^{[(\beta-d)-(\sigma-d/2)]l} \gamma(l)^{-(\beta-d)}]. \end{aligned}$$

In the case where $(\beta - d) < (\sigma - d/2)$ (resp.$(\beta - d) > (\sigma - d/2)$), the factor on the right of 2^{-sJ} is bounded by $C2^{-J(\beta-d-ds)/d}$ (resp. $C2^{-J(\sigma-d/2-ds)/d}$) with C a constant independent of J and λ. Thus, when $\beta - d \neq \sigma - d/2$, we obtain the desired estimate of $\|\mathbf{B}_J - \tilde{\mathbf{B}}_J\|$ for all $s \le s^*$. On the other hand, when $\beta - d = \sigma - d/2$, this factor is still bounded by a fixed constant provided $s < s^*$. □

Remark 3.5 *In the case that the matrix* $\mathbf{B}$ *of Proposition 3.4 is the preconditioned matrix representation of an elliptic operator* A *which is local (i.e.,* $\operatorname{supp} A\psi_\lambda \subset \operatorname{supp} \psi_\lambda$, $\lambda \in \nabla$*) then the truncation in space in the proof of this proposition is not needed and the Proposition holds for* $ds \le \sigma - d/2$.

3.3 Fast Multiplication

We now come to the main result of this section which is the fast computation of quasi-sparse matrices applied to vectors. We continue to denote the spectral norm of a matrix $\mathbf{B}$ by $\|\mathbf{B}\|$.

We have seen that decay estimates like (2.28) imply compressibility in the sense of Proposition 3.4. To emphasize that only this compressibility (which may actually hold also for other operators than those discussed in connection with (2.28)) matters for the subsequent analysis we introduce the following class $\mathcal{B}_s$ of compressible matrices.

Definition 3.6 *We say a matrix $\mathbf{B}$ is in the class $\mathcal{B}_s$ if there are two positive sequences $(\alpha_j)_{j\geq 0}$ and $(\beta_j)_{j\geq 0}$ that are both summable and for every $j \geq 0$ there exists a matrix $\mathbf{B}_j$ with at most $2^j\alpha_j$ nonzero entries per row and column such that*

$$\|\mathbf{B} - \mathbf{B}_j\| \leq 2^{-js}\beta_j. \tag{3.19}$$

We further define

$$\|\mathbf{B}\|_{\mathcal{B}_s} := \min \max \left\{ \sum_{j\geq 0} \alpha_j, \sum_{j\geq 0} \beta_j \right\} \tag{3.20}$$

where the minimum is taken over all such sequences $(\alpha_j)_{j\geq 0}$ and $(\beta_j)_{j\geq 0}$.

We record the following consequence of Proposition 3.4.

Corollary 3.7 *Let s^* be defined by (3.16). Then for every $0 \leq s < s^*$ one has*

$$\mathcal{A}_{\sigma,\beta} \subset \mathcal{B}_s. \tag{3.21}$$

Note that the sequences (α_j), (β_j) can in this case be chosen to decay exponentially and that $\|\mathbf{B}\|_{\mathcal{B}_s}$ grows when s approaches s^*.

The main result of this section reads as follows.

Proposition 3.8 *If the matrix $\mathbf{B}$ is in the class $\mathcal{B}_s$, then $\mathbf{B}$ maps $\ell^\tau_w(\nabla)$ boundedly into itself for $1/\tau = 1/2 + s$, that is, for any $\mathbf{v} \in \ell^w_\tau(\nabla)$, we have*

$$\|\mathbf{B}\mathbf{v}\|_{\ell^w_\tau(\nabla)} \leq C\|\mathbf{v}\|_{\ell^w_\tau(\nabla)}. \tag{3.22}$$

with the constant C depending only on $\|\mathbf{B}\|_{\tilde{\mathcal{B}}_s}$ and the spectral norm $\|\mathbf{B}\|$.

Proof: Let $\mathbf{v} \in \ell^w_\tau(\nabla)$ and for any $j \geq 0$, we denote by $\mathbf{v}_{[j]} \in \Sigma_{2^j}$ be a best 2^j-term approximation to $\mathbf{v}$ in $\|\cdot\|_{\ell_2(\nabla)}$. We recall that $\mathbf{v}_{[j]}$ is obtained by retaining the 2^j biggest coefficients of $\mathbf{v}$ and setting all other coefficients to zero. Then, from Proposition 3.2, we have

$$\|\mathbf{v} - \mathbf{v}_{[j]}\|_{\ell_2(\nabla)} \leq C2^{-js}\|\mathbf{v}\|_{\ell^w_\tau(\nabla)} \tag{3.23}$$

with the constant depending only on τ. Using the matrices of (3.19), we define

$$\mathbf{w}_j := \mathbf{B}_j\mathbf{v}_{[0]} + \mathbf{B}_{j-1}(\mathbf{v}_{[1]} - \mathbf{v}_{[0]}) + \cdots + \mathbf{B}_0(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}). \tag{3.24}$$

This gives

$$\mathbf{B}\mathbf{v} - \mathbf{w}_j = \mathbf{B}(\mathbf{v} - \mathbf{v}_{[j]}) + (\mathbf{B} - \mathbf{B}_0)(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}) + \cdots + (\mathbf{B} - \mathbf{B}_j)\mathbf{v}_{[0]}.$$

It follows then from the summability of the β_j that

$$\begin{aligned}
\|\mathbf{B}\mathbf{v} - \mathbf{w}_j\|_{\ell_2(\nabla)} &\lesssim \|\mathbf{B}\|\|\mathbf{v} - \mathbf{v}_{[j]}\|_{\ell_2(\nabla)} \\
&+ \|\mathbf{B} - \mathbf{B}_0\|\|\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\|_{\ell_2(\nabla)} + \cdots + \|\mathbf{B} - \mathbf{B}_j\|\|\mathbf{v}_{[0]}\|_{\ell_2(\nabla)} \\
&\lesssim \|\mathbf{B}\|\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} 2^{-sj} + 2^{-s}\beta_0\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} 2^{-s(j-1)} + \cdots + 2^{-sj}\beta_j\|\mathbf{v}_{[0]}\|_{\ell_2(\nabla)} \\
&\lesssim 2^{-sj}\|\mathbf{v}\|_{\ell_\tau^w(\nabla)}, \qquad (3.25)
\end{aligned}$$

where for the last term, we have used the simple inequalities $\|\mathbf{v}_0\|_{\ell_2(\nabla)} \leq \|\mathbf{v}_0\|_{\ell_\tau^w(\nabla)} \leq \|\mathbf{v}\|_{\ell_\tau^w(\nabla)}$.

The number N_j of nonzero entries of $\mathbf{w}_j$ is estimated by

$$N_j \leq \alpha_j 2^j + 2\alpha_{j-1}2^{j-1} + \cdots + 2^j\alpha_0 \lesssim 2^j.$$

We apply now Proposition 3.2 and obtain (3.22). □

We state an immediate consequence of Corollary 3.7.

Corollary 3.9 *The conclusions of Proposition 3.8 hold for any matrix* $\mathbf{B} \in \mathcal{A}_{\sigma,\beta}$ *provided* $s < \min\{\sigma/d - 1/2, \beta/d - 1\} = s^*$.

Note that the number of arithmetic operations needed to compute $\mathbf{w}_j$ in (3.24) is estimated as N_j above, so that this multiplication algorithm is optimal. This is stated in the following corollary in which we also reformulate our result in terms of a prescribed tolerance.

Corollary 3.10 *Under the hypotheses of Proposition 3.8, for each* $\mathbf{v} \in \ell_\tau^w(\nabla)$, *and for each* $\epsilon > 0$, *there is a* $\mathbf{w}_\epsilon$ *such that*

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_\epsilon\|_{\ell_2(\nabla)} \leq \epsilon,$$

and

$$\#\operatorname{supp}\mathbf{w}_\epsilon \lesssim \epsilon^{-1/s}\|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^{1/s},$$

with s *and* τ *related as in* (3.9). *Moreover, the approximation* $\mathbf{w}_\epsilon$ *can be computed with* $C\|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^{1/s}\epsilon^{-1/s}$ *arithmetic operations. In both of these statements, the constants* C *depends only on* $\|\mathbf{B}\|_{\mathcal{B}_s}$ *and the spectral norm of* $\mathbf{B}$.

4 An Adaptive Galerkin Scheme

We have shown in §2, that the elliptic equation (2.5) is equivalent to the infinite system of equations (2.17)

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \tag{4.1}$$

where $\mathbf{A}$ is an isomorphism on $\ell_2(\nabla)$. This system results from expanding the solution and right hand side of (2.5) in a primal and dual wavelet basis, respectively, and then using a diagonal preconditioning. We have also noted in that section that, for a given set $\Lambda \subset \nabla$, solving (4.1) with trial space $\ell_2(\Lambda)$ is the same as solving (2.5) with the trial space S_Λ.

We are not only interested in rapidly solving the linear system (2.32) of equations for a given selection Λ of basis functions for the trial space S_Λ but also in adaptively generating possibly economic sets Λ needed to achieve a desired accuracy. Since adaptive approximation is a form of nonlinear approximation, it is reasonable to benchmark the performance of such an adaptive method against nonlinear N-term approximation as discussed in § 3. We recall that the results of §3.1 show that a vector $\mathbf{v}$ can be approximated with order $O(N^{-s})$ by N-term approximation (i.e., by a vector with at most N nonzero coordinates) if and only if $\mathbf{v} \in \ell_\tau^w(\nabla)$, $\tau := (s+1/2)^{-1}$. We shall strive therefore to meet the following goal.

Goal: *Construct an adaptive algorithm so that the following property holds for a wide range of $s > 0$: for each $\mathbf{u} \in \ell_\tau^w(\nabla)$, $\tau := (s+1/2)^{-1}$, the algorithm generates sets Λ_j, $j = 1, 2, \ldots$, such that the Galerkin approximation $\mathbf{u}_{\Lambda_j}$ to $\mathbf{u}$ provides the approximation error*

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\| \le C\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}(\#\Lambda_j)^{-s}. \tag{4.2}$$

Recall that this goal can also be expressed in terms of achieving a certain tolerance with an optimal number of degrees of freedom as stated in (1.2) and (1.3).

In this section, we shall describe a *first* adaptive algorithm, initially developed in [17], for solving (4.1). Starting with an initial set Λ_0, this algorithm adaptively generates a sequence of (nested) sets Λ_j, $j = 1, 2, \ldots$. The Galerkin solutions $\mathbf{u}_{\Lambda_j}$, $j = 1, 2 \ldots$, to (4.1) provide our numerical approximation to $\mathbf{u}$ and these in turn determine our approximations u_{Λ_j} to the solution u of the original elliptic equation (2.5).

At present, we can only show that the algorithm of this section meets our goal for a small range of $s > 0$ (see Corollary 4.10). Nevertheless, this algorithm is simple and interesting in several respects and the analysis of this algorithm brings forward natural questions concerning Galerkin approximations.

In § 5 we shall present a second adaptive algorithm which will meet our goal for a natural range $s^* > s > 0$. This range of $s > 0$ is limited only by the decay properties of the stiffness matrix $\mathbf{A}$ which in turn are related to properties of the wavelet basis (smoothness and vanishing moments) and the order of $\mathbf{A}$.

The analysis we give in this and the following section for these adaptive algorithms is *idealized* since it will address only questions of approximation order in terms of the cardinality of the sets Λ_j. At this stage we shall ignore certain computational issues. In

particular, we will assume that we are able to access the values of possibly infinitely many wavelet coefficients, e.g. of residuals, which is of course unrealistic. However, this will facilitate a more transparent analysis of the adaptive algorithms and their ingredients. Later in § 6-7 we will develop corresponding computable counterparts by introducing suitable truncation and approximation procedures. Moreover, we will provide a complete analysis of their computational complexity.

4.1 Algorithm I

The idea behind our first adaptive algorithm is to generate step by step an ascending sequence of (nested) sets Λ_j so that on the one hand $\#(\Lambda_j \setminus \Lambda_{j-1})$ stays as small as possible, while on the other hand, the error for the corresponding Galerkin solutions is reduced by some *fixed* factor, that is, for some $\theta \in (0,1)$ one has

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_{j+1}}\| \leq \theta \|\mathbf{u} - \mathbf{u}_{\Lambda_j}\|. \tag{4.3}$$

We remind the reader that $\|\cdot\| := \mathbf{a}(\cdot,\cdot)^{1/2}$ is the discrete energy norm when applied to vectors. The Λ_j will be generated adaptively, that is Λ_j depends on the given data $\mathbf{f}$ and on the previous solution $\mathbf{u}_{\Lambda_{j-1}}$.

We will first explain the basic principle that has been already used in [10, 17] to guarantee a reduction of the form (4.3). The idea is, given Λ, find $\tilde{\Lambda}$ containing Λ such that

$$\|\mathbf{u}_{\tilde{\Lambda}} - \mathbf{u}_\Lambda\| \geq \beta \|\mathbf{u} - \mathbf{u}_\Lambda\| \tag{4.4}$$

holds for some $\beta \in (0,1)$. By the orthogonality of the Galerkin solutions with respect to the energy inner product, (4.4) implies

$$\|\mathbf{u} - \mathbf{u}_\Lambda\|^2 = \|\mathbf{u} - \mathbf{u}_{\tilde{\Lambda}}\|^2 + \|\mathbf{u}_\Lambda - \mathbf{u}_{\tilde{\Lambda}}\|^2. \tag{4.5}$$

Hence (4.4) (applied with $\Lambda = \Lambda_j$ and $\tilde{\Lambda} = \Lambda_{j+1}$) implies (4.3) with

$$\theta := \sqrt{1-\beta^2}. \tag{4.6}$$

Therefore, our strategy is to establish (4.4). This is also a common approach in the context of finite element discretizations, see e.g. [10]. There the role of $u_{\tilde{\Lambda}}$ is played by an approximate solution of higher order or with respect to a finer mesh. In most studies, however, the property (4.4), often referred to as *saturation property*, is *assumed* and not *proven* to be valid.

We shall show how such sets $\tilde{\Lambda}$ can be selected. For this we shall use the residual

$$\mathbf{r}_\Lambda := \mathbf{A}\mathbf{u} - \mathbf{A}\mathbf{u}_\Lambda = \mathbf{f} - \mathbf{A}\mathbf{u}_\Lambda. \tag{4.7}$$

Since $\mathbf{u}_\Lambda$ and $\mathbf{f}$ are known to us, the coordinates of this residual can in principle be computed to any desired accuracy. We leave aside the issue of the computational cost for a given accuracy in this residual until § 6, and work with the simplified assumption that we have the exact knowledge of its coordinates.

We recall the orthogonal projector $\mathbf{P}_\Lambda$ from $\ell_2(\nabla)$ to $\ell_2(\Lambda)$ in the norm $\|\cdot\|_{\ell_2(\nabla)}$. For $\mathbf{v} \in \ell_2(\nabla)$, $\mathbf{P}_\Lambda \mathbf{v}$ is the vector in $\ell_2(\Lambda)$ which agrees with $\mathbf{v}$ on Λ.

Lemma 4.1 *Let $\Lambda \subset \nabla$ and let $\mathbf{r}_\Lambda := \mathbf{f} - \mathbf{A}\mathbf{u}_\Lambda$ be the residual associated to Λ. If $0 < \alpha < 1$, and $\tilde{\Lambda} \subset \nabla$ is any set that satisfies*

$$\|\mathbf{P}_{\tilde{\Lambda}}\mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \geq \alpha \|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)}, \tag{4.8}$$

then

$$\|\mathbf{u}_{\tilde{\Lambda}} - \mathbf{u}_\Lambda\| \geq \beta \|\mathbf{u} - \mathbf{u}_\Lambda\| \tag{4.9}$$

where $\beta := c_2^{-1/2} c_1^{1/2} \alpha$ and c_1, c_2 are the constants of (2.21)*. As a consequence,*

$$\|\mathbf{u} - \mathbf{u}_{\tilde{\Lambda}}\| \leq \theta \|\mathbf{u} - \mathbf{u}_\Lambda\| \tag{4.10}$$

with $\theta := \sqrt{1 - \beta^2}$.

Proof: From (2.27), we have

$$\begin{aligned}
\|\mathbf{u}_{\tilde{\Lambda}} - \mathbf{u}_\Lambda\| &\geq c_2^{-1/2} \|\mathbf{A}(\mathbf{u}_{\tilde{\Lambda}} - \mathbf{u}_\Lambda)\|_{\ell_2(\nabla)} \geq c_2^{-1/2} \|\mathbf{A}(\mathbf{u}_{\tilde{\Lambda}} - \mathbf{u}_\Lambda)\|_{\ell_2(\tilde{\Lambda})} \\
&= c_2^{-1/2} \|\mathbf{A}(\mathbf{u} - \mathbf{u}_\Lambda)\|_{\ell_2(\tilde{\Lambda})} = c_2^{-1/2} \|\mathbf{P}_{\tilde{\Lambda}}\mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \\
&\geq c_2^{-1/2} \alpha \|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)} = c_2^{-1/2} \alpha \|\mathbf{A}(\mathbf{u} - \mathbf{u}_\Lambda)\|_{\ell_2(\nabla)} \\
&\geq c_2^{-1/2} c_1^{1/2} \alpha \|\mathbf{u} - \mathbf{u}_\Lambda\|,
\end{aligned}$$

where the second to last equality uses the fact that $\mathbf{A}\mathbf{u} = \mathbf{f}$, and $\mathbf{A}\mathbf{u}_{\tilde{\Lambda}}$ agree on $\tilde{\Lambda}$. This proves (4.9) while (4.10) follows from (4.5). □

We consider now our first algorithm for choosing the sets Λ_j in which we take $\alpha = 1/2$ (similar algorithms and analysis hold for any $0 < \alpha < 1$). We introduce the following steps which will be part of our adaptive algorithms.

GALERKIN: *Given a set Λ,* **GALERKIN** *determines the Galerkin approximation $\mathbf{u}_\Lambda$ to $\mathbf{u}$ by solving the finite system of equations* (2.32).

GROW: *Given a set Λ and the Galerkin solution $\mathbf{u}_\Lambda$,* **GROW** *produces the smallest set $\tilde{\Lambda}$ which contains Λ and satisfies*

$$\|\mathbf{P}_{\tilde{\Lambda}}\mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \geq \frac{1}{2} \|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)}. \tag{4.11}$$

We note that the set $\tilde{\Lambda}$ is obtained by taking the indices of the largest coefficients of $\mathbf{r}_\Lambda$; the number of these indices to be chosen is determined by the criterion (4.11).

Algorithm I:

- Let $\Lambda_0 = \emptyset$ and $\mathbf{r}_{\Lambda_0} = \mathbf{f}$.
- For $j = 0, 1, 2, \ldots$, determine Λ_{j+1} from Λ_j by first applying **GALERKIN** (in order to find $\mathbf{u}_{\Lambda_j}$) and then applying **GROW**.

As a consequence of Lemma 4.1, we have the following.

Corollary 4.2 *For the sets Λ_j given by Algorithm I, the corresponding Galerkin approximations $\mathbf{u}_{\Lambda_j}$ of $\mathbf{u}$ satisfy*

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_{j+1}}\| \leq \theta \|\mathbf{u} - \mathbf{u}_{\Lambda_j}\|, \quad j = 1, 2, \ldots. \tag{4.12}$$

where

$$\theta := \sqrt{1 - \frac{c_1}{4c_2}}. \tag{4.13}$$

Consequently,

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\| \leq \theta^j \|\mathbf{u}\|, \quad j = 1, 2, \ldots. \tag{4.14}$$

Proof: The inequality (4.12) follows from (4.10) while (4.14) follows by repeatedly applying (4.12). □

4.2 Error Analysis for Algorithm I

While the last Corollary shows that for each $\mathbf{u} \in \ell_2(\nabla)$, the sequence $\{\mathbf{u}_{\Lambda_j}\}$ converges in the energy norm to $\mathbf{u}$, we would like to go further and understand how the error decreases with $\#\Lambda_j$. In particular, we would like to see if this algorithm meets our goal for certain $s > 0$. We begin with the following lemma.

Lemma 4.3 *Let $s > 0$, let $\mathbf{A}$ be in the class $\mathcal{B}_s$, and let $\mathbf{u} \in \ell_\tau^w(\nabla)$, $\tau := (s + 1/2)^{-1}$. Given any set $\Lambda \subset \nabla$, let $\tilde{\Lambda} \subset \nabla$ be the smallest set such that $\Lambda \subset \tilde{\Lambda}$ and*

$$\|\mathbf{P}_{\tilde{\Lambda}} \mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \geq \frac{1}{2} \|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)}. \tag{4.15}$$

Then one has

$$\#(\tilde{\Lambda} \setminus \Lambda) \leq c_3 \left(\frac{\|r_\Lambda\|_{\ell_\tau^w(\nabla)}}{\|r_\Lambda\|_{\ell_2(\nabla)}} \right)^{1/s}, \tag{4.16}$$

where c_3 is a constant depending only on s when s is large.

Proof: We will make frequent use of the following simple fact.

Remark 4.4 *Since Λ is finite, $\mathbf{u}_\Lambda$ is in $\ell_\tau^w(\nabla)$. By assumption $\mathbf{u} \in \ell_\tau^w(\nabla)$ and hence $\mathbf{u} - \mathbf{u}_\Lambda$ is also in $\ell_\tau^w(\nabla)$. Applying Proposition 3.8 we see that $\mathbf{r}_\Lambda$ is also in $\ell_\tau^w(\nabla)$.*

Now, for any $N \geq 1$, let Λ_N denote the indices of the N largest coefficients of $\mathbf{r}_\Lambda$ in absolute value. According to Proposition 3.2,

$$\|\mathbf{r}_\Lambda - \mathbf{P}_{\Lambda_n}\mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \leq C_0\|\mathbf{r}_\Lambda\|_{\ell_\tau^w(\nabla)}N^{-s}, \tag{4.17}$$

where C_0 depends only on s when s is large. We may assume that $C_0 \geq 1$. We choose N as the smallest integer such that

$$2C_0\|\mathbf{r}_\Lambda\|_{\ell_\tau^w(\nabla)}N^{-s} \leq \|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)},$$

and define $\tilde{\Lambda} := \Lambda \cup \Lambda_N$. Then, clearly (4.15) is satisfied. Moreover,

$$N \leq \left(\frac{2C_0\|\mathbf{r}_\Lambda\|_{\ell_\tau^w(\nabla)}}{\|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)}}\right)^{1/s} + 1 \leq 2\left(\frac{2C_0\|\mathbf{r}_\Lambda\|_{\ell_\tau^w(\nabla)}}{\|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)}}\right)^{1/s}$$

and so (4.16) is also satisfied. □

Lemma 4.3 gives our first hint of the importance of controlling the $\ell_\tau^w(\nabla)$ norms of the residuals $\mathbf{r}_{\Lambda_j}$. The following Theorem and Corollary will draw this out more and will provide our first error estimate for **Algorithm I**.

Theorem 4.5 *Let $s > 0$, let $\mathbf{A}$ be in the class $\mathcal{B}_s$, and let $\mathbf{u} \in \ell_\tau^w(\nabla)$, $\tau := (s + 1/2)^{-1}$. Define*

$$\theta := \sqrt{1 - \frac{c_1}{4c_2}}$$

with c_1, c_2 the constants of §2.4. Then, the Galerkin approximations $\mathbf{u}_{\Lambda_k}$, $k = 0, 1, \ldots$, generated by **Algorithm I** *satisfy*

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_k}\| = C_k^s(\#\Lambda_k)^{-s}, \tag{4.18}$$

where

$$C_1 \leq c_3c_1^{-1/2s}\theta^{1/s}\|\mathbf{f}\|_{\ell_\tau^w(\nabla)}^{1/s} \tag{4.19}$$

and the constants C_k, $k > 1$, satisfy

$$C_{k+1} \leq \theta^{1/s}(C_k + c_3c_1^{-1/2s}\|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}^{1/s}) \tag{4.20}$$

with c_3 the constant of Lemma 4.3.

Proof: We use the abbreviations $e_k := \|\mathbf{u} - \mathbf{u}_{\Lambda_k}\|$, $N_k := \#\Lambda_k$, $\rho_k := \|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}$, $k = 0, 1, \ldots$. The constants C_k, $k = 1, 2, \ldots$, are defined by (4.18). For any $k \geq 0$, we know

$$e_{k+1} \leq \theta e_k$$

and from Lemma 4.3 and (2.27), we obtain

$$N_{k+1} \leq N_k + c_3\rho_k^{1/s}\|\mathbf{A}(\mathbf{u} - \mathbf{u}_{\Lambda_k})\|_{\ell_2(\nabla)}^{-1/s} \leq N_k + c_3c_1^{-1/2s}\rho_k^{1/s}e_k^{-1/s}.$$

This means that for $k \geq 1$,

$$C_{k+1} := N_{k+1} e_{k+1}^{1/s} \leq (N_k + c_3 c_1^{-1/2s} \rho_k^{1/s} e_k^{-1/s}) \theta^{1/s} e_k^{1/s} \leq \theta^{1/s} (C_k + c_3 c_1^{-1/2s} \rho_k^{1/s}).$$

This proves (4.20). The same argument gives (4.19) because $\rho_0 = \|\mathbf{f}\|_{\ell_\tau^w(\nabla)}$ and $N_0 = 0$. □

Theorem 4.5 reveals that the growth of the constants C_k can be controlled by the size of the residual norms $\|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}$. The following Corollary shows that if these norms are bounded then so are the constants C_k.

Corollary 4.6 *If the hypotheses of Theorem 4.5 are valid and in addition*

$$\|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)} \leq M_0, \quad k = 0, 1, \ldots, \tag{4.21}$$

then

$$C_k \leq C(\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}^{1/s} + M_0^{1/s}), \quad k = 1, 2 \ldots, \tag{4.22}$$

with C a constant such that C^s depends only on s when $s \to \infty$. Consequently,

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_k}\| \leq C^s (M_0^{1/s} + \|\mathbf{u}\|_{\ell_\tau^w(\nabla)}^{1/s})^s (\#\Lambda_k)^{-1/s}. \tag{4.23}$$

Proof: We use the same notation as in the proof of Theorem 4.5. We define $M := c_3 c_1^{-1/2s} M_0^{1/s}$ and find

$$\begin{aligned} C_k &\leq \theta^{1/s} C_{k-1} + \theta^{1/s} M \leq \theta^{2/s} C_{k-2} + \theta^{2/s} M + \theta^{1/s} M \\ &\leq C_1 \theta^{(k-1)/s} + M \sum_{j=1}^{k-1} \theta^{j/s}. \end{aligned}$$

Now, $\theta < 1$, and from (4.19) and Proposition 3.8

$$C_1^s \lesssim \|\mathbf{f}\|_{\ell_\tau^w(\nabla)} \lesssim \|\mathbf{u}\|_{\ell_\tau^w(\nabla)}.$$

This proves (4.22). The estimate (4.23) then follows from (4.18). □

Remark 4.7 *Corollary 4.6 shows that if $\|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}$ is bounded independently of k, then we are successful in the goal that we have fixed in the beginning of this section. One can also check that optimality is achieved in the sense of a target accuracy $\epsilon > 0$: Let $j(\epsilon)$ be the smallest j such that $\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\| \leq \epsilon$. Then, since $\|\mathbf{u} - \mathbf{u}_{\Lambda_{j(\epsilon)-1}}\| > \epsilon$, we obtain the estimate $\#\Lambda_{j(\epsilon)-1} \lesssim \epsilon^{-1/s}$ from (4.23). From (4.16), we also derive that $\#(\Lambda_{j(\epsilon)} \setminus \Lambda_{j(\epsilon)-1}) \lesssim \epsilon^{-1/s}$. It follows that we have the desired estimate $\#\Lambda_{j(\epsilon)} \lesssim \epsilon^{-1/s}$.*

4.3 Bounding $\|\mathbf{r}_{\Lambda_k}\|_{\ell^w_\tau(\nabla)}$

Corollary 4.6 shows that if for each $\mathbf{u} \in \ell^w_\tau(\nabla)$, $\tau := (s+1/2)^{-1}$, the boundedness condition (4.21) holds with $M_0 \leq C\|\mathbf{u}\|_{\ell^w_\tau(\nabla)}$, then the algorithm meets our goal for $s = \frac{1}{\tau} - \frac{1}{2}$. We can give sufficient conditions for the validity of (4.21) in terms of the (finite) sections

$$\mathbf{A}_\Lambda := (a_{\lambda,\nu})_{\lambda,\nu\in\Lambda} \tag{4.24}$$

of the matrix $\mathbf{A}$. Note that in terms of these sections the Galerkin equations (2.32) take the form

$$\mathbf{A}_\Lambda \mathbf{u}_\Lambda = \mathbf{P}_\Lambda \mathbf{f}, \tag{4.25}$$

where according to our convention we always employ the same notation for the finitely supported vector $\mathbf{u}_\Lambda$ and the infinite sequence obtained by setting all components outside Λ to zero. Likewise, depending on the context, it will be convenient to treat $\mathbf{P}_\Lambda \mathbf{v}$ for $\mathbf{v} \in \ell_2(\nabla)$ either as an infinite sequence with zero entries outside Λ or as a finitely supported vector defined on Λ.

Recall also from (2.26) that the ellipticity of $\mathbf{A}$ implies the boundedness of $\mathbf{A}_\Lambda$ and its inverse in the spectral norm, uniformly in Λ. Also, from Proposition 3.8, it follows that $\mathbf{A}$ is a bounded operator on $\ell^w_\tau(\nabla)$. Therefore, the matrices $\mathbf{A}_\Lambda$ are uniformly bounded (independently of Λ) on $\ell^w_\tau(\Lambda)$ (where $\ell^w_\tau(\Lambda)$ is defined in analogy to $\ell_2(\Lambda)$).

Remark 4.8 *Under the assumptions of Lemma 4.3, if the inverse matrices $\mathbf{A}_\Lambda^{-1}$ are uniformly bounded on $\ell^w_\tau(\Lambda)$, i.e.,*

$$\sup_{\|\mathbf{v}\|_{\ell^w_\tau(\Lambda)}\leq 1} \|\mathbf{A}_\Lambda^{-1}\mathbf{v}\|_{\ell^w_\tau(\Lambda)} \leq M_1, \quad \Lambda \subset \nabla, \tag{4.26}$$

with $M_1 \geq 1$, then

$$\|\mathbf{r}_\Lambda\|_{\ell^w_\tau(\nabla)} \leq CM_1\|\mathbf{u}\|_{\ell^w_\tau(\nabla)}, \qquad \Lambda \subset \nabla. \tag{4.27}$$

with the constant C independent of Λ.

Proof: By assumption $\mathbf{u} \in \ell^w_\tau(\nabla)$. From Proposition 3.8, we find that $\mathbf{f}$ is also in $\ell^w_\tau(\nabla)$ and for all Λ,

$$\|\mathbf{P}_\Lambda \mathbf{f}\|_{\ell^w_\tau(\Lambda)} \leq \|\mathbf{f}\|_{\ell^w_\tau(\nabla)} \leq C_1\|\mathbf{u}\|_{\ell^w_\tau(\nabla)},$$

where C_1 is the norm of $\mathbf{A}$ on $\ell^w_\tau(\nabla)$. By our assumptions on $\mathbf{A}_\Lambda^{-1}$, we derive that

$$\|\mathbf{u}_\Lambda\|_{\ell^w_\tau(\nabla)} = \|\mathbf{u}_\Lambda\|_{\ell^w_\tau(\Lambda)} \leq M_1\|\mathbf{P}_\Lambda \mathbf{f}\|_{\ell^w_\tau(\nabla)} \leq C_1M_1\|\mathbf{u}\|_{\ell^w_\tau(\nabla)}.$$

This gives

$$\begin{aligned} \|\mathbf{r}_\Lambda\|_{\ell^w_\tau(\nabla)} &\leq C_1\|\mathbf{u} - \mathbf{u}_\Lambda\|_{\ell^w_\tau(\nabla)} \\ &\leq C_2\left(\|\mathbf{u}\|_{\ell^w_\tau(\nabla)} + \|\mathbf{u}_\Lambda\|_{\ell^w_\tau(\nabla)}\right) \leq C_2(1 + C_1M_1)\|\mathbf{u}\|_{\ell^w_\tau(\nabla)}, \end{aligned} \tag{4.28}$$

which implies (4.27). □

There is a soft functional analysis argument which shows that the boundedness condition (4.26) is satisfied for a certain range of τ close to 2.

Theorem 4.9 *Let $\mathbf{A} \in \mathcal{B}_{s_0}$ for some $s_0 > 0$. Then there is a $0 < \tilde{\tau} < 2$ and a constant $C > 0$ such that for all $\Lambda \subset \nabla$ and all $\tilde{\tau} \le \tau \le 2$,*

$$\|\mathbf{A}_\Lambda^{-1}\|_{\ell_\tau^w(\nabla)\to\ell_\tau^w(\nabla)} \le C. \tag{4.29}$$

Proof: First recall from (2.26) that the condition numbers κ_Λ of the matrices $\mathbf{A}_\Lambda$ statisfy $\kappa_\Lambda \le \kappa$ for any $\Lambda \subset \nabla$. Let $\mathbf{B}_\Lambda := \mu_\Lambda \mathbf{A}_\Lambda$ where $\mu_\Lambda^{-1} = \frac{\|\mathbf{A}_\Lambda\| + \|\mathbf{A}_\Lambda^{-1}\|^{-1}}{2}$. Then, $\mathbf{B}_\Lambda = \mathbf{I} - \mathbf{R}_\Lambda$ where $\|\mathbf{R}_\Lambda\| < \frac{\kappa-1}{\kappa+1}$.

Now let $\tau_0 := (s_0 + 1/2)^{-1}$. Then, both $\mathbf{I}$ and $\mathbf{A}$ are bounded on $\ell_{\tau_0}^w(\nabla)$. Hence, we have $\|\mathbf{R}_\Lambda\|_{\ell_{\tau_0}^w(\Lambda)\to\ell_{\tau_0}^w(\Lambda)} \le C_0$ for some positive constant C_0 independent of Λ. Using the Riesz-Thorin interpolation theorem for $\ell_2(\Lambda)$ and $\ell_\tau^w(\Lambda)$, we can find some $\tilde{\tau} < 2$ such that $\|\mathbf{R}_\Lambda\|_{\ell_\tau} \le r_0 < 1$, uniformly in Λ and $\tilde{\tau} \le \tau \le 2$. By the standard Neumann series argument, we obtain (4.29). □

Corollary 4.10 *If $\mathbf{A} \in \mathcal{B}_{s_0}$ for some $s_0 > 0$, then there is an $\tilde{s} > 0$ such that Algorithm I meets our goal for all $0 < s \le \tilde{s}$. That is, for each $\mathbf{u} \in \ell_\tau^w(\nabla)$, with $\frac{1}{\tau} - \frac{1}{2} =: s \le \tilde{s}$, Algorithm I generates a sequence of sets Λ_j, $j = 1, 2, \ldots$, such that*

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\| \le C\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}(\#\Lambda_j)^{-s}, \quad j = 1, 2, \ldots \tag{4.30}$$

with C a constant.

Proof: From Theorem 4.9, there is a $\tilde{\tau} < 2$ such that (4.29) holds uniformly for all $\tilde{\tau} \le \tau \le 2$ and $\Lambda \subset \nabla$. Remark 4.8 then shows the validity of (4.27). We now apply Corollary 4.6 and obtain (4.30) from (4.23). □

We close this section by making some observations about the growth of $\|\mathbf{r}_\Lambda\|_{\ell_\tau^w(\nabla)}$ and $\|\mathbf{u}_\Lambda\|_{\ell_\tau^w(\nabla)}$ for an *arbitrary* range of s which is only limited by the properties of the wavelet bases. We shall use these observations in the following section when we modify **Algorithm I**.

Lemma 4.11 *Suppose that $\mathbf{u} \in \ell_\tau^w(\nabla)$ and $\tau = (s + 1/2)^{-1}$ with $s > 0$. Then, for any $\Lambda \subset \nabla$ one has*

$$\|\mathbf{u}_\Lambda\|_{\ell_\tau^w(\nabla)} \le c_4 \left(\|\mathbf{u}\|_{\ell_\tau^w(\nabla)} + (\#\Lambda)^s \|\mathbf{u} - \mathbf{u}_\Lambda\|_{\ell_2(\nabla)} \right) \tag{4.31}$$

with the constant c_4 depending only on τ when τ tends to 0.

Proof: First note that if $\mathbf{v} \in \ell_2(\Lambda)$, then $\mathbf{v}$ has at most $\#\Lambda$ nonzero coordinates. Using (3.8) and Hölder's inequality gives for such $\mathbf{v}$ the inverse estimate

$$|\mathbf{v}|_{\ell_\tau^w(\nabla)} \le \|\mathbf{v}\|_{\ell_\tau(\Lambda)} \le \left(\sum_{\lambda\in\Lambda} |v_\lambda|^2 \right)^{1/2} (\#\Lambda)^{\frac{1}{\tau}-\frac{1}{2}} \le (\#\Lambda)^s \|\mathbf{v}\|_{\ell_2(\Lambda)}. \tag{4.32}$$

Now let $\mathbf{u}_N$ denote the best N-term approximation to $\mathbf{u}$ which we recall is obtained by retaining the N largest coefficients. Invoking the direct estimate from Remark 3.2, we use (4.32) to conclude that

$$\begin{aligned} |\mathbf{u}_\Lambda|_{\ell_\tau^w(\nabla)} &\leq C\left(|\mathbf{u}_\Lambda - \mathbf{u}_{\#\Lambda}|_{\ell_\tau^w(\nabla)} + |\mathbf{u}_{\#\Lambda}|_{\ell_\tau^w(\nabla)}\right) \\ &\leq C\left((2\#\Lambda)^s\|\mathbf{u}_\Lambda - \mathbf{u}_{\#\Lambda}\|_{\ell_2(\nabla)} + |\mathbf{u}|_{\ell_\tau^w(\nabla)}\right) \\ &\leq C\left((2\#\Lambda)^s\|\mathbf{u}_\Lambda - \mathbf{u}\|_{\ell_2(\nabla)} + \|\mathbf{u}\|_{\ell_\tau^w(\nabla)}\right), \end{aligned} \tag{4.33}$$

where we have used (3.11) of Proposition 3.2. We add $\|\mathbf{u}_\Lambda\|_{\ell_2(\nabla)}$ to both sides of (4.33) and observe that

$$\|\mathbf{u}_\Lambda\|_{\ell_2(\nabla)} \leq C\|\mathbf{u}\|_{\ell_2(\nabla)} \leq C\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}$$

to finish the proof. □

We next apply this lemma to bound residuals.

Lemma 4.12 *Let $s > 0$, let $\mathbf{A} \in \mathcal{B}_s$ and let the solution $\mathbf{u}$ to (4.1) be in $\ell_\tau^w(\nabla)$. For any index set Λ_k generated by* **Algorithm I**, *we have*

$$\|\mathbf{r}_{\Lambda_{k+1}}\|_{\ell_\tau^w(\nabla)} \leq c_5\left(\|\mathbf{u}\|_{\ell_\tau^w(\nabla)} + \|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}\right), \quad k = 1, 2, \ldots \tag{4.34}$$

with the constant c_5 independent of k and $\mathbf{u}$.

Proof: The algorithm determines the set Λ_{k+1} from Λ_k in the same way for each $k = 1, 2, \ldots$. Therefore, we can assume that $k = 1$. By (4.31) we have

$$\|\mathbf{u}_{\Lambda_2}\|_{\ell_\tau^w(\nabla)} \leq c_4\left(\|\mathbf{u}\|_{\ell_\tau^w(\nabla)} + (\#\Lambda_2)^s\|\mathbf{u} - \mathbf{u}_{\Lambda_2}\|_{\ell_2(\nabla)}\right).$$

We use (2.21) and Theorem 4.5 to bound the second term:

$$(\#\Lambda_2)^s\|\mathbf{u} - \mathbf{u}_{\Lambda_2}\|_{\ell_2(\nabla)} \leq c_1^{-1/2}(\#\Lambda_2)^s\|\mathbf{u} - \mathbf{u}_{\Lambda_2}\| = c_1^{-1/2}C_2^s \lesssim \|\mathbf{f}\|_{\ell_\tau^w(\nabla)} + \|\mathbf{r}_{\Lambda_1}\|_{\ell_\tau^w(\nabla)}.$$

Because of Proposition 3.8 we can replace $\|\mathbf{f}\|_{\ell_\tau^w(\nabla)}$ by $C\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}$. □

5 A Second Adaptive Algorithm

In this section, we shall present a second adaptive algorithm which will meet our goal for the full range of $s > 0$ that is permitted by the wavelet basis. We begin with some heuristics which motivate the structure of the second algorithm.

The deficiency of **Algorithm I** of the last section is that it is only proven to meet our goal for a small range of $s > 0$. This in turn is caused by our inability to prevent the possible growth of $\|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}$ as k increases. Since by assumption $\mathbf{u} \in \ell_\tau^w(\nabla)$, growth in $\|\mathbf{r}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}$ can only occur if $\|\mathbf{u}_{\Lambda_k}\|_{\ell_\tau^w(\nabla)}$ gets large with k. On the other hand, we

know that $\|\mathbf{u}_{\Lambda_k}\|_{\ell_2(\nabla)}$ are bounded uniformly. Typically, for a vector $\mathbf{v}$, its $\ell_\tau^w(\nabla)$ norm is much larger than its $\ell_2(\nabla)$ norm when $\mathbf{v}$ has many small entries which do not effect its $\ell_2(\nabla)$ norm but combine to have a serious effect on the $\ell_\tau^w(\nabla)$ norm. We can try to prevent this from happening by thresholding the coefficients in $\mathbf{v}$ and keeping only the large coefficients. In our application to $\mathbf{u}_{\Lambda_k}$, this is very hopeful since the large coefficients contain the main source of the approximation to $\mathbf{u}$.

Motivated by the above heuristics, we would like to use thresholding in our second algorithm. We introduce the thresholding operator $\mathcal{T}_\eta$ which for $\eta > 0$ and a sequence $\mathbf{v} := (v_\lambda)_{\lambda\in\nabla}$ is defined by

$$(\mathcal{T}_\eta \mathbf{v})_\lambda := \begin{cases} v_\lambda & \text{if} \quad |v_\lambda| \geq \eta; \\ 0 & \text{if} \quad |v_\lambda| < \eta. \end{cases}$$

We shall use the following trivial estimates for thresholding (see §7 of [26]): for any $\mathbf{v} \in \ell_\tau^w(\nabla)$, we have

$$\|\mathbf{v} - \mathcal{T}_\eta \mathbf{v}\|_{\ell_2(\nabla)}^2 = \sum_{|v_\lambda|<\eta} |v_\lambda|^2 \leq c_6^2 \|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^\tau \eta^{2-\tau}, \tag{5.1}$$

and

$$\#\{\lambda : |v_\lambda| \geq \eta\} \leq c_6 \|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^\tau \eta^{-\tau} \tag{5.2}$$

with $c_6 \geq 1$ a constant depending only on τ as $\tau \to 0$.

Lemma 5.1 *Suppose that $\mathbf{v} \in \ell_\tau^w(\nabla)$, $0 < \tau < 2$, and that $\mathbf{w} \in \ell_2(\nabla)$ satisfies*

$$\|\mathbf{v} - \mathbf{w}\|_{\ell_2(\nabla)} \leq \epsilon \tag{5.3}$$

for some $\epsilon > 0$. Then, for any $\eta > 0$, we have

$$\|\mathbf{v} - \mathcal{T}_\eta \mathbf{w}\|_{\ell_2(\nabla)} \leq 2\epsilon + 2c_6 \|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^{\tau/2} \eta^{1-\tau/2}, \tag{5.4}$$

and

$$\#\{\lambda \in \nabla : (\mathcal{T}_\eta \mathbf{w})_\lambda \neq 0\} \leq \frac{4\epsilon^2}{\eta^2} + 4c_6 \|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^\tau \eta^{-\tau}. \tag{5.5}$$

Proof: Let $\mathbf{z} := \mathcal{T}_\eta \mathbf{w}$ and consider the sets $\Lambda_1 := \{\lambda : |w_\lambda| \geq \eta\}$, $\Lambda_2 := \{\lambda : |w_\lambda| < \eta, \text{ and } |v_\lambda| \geq 2\eta\}$, $\Lambda_3 := \{\lambda : |w_\lambda| < \eta \text{ and } |v_\lambda| < 2\eta\}$. Then,

$$\begin{aligned} \|\mathbf{v} - \mathbf{z}\|_{\ell_2(\nabla)}^2 &= \sum_{\lambda\in\Lambda_1\cup\Lambda_2} |v_\lambda - z_\lambda|^2 + \sum_{\lambda\in\Lambda_3} |v_\lambda - z_\lambda|^2 \\ &\leq 4 \sum_{\lambda\in\nabla} |v_\lambda - w_\lambda|^2 + \sum_{|v_\lambda|<2\eta} |v_\lambda|^2 \\ &\leq 4\epsilon^2 + 4c_6^2 \|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^\tau \eta^{2-\tau}, \end{aligned}$$

where we used (5.1) and the fact that $|\mathbf{v}_\lambda| \leq 2|\mathbf{v}_\lambda - \mathbf{w}_\lambda|$ for $\lambda \in \Lambda_2$. This proves (5.4).

For the proof of (5.5), we consider the two sets $\Lambda_4 := \{\lambda : |\mathbf{w}_\lambda| \geq \eta \text{ and } |\mathbf{v}_\lambda| > \eta/2\}$ and $\Lambda_5 := \{\lambda : |\mathbf{w}_\lambda| \geq \eta \text{ and } |\mathbf{v}_\lambda| \leq \eta/2\}$. Then, from (5.2),

$$\#\Lambda_4 \leq \#\{\lambda : |\mathbf{v}_\lambda| > \eta/2\} \leq 2^\tau c_6 \|\mathbf{v}\|^\tau_{\ell^w_\tau(\nabla)} \eta^{-\tau} \leq 4c_6 \|\mathbf{v}\|^\tau_{\ell^w_\tau(\nabla)} \eta^{-\tau}$$

and

$$(\eta/2)^2(\#\Lambda_5) \leq \sum_{\lambda\in\Lambda_5} |\mathbf{v}_\lambda - \mathbf{w}_\lambda|^2 \leq \epsilon^2$$

which proves (5.5). □

We shall use our previous notation which for an integer $N > 0$ and a vector $\mathbf{w} \in \ell_2(\nabla)$ defines $\mathbf{w}_N$ as the vector whose N largest coordinates agree with those of $\mathbf{w}$ and whose other coordinates are zero.

Corollary 5.2 *Suppose that* $\mathbf{v} \in \ell^w_\tau(\nabla)$, $0 < \tau < 2$, *and that* $\mathbf{w} \in \ell_2(\nabla)$ *satisfies*

$$\|\mathbf{v} - \mathbf{w}\|_{\ell_2(\nabla)} \leq \epsilon \tag{5.6}$$

for some $\epsilon > 0$. *Let* $N := N(\epsilon)$ *be chosen as the smallest integer such that*

$$\|\mathbf{w} - \mathbf{w}_N\|_{\ell_2(\nabla)} \leq 4\epsilon \tag{5.7}$$

Then,

$$\|\mathbf{v} - \mathbf{w}_N\|_{\ell_2(\nabla)} \leq 5\epsilon \tag{5.8}$$

and

$$\|\mathbf{v} - \mathbf{w}_N\|_{\ell_2(\nabla)} \leq c_7 \|\mathbf{v}\|_{\ell^w_\tau(\nabla)} N^{-s}, \tag{5.9}$$

with $s := \frac{1}{\tau} - \frac{1}{2}$ *and* c_7 *a constant depending only on* s *as* $s \to \infty$.

Proof: We clearly have (5.8). To prove (5.9), we shall give a bound for N.

In the case where $\|\mathbf{w}\|_{\ell_2(\nabla)} \leq 4\epsilon$, we trivially have (5.7) with $N = 0$ and $\mathbf{w}_N = 0$. In the case where $\|\mathbf{w}\|_{\ell_2(\nabla)} > 4\epsilon$, let η be the absolute value of the smallest nonzero coefficient in $\mathbf{w}_N$. For any $\eta' > \eta$, we have

$$\|\mathbf{w} - \mathcal{T}_{\eta'}\mathbf{w}\|_{\ell_2(\nabla)} > 4\epsilon. \tag{5.10}$$

On account of (5.4), we have

$$\|\mathbf{w} - \mathcal{T}_{\eta'}\mathbf{w}\|_{\ell_2} - \|\mathbf{v} - \mathbf{w}\|_{\ell_2(\nabla)} \leq \|\mathbf{v} - \mathcal{T}_{\eta'}\mathbf{w}\|_{\ell_2(\nabla)} \leq 2\epsilon + 2c_6\|\mathbf{v}\|^{\tau/2}_{\ell^w_\tau(\nabla)}(\eta')^{1-\tau/2},$$

so that (5.6) and (5.10) ensure that

$$\epsilon < 2c_6\|\mathbf{v}\|^{\tau/2}_{\ell^w_\tau(\nabla)}(\eta')^{1-\tau/2} \tag{5.11}$$

holds for all $\eta' > \eta$. Therefore,

$$\epsilon \leq 2c_6\|\mathbf{v}\|^{\tau/2}_{\ell^w_\tau(\nabla)}\eta^{1-\tau/2} = 2c_6\|\mathbf{v}\|^{\tau/2}_{\ell^w_\tau(\nabla)}\eta^{s\tau}. \tag{5.12}$$

On the other hand, from (5.5) we find

$$N \le \#\{\lambda \in \nabla : (\mathcal{T}_\eta \mathbf{w})_\lambda \neq 0\} \le \frac{4\epsilon^2}{\eta^2} + 4c_6 \|\mathbf{v}\|^\tau_{\ell^w_\tau(\nabla)} \eta^{-\tau}. \tag{5.13}$$

We use (5.12) to estimate each of the two terms on the right of (5.13). For example, for the second term, we have

$$4c_6 \|\mathbf{v}\|^\tau_{\ell^w_\tau(\nabla)} \eta^{-\tau} \le 2(2c_6)^{1+1/s} \|\mathbf{v}\|^{\tau(1+\frac{1}{2s})}_{\ell^w_\tau(\nabla)} \epsilon^{-1/s} = 2(2c_6)^{1+1/s} \|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} \epsilon^{-1/s}. \tag{5.14}$$

A similar estimate shows that the first term on the right of (5.13) does not exceed $4(2c_6)^{1+1/s} \|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} \epsilon^{-1/s}$. In other words,

$$N \le 6(2c_6)^{1+1/s} \|\mathbf{v}\|^{\tau(1+\frac{1}{2s})}_{\ell^w_\tau(\nabla)} \epsilon^{-1/s} = (c_7/5)^{1/s} \|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} \epsilon^{-1/s}, \tag{5.15}$$

where the last equality serves to define c_7. When this estimate for N is used in (5.8), we arrive at (5.9). □

Algorithm II will modify **Algorithm I** by the introduction of the following step:

COARSE: *Given a set Λ and a Galerkin solution $\mathbf{u}_\Lambda$ associated to this set, take $\epsilon := c_1^{-1} \|\mathbf{r}_\Lambda\|_{\ell_2(\nabla)}$ and apply Corollary 5.2 with $\mathbf{v} := \mathbf{u}$ and $\mathbf{w} := \mathbf{u}_\Lambda$ to produce the vector $\mathbf{w}_N$. Then,* **COARSE** *produces the set $\tilde{\Lambda}$ of indices for the nonzero coordinates of $\mathbf{w}_N$ and then applies* **GALERKIN** *to this new set to obtain $\mathbf{u}_{\tilde{\Lambda}}$.*

Remark 5.3 *If Λ is any set, it follows from Corollary 5.2 that the input of Λ into* **COARSE** *yields a set $\tilde{\Lambda}$ with a Galerkin solution $\mathbf{u}_{\tilde{\Lambda}}$ which satisfies*

$$\|\mathbf{u} - \mathbf{u}_{\tilde{\Lambda}}\| \le c_8 \|\mathbf{u}\|_{\ell^w_\tau(\nabla)} (\#\tilde{\Lambda})^{-s}, \tag{5.16}$$

where $c_8 = c_2^{1/2} c_7$ with c_2 from (2.21) *and c_7 from* (5.9).

Proof: We have

$$\|\mathbf{u} - \mathbf{u}_{\tilde{\Lambda}}\| \le \|\mathbf{u} - \mathbf{w}_N\| \le c_2^{1/2} \|\mathbf{u} - \mathbf{w}_N\|_{\ell_2(\nabla)}, \tag{5.17}$$

because the Galerkin projection gives the best approximation $\mathbf{u}_\Lambda$ to $\mathbf{u}$ from $\ell_2(\Lambda)$ in the energy norm. We bound the right side of (5.17) by (5.9). □

Remark 5.4 *It also follows from Corollary 5.2 together with Lemma 4.11 that the input of Λ into* **COARSE** *yields a set $\tilde{\Lambda}$ with a Galerkin solution $\mathbf{u}_{\tilde{\Lambda}}$ such that*

$$\|\mathbf{u}_{\tilde{\Lambda}}\|_{\ell^w_\tau(\nabla)} \le c_9 \|\mathbf{u}\|_{\ell^w_\tau(\nabla)} \tag{5.18}$$

with the constant c_9 depending only on τ as $\tau \to 0$. Of course, this also implies that $\|\mathbf{r}_{\tilde{\Lambda}}\|_{\ell^w_\tau(\nabla)} \lesssim \|\mathbf{u}\|_{\ell^w_\tau(\nabla)}$. Thus, the thresholding step allows the control of the $\ell^w_\tau(\nabla)$ norm of the residual.

We can now describe **Algorithm II**.

Algorithm II:

- Let $\Lambda_0 = \emptyset$ and $\mathbf{r}_{\Lambda_0} = \mathbf{f}$.
- For $j = 0, 1, 2, \ldots$, determine Λ_{j+1} from Λ_j as follows. Let $\Lambda_{j,0} := \Lambda_j$. For $k = 1, 2, \ldots$ determine $\Lambda_{j,k}$ from $\Lambda_{j,k-1}$ by applying **GALERKIN** and then **GROW** to $\Lambda_{j,k-1}$. Apply **COARSE** to $\Lambda_{j,k}$ to determine $\tilde{\Lambda}_{j,k}$ and $\mathbf{u}_{\tilde{\Lambda}_{j,k}}$. If $\|\mathbf{r}_{\tilde{\Lambda}_{j,k}}\|_{\ell_2(\nabla)} \leq \frac{1}{2}\|\mathbf{r}_{\Lambda_j}\|_{\ell_2(\nabla)}$ then define $\Lambda_{j+1} := \tilde{\Lambda}_{j,k}$, $k_j := k$, and stop the iteration on k. Otherwise advance k and continue.

Theorem 5.5 *If $\mathbf{A} \in \mathcal{B}_s$, for some $s > 0$, then* **Algorithm II** *satisfies our goal for this s. Namely, if $\mathbf{u} \in \ell^w_\tau(\nabla)$, then the algorithm produces sets Λ_j, $j = 1, 2, \ldots$, such that*

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\| \leq c_8 (\#\Lambda_j)^{-s} \|\mathbf{u}\|_{\ell^w_\tau(\nabla)} \tag{5.19}$$

with c_8 the constant of Remark 5.3. In addition, for $j = 1, 2, \ldots$, we have

$$\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\|_{\ell_2(\nabla)} \leq c_1^{-1} c_2 \|\mathbf{u}\|_{\ell_2(\nabla)} 2^{-j} \tag{5.20}$$

with c_1 and c_2 the constants of (2.21).

Proof: Since the set Λ_j is the output of **COARSE**, the estimate (5.19) follows from Remark 5.3. By the definition $\Lambda_{j+1} := \tilde{\Lambda}_{j,k_j}$ in **Algorithm II**, we have

$$\|\mathbf{r}_{\Lambda_{j+1}}\|_{\ell_2(\nabla)} = \|\mathbf{r}_{\tilde{\Lambda}_{j,k_j}}\|_{\ell_2(\nabla)} \leq \frac{1}{2}\|\mathbf{r}_{\Lambda_j}\|_{\ell_2(\nabla)}. \tag{5.21}$$

Iterating this inequality, we obtain

$$\|\mathbf{r}_{\Lambda_j}\|_{\ell_2(\nabla)} \leq 2^{-j}\|\mathbf{r}_{\Lambda_0}\|_{\ell_2(\nabla)} = 2^{-j}\|\mathbf{f}\|_{\ell_2(\nabla)} \leq c_2 2^{-j}\|\mathbf{u}\|_{\ell_2(\nabla)}.$$

Since, $\|\mathbf{u} - \mathbf{u}_{\Lambda_j}\|_{\ell_2(\nabla)} \leq c_1^{-1}\|\mathbf{r}_{\Lambda_j}\|_{\ell_2(\nabla)}$, we arrive at (5.20). □

The following remark will be important in the following section on numerical computation. It shows that the intermediate steps between Λ_j and Λ_{j+1} do not generate sets $\Lambda_{j,k}$ which might be very large in comparison to Λ_j and Λ_{j+1}.

Remark 5.6 *Under the assumptions of Theorem 5.5 we have*

$$k_j \leq K, \quad j = 1, 2, \ldots, \tag{5.22}$$

with K the smallest integer such that $5c_1^{-2}c_2^2\theta^K \leq 1/2$ where c_1, c_2 are the constants of (2.21) *and θ is given by* (4.13).

Proof: This follows from the following string of inequalities, where we denote by $\mathbf{w}^k$ the intermediate output of **COARSE** obtained by thresholding $\mathbf{u}_{\Lambda_{j,k}}$ before computing the new Galerkin solution:

$$\begin{aligned}
\|\mathbf{r}_{\tilde{\Lambda}_{j,k}}\|_{\ell_2(\nabla)} &\leq c_2\|\mathbf{u}-\mathbf{u}_{\tilde{\Lambda}_{j,k}}\|_{\ell_2(\nabla)}\\
&\leq c_2c_1^{-1/2}\|\mathbf{u}-\mathbf{u}_{\tilde{\Lambda}_{j,k}}\|\\
&\leq c_2c_1^{-1/2}\|\mathbf{u}-\mathbf{w}^k\|\\
&\leq c_2c_1^{-1/2}c_2^{1/2}\|\mathbf{u}-\mathbf{w}^k\|_{\ell_2(\nabla)}\\
&\leq 5c_2c_1^{-1/2}c_2^{1/2}c_1^{-1}\|\mathbf{r}_{\Lambda_{j,k}}\|_{\ell_2(\nabla)}\\
&\leq 5c_2^2c_1^{-3/2}\|\mathbf{u}-\mathbf{u}_{\Lambda_{j,k}}\|\\
&\leq 5c_2^2c_1^{-3/2}\theta^k\|\mathbf{u}-\mathbf{u}_{\Lambda_j}\|\\
&\leq 5c_1^{-2}c_2^2\theta^k\|\mathbf{r}_{\Lambda_j}\|_{\ell_2(\nabla)}.
\end{aligned}$$

The fifth inequality follows from (5.8) and the fact that $\epsilon = c_1^{-1}\|\mathbf{r}_{\Lambda_{j,k}}\|_{\ell_2(\nabla)}$ in the application of **COARSE** to $\Lambda_{j,k}$. All other inequalities use norm equivalences of the type (2.21). From this estimate we see that the criterion $\|\mathbf{r}_{\Lambda_{j,k}}\|_{\ell_2(\nabla)} \leq \frac{1}{2}\|\mathbf{r}_{\Lambda_j}\|_{\ell_2(\nabla)}$ is met whenever $k > K$. □

Note that this remark, combined with Lemma 4.12, has the following consequence.

Remark 5.7 *The residuals in the intermediate steps $\mathbf{r}_{\Lambda_{j,k}}$ are also uniformly bounded in $\ell_\tau^w(\nabla)$ and that the cardinalities $\#\Lambda_{k,j}$ can always be controlled by $\#\Lambda_j$. The intermediate steps remain within our goal of optimal accuracy with respect to the number of parameters.*

Theorem 5.5 shows that **Algorithm II** is optimal for the full range of s permitted by the wavelet bases. By the same considerations as in Remark 4.7, this algorithm is also optimal in the sense of achieving a prescribed tolerance ϵ.

6 Numerical Realization: Basic Ingredients

The previous sections have introduced and analyzed the performance of two adaptive methods for resolving elliptic equations. The analysis however was more from a theoretical perspective and did *not* incorporate computational issues. Our purpose is now to address these computational issues. More precisely, we want to develop a *numerically realizable* version of **Algorithm II** and to analyze its complexity. In the present section, we shall introduce the basic subroutines that constitute the resulting **Algorithm III** which will be described and analyzed in the final section.

Let us first explain the basic principle of **Algorithm III**. This algorithm will iteratively generate a sequence of index sets Λ_j and approximate solutions $\bar{\mathbf{u}}_{\Lambda_j}$ supported in Λ_j ($\bar{\mathbf{u}}_{\Lambda_j}$ differs in general from the Galerkin solution $\mathbf{u}_{\Lambda_j}$), with the property that

$$\|\mathbf{u}-\bar{\mathbf{u}}_{\Lambda_j}\|_{\ell_2(\nabla)} \leq \epsilon_j := 2^{-j}\epsilon_0, \tag{6.1}$$

where ϵ_0 is an estimate from above of $\|\mathbf{u}\|_{\ell_2(\nabla)}$ (which will allow us to take as an admissible starting point $\mathbf{u}_{\Lambda_0} = 0$ and Λ_0 empty). This progression toward finer accuracy will be performed by the main subtroutine **PROG** which will be assembled in §7 from the ingredients that we shall introduce in the present section.

If we are given a tolerance ϵ that gives the target accuracy with which we wish to resolve the solution to (2.17), we shall thus need J applications of **PROG** where J is the smallest integer such that $\epsilon_J \leq \epsilon$.

We then ask what the total computational cost will be to attain this accuracy. There will be two sources of computational expense: arithmetic operations and sorting. Arithmetic operations include additions, multiplications, and square roots. We shall ignore error due to roundoff. We shall estimate the number of arithmetic computations $N(\epsilon)$ and the number of sorts $\tilde{N}(\epsilon)$ needed to achieve this accuracy. We shall see that $N(\epsilon)$ can be related to ϵ in the same way that the error analysis of the preceding section related error to the size of the sets Λ_j. The sorting will introduce an additional logarithmic factor.

Our subroutines will be described so as to apply to *any* vectors and thereby **Algorithm III** will allow us to solve (2.17) for *any* right hand side $\mathbf{f}$. However, we shall analyze its performance only when $\mathbf{u}$ is in $\ell^w_\tau(\nabla)$, $\tau := (s+1/2)^{-1}$, for some $s > 0$ in the same range of optimality as for **Algorithm II**. Note that this range is limited only by s^* in (3.16), i.e. the compressibility order of the operator in the wavelet bases.

Our analysis will show that if $\mathbf{u}$ has the ℓ^w_τ smoothness, then the computational cost and memory size needed to achieve accuracy ϵ_j is controlled by $\|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)} 2^{j/s}$ so that the last step $J-1 \mapsto J$ dominates the overall computational cost. This should be compared to the optimality analysis of the full multigrid algorithm, for which the complexity is also dominated by the last step of the nested iteration. However, in the multigrid algorithm, each step of the nested iteration is associated to a uniform discretization at a scale j, which corresponds to imposing that Λ_j is the set of all indices $|\lambda| \leq j$, rather than an adaptive set. In this case, the new layer Λ_{j+1}/Λ_j updating the computation thus corresponds to a *scale* level, while in our adaptive algorithm it is rather associated to a certain *size* level of the wavelet coefficients of $\mathbf{u}$. Accordingly the classical Sobolev smoothness which enters the analysis of multigrid algorithms is replaced by the weaker Besov smoothness expressed by the ℓ^w_τ property.

Algorithm III will involve numerical versions of procedures like **GROW**, **COARSE** or **GALERKIN**. In these subroutines *exact calculations* will have to be replaced by *approximate counterparts* whose accuracy is controlled by corresponding parameters. Thus the input will consist of the objects like index sets or vectors to be processed as well as control and threshold parameters. To keep track of these parameters and their interdependencies we will consistently use the following format for such subroutines
NAME$[IP_1, \ldots, IP_\ell] \;\rightarrow\; (OP_1, \ldots, OP_r)$, meaning that given the input $IP_1, \ldots, IP_\ell$ the procedure **NAME** generates output quantities $OP_1, \ldots, OP_r$.

Some of the subroutines will make use of estimates of several constants like c_1, c_2 from previous sections, in which case we shall specify them and explain how such estimates can be obtained. All other constants entering the analysis of the algorithm but not its concrete implementation will be denoted by C without further distinction. Their specific value only

affects the constant in our asymptotic estimates. In particular, they are independent of the data $\mathbf{f}$ the solution $\mathbf{u}$ or its various approximations. If necessary their dependence on the parameters τ or s will be explained.

6.1 The Assembly of f

We shall take the viewpoint that we have *complete* knowledge about the data f in the sense that we already know or can compute its wavelet coefficient to any desired accuracy by an appropriate quadrature. This in turn enables us to approximate $\mathbf{f}$ to any accuracy by a finite wavelet expansion. We formulate this as

Assumption N1: *We assume that for any given tolerance $\eta > 0$, we are provided with the set $\bar{\Lambda} := \bar{\Lambda}(f, \eta)$ of minimal size such that $\bar{\mathbf{f}} = \mathbf{P}_{\bar{\Lambda}}\mathbf{f}$ satisfies*

$$\|\mathbf{f} - \bar{\mathbf{f}}\|_{\ell_2(\nabla)} \leq \eta. \tag{6.2}$$

For the purpose of our asymptotic analysis, we could actually replace "minimal" by "nearly minimal", in the sense that the following property holds: if $\mathbf{f}$ is in $\ell^w_\tau(\nabla)$ for some $\tau < 2$, then we have the estimate

$$\#(\bar{\Lambda}) \leq C\eta^{-1/s}\|\mathbf{f}\|^{1/s}_{\ell^w_\tau(\nabla)}, \tag{6.3}$$

with $s = 1/2 - 1/\tau$ and C a constant that depends only on s as s tends to $+\infty$. This modified assumption is much more realistic, since in practice one can only have approximate knowledge of the index set corresponding to the largest coefficients in $\mathbf{f}$, using some a-priori information on the smooth and singular parts of the function f. However, in order to simplify the notation and analysis in what follows we shall assume that the set $\bar{\Lambda}$ is minimal.

In the implementation of **Algorithm III**, the above tolerance η will typically be related to the target accuracy ϵ of the solution by a fixed multiplicative constant. We perform the following two preprocessing steps on $\bar{\mathbf{f}}$:

(i) sort the entries in $\bar{\mathbf{f}}$ to determine the vector $\lambda^* = (\lambda_1, \lambda_2, \ldots, \lambda_{\bar{N}})$ of indices which gives the decreasing rearrangement $\bar{\mathbf{f}}^* = (|f_{\lambda_1}|, |f_{\lambda_2}|, \ldots, |f_{\lambda_{\bar{N}}}|)$. The cost of this operation is in $\mathcal{O}(\bar{N}\log\bar{N})$ operations.

(ii) compute $F^2 := \|\bar{\mathbf{f}}\|^2_{\ell_2(\nabla)} + \eta^2 = \sum_{i=1}^{\bar{N}} |f_{\lambda_i}|^2 + \eta^2$. The cost of this is $2\bar{N} - 1$ arithmetic operations.

The second step gives us the estimate $\|\mathbf{f}\|_{\ell_2(\nabla)} \leq F$. We store F and the vector λ^*.

6.2 A Numerical Version of COARSE

Algorithm III will also make use of less accurate approximations of $\mathbf{f}$ in its intermediate steps. This is one instance of the frequent need to provide a good coarser approximation

to a finitely supported vector. Such approximations will be generated by the routine **NCOARSE** that we shall now describe.

NCOARSE $[\mathbf{w}, \eta] \to (\Lambda, \bar{\mathbf{w}})$:

(i) *Define* $N := \#(\operatorname{supp} \mathbf{w})$ *and sort the nonzero entries of* $\mathbf{w}$ *into decreasing order. Thereby one obtains the vector* $\lambda^* := \lambda^*(\mathbf{w}) = (\lambda_1, \lambda_2, \cdots, \lambda_N)$ *of indices which gives the decreasing rearrangement* $\mathbf{w}^* = (|\mathbf{w}_{\lambda_1}|, |\mathbf{w}_{\lambda_2}|, \cdots, |\mathbf{w}_{\lambda_N}|)$ *of the nonzero entries of* $\mathbf{w}$*; then compute* $\|\mathbf{w}\|^2_{\ell_2(\nabla)} = \sum_{i=1}^N |\mathbf{w}_{\lambda_i}|^2$.

(ii) *For* $k = 1, 2, \cdots$*, form the sum* $\sum_{j=1}^k |\mathbf{w}_{\lambda_j}|^2$ *in order to find the smallest value* k *such that this sum exceeds* $\|\mathbf{w}\|^2_{\ell_2(\nabla)} - \eta^2$*. For this* k*, define* $K := k$ *and set* $\Lambda := \{\lambda_j \,;\, j = 1, \cdots, K\}$*; define* $\bar{\mathbf{w}}$ *by* $\bar{w}_\lambda := w_\lambda$ *for* $\lambda \in \Lambda$ *and* $\bar{w}_\lambda := 0$ *for* $\lambda \notin \Lambda$.

We first describe the computational cost of **NCOARSE**.

Properties 6.1 *For any* $\mathbf{w}$*, and* η*, we need at most* $2N$ **arithmetic operations** *and* $N \log N$ **sorts**, $N := \#(supp\, \mathbf{w})$*, to compute the output* $\bar{\mathbf{w}}$ *of* **NCOARSE** *which, by construction, satisfies*

$$\|\mathbf{w} - \bar{\mathbf{w}}\|_{\ell_2(\nabla)} \le \eta. \tag{6.4}$$

We shall also apply **NCOARSE** to the initial approximation $\bar{\mathbf{f}}$ of the data in order to produce other near optimal N-term approximations of $\mathbf{f}$ with fewer parameters. Thanks to the preprocessing steps, in this case we can save on the computational cost of this procedure. An immediate consequence of (5.15) and Properties 6.1 is the following.

Properties 6.2 *Assume that* $\bar{\mathbf{f}}$ *is an optimal* $\bar{N}$*-term approximation of the data* $\mathbf{f}$ *with accuracy* η*, as described by* (6.2)*. Then, for* $\tilde{\eta} \ge \eta$, **NCOARSE** $[\bar{\mathbf{f}}, \tilde{\eta} - \eta]$ *produces an approximation* $\mathbf{g}$ *to* $\mathbf{f}$ *with support* Λ*, such that* $\|\mathbf{g} - \mathbf{f}\|_{\ell_2(\nabla)} \le \tilde{\eta}$*. In addition, if* $\mathbf{f} \in \ell^w_\tau(\nabla)$*, we have*

$$\#(\Lambda) \le C \tilde{\eta}^{-1/s} \|\mathbf{f}\|^{1/s}_{\ell^w_\tau(\nabla)}, \tag{6.5}$$

with C *depending only on* s*.*

Moreover, determining $\mathbf{g}$ *requires at most* $2\#(\Lambda)$ *arithmetic operations and no sorts since sorting of* $\bar{\mathbf{f}}$ *was done in the preprocessing stage.*

To simplify notation we will denote throughout the remainder of the paper by **NCOARSE** $[\mathbf{f}, \tilde{\eta}]$ the output of **NCOARSE** $[\bar{\mathbf{f}}, \tilde{\eta} - \eta]$, since it has the same optimal approximation properties as thresholding the exact data.

We now turn to the primary purpose of the coarsening procedure. Recall that the role of **COARSE** in **Algorithm II** above is the following. If $\mathbf{v}$ is a given vector from $\ell^w_\tau(\nabla)$ and $\mathbf{w}$ is a good (finitely supported) approximation to $\mathbf{v}$ in the $\ell_2(\nabla)$-norm but has large $\ell^w_\tau(\nabla)$ norm then **COARSE** uses thresholding to produce a new approximation with slightly worse $\ell_2(\nabla)$-approximation properties but guaranteed good $\ell^w_\tau(\nabla)$ norms. The following algorithm gives the numerical form of **COARSE** that we shall use. The following additional properties of **NCOARSE** follow from the results in § 5.

Properties 6.3 *Given a vector* $\mathbf{v}$*, a tolerance* $0 < \eta \leq \|\mathbf{v}\|_{\ell_2(\nabla)}$*, and a finitely supported approximation* $\mathbf{w}$ *to* $\mathbf{v}$ *that satisfies*

$$\|\mathbf{v} - \mathbf{w}\|_{\ell_2(\nabla)} \leq \eta/5, \tag{6.6}$$

the algorithm **NCOARSE** $[\mathbf{w}, 4\eta/5]$ *produces a new approximation* $\bar{\mathbf{w}}$ *to* $\mathbf{v}$*, supported on* Λ*, which satisfies*

$$\|\mathbf{v} - \bar{\mathbf{w}}\|_{\ell_2(\nabla)} \leq \eta. \tag{6.7}$$

Moreover, the following properties hold:

(i) *If* $\mathbf{v} \in \ell^w_\tau(\nabla)$*,* $\tau = (s + 1/2)^{-1}$*, for some* $s > 0$*, then the outputs* $\bar{\mathbf{w}}$ *and* Λ *of* **NCOARSE** *satisfy*

$$\|\mathbf{v} - \bar{\mathbf{w}}\|_{\ell_2(\nabla)} \leq C\|\mathbf{v}\|_{\ell^w_\tau(\nabla)} \#(\Lambda)^{-s}. \tag{6.8}$$

(ii) *If* $\mathbf{v} \in \ell^w_\tau(\nabla)$*,* $\tau = (s + 1/2)^{-1}$*, for some* $s > 0$*, then the output* $\bar{\mathbf{w}}$ *of* **NCOARSE** *satisfies*

$$\|\bar{\mathbf{w}}\|_{\ell^w_\tau(\nabla)} \leq C\|\mathbf{v}\|_{\ell^w_\tau(\nabla)}, \tag{6.9}$$

where C *depends only on* s *as* $s \to \infty$*.*

(iii) *The cardinality of the support* Λ *of* $\bar{\mathbf{w}}$ *is bounded by*

$$\#(\Lambda) \leq C\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} \eta^{-1/s}. \tag{6.10}$$

Proof: The estimate (6.7) is an immediate consequence of the steps in **NCOARSE**. (i) follows from Corollary 5.2 (see (5.9)). (ii) is proved in a similar fashion to Lemma 4.11. Let $K := \#(\Lambda)$ and let $\mathbf{v}_K$ be the best approximation to $\mathbf{v}$ from Σ_K. Then, as in (4.33), we derive

$$\begin{aligned} |\bar{\mathbf{w}}|_{\ell^w_\tau(\nabla)} &\leq C\left(|\bar{\mathbf{w}} - \mathbf{v}_K|_{\ell^w_\tau(\nabla)} + |\mathbf{v}_K|_{\ell^w_\tau(\nabla)}\right) \\ &\leq C\left((2K)^s\|\bar{\mathbf{w}} - \mathbf{v}_K\|_{\ell_2(\nabla)} + |\mathbf{v}|_{\ell^w_\tau(\nabla)}\right) \\ &\leq C\left(K^s\|\mathbf{v} - \bar{\mathbf{w}}\|_{\ell_2(\nabla)} + \|\mathbf{v}\|_{\ell^w_\tau(\nabla)}\right), \end{aligned} \tag{6.11}$$

where we used (4.32). We insert (6.8) into (6.11) and add $\|\mathbf{w}\|_{\ell_2(\nabla)}$ to both sides and arrive at (6.9). The estimate (iii) is an immediate consequence of (5.15). □

6.3 The Assembly of A

We shall need to compute a certain finite number of entries of $\mathbf{A}$. The entries that need to be computed will be prescribed as the adaptive algorithm proceeds and are not known in advance. They are associated to the application of one of the compressed matrices $\mathbf{A}_k$ to a finite vector $\mathbf{v}$, as will be discussed below. Therefore, the entries are computed as their need arises. When we compute an entry of $\mathbf{A}$ we store it for future possible use. We

shall make the following

Assumption N2: *Any entry $a_{\lambda,\mu}$ of* $\mathbf{A}$ *can be computed (up to roundoff error) at unit cost.*

In some cases, this assumption is completely justified. For example, if the operator A is a constant coefficient differential operator and the domain is a union of cubes, then suitable biorthogonal wavelet bases are known where the primal multiresolution spaces are generated by $B-$splines. In this case, the functions which appear in the integrals defining the entries of $\mathbf{A}$ are piecewise polynomials. Therefore, they can be computed exactly. When A is a differential operator with varying coeffcicients or when A is a singular integral operator the entries of $\mathbf{A}$ have to be approximated with an accuracy depending on the desired final accuracy ϵ of the solution. It is then by far less obvious how to realize **Assumption N2** and a detailed discussion of this issue (which very much depends on the individual concrete properties of A) is beyond the scope of this paper. We therefore content ourselves with the following indications that **(N2)** is not quite unreasonable. A central issue in [23, 35, 36] is to design suitably adapted quadrature schemes for computing the significant entries of wavelet representation of the underlying singular integral operator in the following sense. The trial spaces under consideration are spanned by *all* wavelets up to a highest level J, say. Then, it is shown how to compute a compressed matrix having only the order of $N_J = 2^{Jd}$ nonzero entries (up to possible log factors in some studies) at a computational expense which also stays proportional to N_J (again possibly times a log factor). Since the compression in these papers is slightly different from the one used here and since only fully refined spaces have been investigated these results do not apply here directly. Nevertheless, they indicate that the development of schemes that keep the computational work per entry low is not completely unrealistic.

In the development of the numerical algorithm, we shall need constants c_1 and c_2 such that (2.21) holds. In practice, it is not difficult to obtain sharp estimates of the optimal constants since as J grows, they are well approximated by the smallest and largest eigenvalues of the preconditioned matrix $\mathbf{A}_{\nabla_J}$ corresponding to the set $\nabla_J = \{\lambda \in \nabla \,;\, |\lambda| < J\}$ associated to the uniform discretization through the trial space S_{∇_J}. For simplicity we will take $\kappa := c_2/c_1$ as an estimate for the condition number of $\mathbf{A}$, see (2.24).

We next discuss the quasi-sparsity assumptions that we shall make on the matrix $\mathbf{A}$.

Assumption N2: *We assume that the matrix* $\mathbf{A}$ *is quasi-sparse in the sense that it is known to be in the class $\mathcal{B}_s$ of §3.3 for $s < s^*$ for some $s^* > 0$. We recall that $\mathbf{A} \in \mathcal{B}_s$ implies that for each $k = 1, 2, \ldots$, there is a matrix $\mathbf{A}_k$, with at most $2^k\alpha_k$ entries in each row and column, that satisfies*

$$\|\mathbf{A} - \mathbf{A}_k\| \leq C2^{-ks}\alpha_k, \tag{6.12}$$

with $(\alpha_k)_{k=0}^{\infty}$ a summable sequence of positive numbers. We will assume that the positions of the entries in $\mathbf{A}_k$ are known to us.

We have discussed previously how this assumption follows from the original elliptic equations and the wavelet basis. In particular, they are implied by decay properties of the type (2.30). The compression rules leading to the matrices $\mathbf{A}_k$ and, in particular, the positions of the significant entries are in this case explicitly given in the proof of Proposition 3.4 and depend only on k.

In the development of the numerical algorithm, we shall make use of the estimates (6.12) in the form

$$\|\mathbf{A} - \mathbf{A}_k\| \leq a_k, \tag{6.13}$$

where the constants a_k are upper bounds for the compression error $\|\mathbf{A} - \mathbf{A}_k\|$. The a_k might simply correspond to a rough estimate of C in (6.12) or result from a more precise estimate of $\|\mathbf{A} - \mathbf{A}_k\|$ that can in practice be obtained by means of the Schur lemma.

The entries we compute in $\mathbf{A}_k$ are determined by the vectors to which $\mathbf{A}_k$ is applied. We only apply $\mathbf{A}_k$ to vectors $\mathbf{v}$ with finite support.To compute $\mathbf{A}_k\mathbf{v}$ requires only that we know the nonzero entries of $\mathbf{A}_k$ in the columns corresponding to the nonzero entries of $\mathbf{v}$. Hence, at most $\alpha_k 2^k(\#\mathrm{supp}\,\mathbf{v})$ entries will need to be computed. We shall keep track of the number of these computations in the analysis that follows.

6.4 Matrix/Vector Multiplication

It is clear from **Algorithm II** that the main numerical tasks are the computation of Galerkin solutions and the evaluation of residuals. Both rest on the repeated application of the quasi-sparse matrix $\mathbf{A}$ to a vector $\mathbf{v}$ with finite support. Since the matrices and vectors are in general only *quasi-sparse* this operation can be carried out only *approximately* in order to retain efficiency. For this, we shall use the algorithm of §3.3 applied to $\mathbf{B} = \mathbf{A}$. We recall our convention concerning the application of an infinite matrix to a finite vector : we consider the vector to be extended to the infinite vector on Δ obtained by setting all new entries to be zero. The extended vector will also be denoted by $\mathbf{v}$.

Given a vector $\mathbf{v}$ of finite support and $N = \#\mathrm{supp}\,\mathbf{v}$, we sort the entries of $\mathbf{v}$ and form the vectors $\mathbf{v}_{[0]}$, $\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}$, $j = 1, \cdots, \lfloor \log N \rfloor$. For $j > \log N$, we define $\mathbf{v}_{[j]} := \mathbf{v}$. Recall from § 3 that $\mathbf{v}_{[j]}$ agrees with $\mathbf{v}$ in its 2^j largest entries and is zero otherwise. This process requires at most $N \log N$ sorts.

We shall numerically approximate $\mathbf{A}\mathbf{v}$ by using the vector

$$\mathbf{w}_k := \mathbf{A}_k\mathbf{v}_{[0]} + \mathbf{A}_{k-1}(\mathbf{v}_{[1]} - \mathbf{v}_{[0]}) + \cdots + \mathbf{A}_0(\mathbf{v}_{[k]} - \mathbf{v}_{[k-1]}) \tag{6.14}$$

for a certain value of k determined by the desired numerical accuracy. As noted ealier, this vector can be computed by using $\leq C_1 2^k$ operations and requires the computation of at most this same number of entries in $\mathbf{A}$, recall Corollary 3.10. Note that if $2^k > \#(\mathrm{supp}\,\mathbf{v})$, then some of the terms in (6.14) will be zero and therefore need not be computed.

By increasing k, we increase the accuracy of the approximation $\mathbf{w}_k$ to $\mathbf{Av}$. In particular, as derived in §3.3, see (3.25), we have the error estimate

$$\|\mathbf{Av} - \mathbf{w}_k\|_{\ell_2(\nabla)} \le c_2\|\mathbf{v} - \mathbf{v}_{[k]}\|_{\ell_2(\nabla)} + a_k\|\mathbf{v}_{[0]}\|_{\ell_2(\nabla)} + \sum_{j=0}^{k-1} a_j\|\mathbf{v}_{[k-j]} - \mathbf{v}_{[k-j-1]}\|_{\ell_2(\nabla)}, \quad (6.15)$$

where a_j is the compression bound from (6.13). Note that $\|\mathbf{v} - \mathbf{v}_{[j]}\|^2_{\ell_2(\nabla)} = \|\mathbf{v}\|^2_{\ell_2(\nabla)} - \|\mathbf{v}_{[j]}\|^2_{\ell_2(\nabla)}$ and $\|\mathbf{v}_{[j]}\|^2_{\ell_2(\nabla)} = \sum_{l=1}^{j}\|\mathbf{v}_{[l]} - \mathbf{v}_{[l-1]}\|^2_{\ell_2(\nabla)}$ Hence, the right hand side of (6.15) can be computed for any k with at most $C(\#(\operatorname{supp}\mathbf{v}))$ operations.

With these remarks in hand, we introduce the following numerical procedure for approximating $\mathbf{Av}$.

APPLY A $[\eta, \mathbf{v}] \to (\mathbf{w}, \Lambda)$:

(i) *Sort the nonzero entries of the vector* $\mathbf{v}$ *and form the vectors* $\mathbf{v}_{[0]}$, $\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}$, $j = 1, \cdots, \lfloor \log N \rfloor$ *with* $N := \#supp\,\mathbf{v}$. *Define* $\mathbf{v}_{[j]} := \mathbf{v}$ *for* $j > \log N$.

(ii) *Compute* $\|\mathbf{v}\|^2_{\ell_2(\nabla)}$, $\|\mathbf{v}_{[0]}\|^2_{\ell_2(\nabla)}$, $\|\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\|^2_{\ell_2(\nabla)}$, $j = 1, \cdots, \lfloor \log N \rfloor + 1$.

(iii) *Set* $k = 0$.

- **(a)** *Compute the right hand side* R_k *of* (6.15) *for the given value of* k.
- **(b)** *If* $R_k \le \eta$ *stop and output* k; *otherwise replace* k *by* $k + 1$ *and return to (a).*

(iv) *For the output* k *of (iii) and for* $j = 0, 1, \cdots, k$, *compute the nonzero entries in the matrices* $\mathbf{A}_{k-j}$ *which have a column index in common with one of the nonzero entries of* $\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}$.

(v) *For the output* k *of (iii), compute* $\mathbf{w}_k$ *as in* (6.14) *and take* $\mathbf{w}(\mathbf{v}, \eta) := \mathbf{w}_k$ *and* $\Lambda = \operatorname{supp}\mathbf{w}$.

Properties 6.4 *Given a tolerance* $\eta > 0$ *and a vector* $\mathbf{v}$ *with finite support, the algorithm* **APPLY A** *produces a vector* $\mathbf{w}(\mathbf{v}, \eta)$ *which satisfies*

$$\|\mathbf{Av} - \mathbf{w}\|_{\ell_2(\nabla)} \le \eta. \quad (6.16)$$

Moreover, if $\mathbf{v} \in \ell^w_\tau(\nabla)$, *with* $\tau = (s + 1/2)^{-1/2}$ *and* $0 < s < s^*$, *then the following properties hold:*

(*i*) *The size of the output* Λ *is bounded by*

$$\#(\Lambda) \le C\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)}\eta^{-1/s}, \quad (6.17)$$

and the number of entries of $\mathbf{A}$ *that need to be computed is* $\le C\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)}\eta^{-1/s}$.

(*ii*) *The number of* **arithmetic operations** *needed to compute* $\mathbf{w}(\mathbf{v}, \eta)$ *does not exceed* $C\eta^{-1/s}\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} + 2N$ *with* $N := \#\operatorname{supp}\mathbf{v}$.

(*iii*) *The number of* **sorts** *needed to assemble the* $\mathbf{v}_{[j]}$, $j = 0, 1, \cdots, \lfloor \log N \rfloor$, *of* $\mathbf{w}(\mathbf{v}, \eta)$ *does not exceed* $CN \log N$.

(*iv*) *The output vector* $\mathbf{w}$ *satisfies*

$$\|\mathbf{w}\|_{\ell^w_\tau(\nabla)} \leq C\|\mathbf{v}\|_{\ell^w_\tau(\nabla)}. \tag{6.18}$$

Proof: The estimate (6.16) follows from the preceding remarks centering upon (6.15). Properties (i)-(iii) follow from the results of § 3.3 (see Corollary 3.10). Property (iv) is proved in the same way that we have proved Proposition 3.8. Namely, for $j = 0, 1, \cdots, k$, we prove that $\|\mathbf{w}_k - \mathbf{w}_j\|_{\ell_2(\nabla)} \leq C2^{-js}\|\mathbf{v}\|_{\ell^w_\tau(\nabla)}$ as in (3.25). This then proves (6.18) because of Proposition 3.2. □

6.5 The Numerical Computation of Residuals

Recall that **Algorithm II** heavily utilizes knowledge of residuals. We suppose that Λ is any given finite subset of ∇, and we denote as usual by $\mathbf{u}_\Lambda$ the Galerkin solution associated to the set Λ. Since, we cannot compute $\mathbf{u}_\Lambda$ nor its residual $\mathbf{A}\mathbf{u}_\Lambda - \mathbf{f}$ exactly, we shall introduce numerical algorithms which begins with an approximation $\mathbf{v}$ to $\mathbf{u}_\Lambda$ and approximately computes the residual $\mathbf{A}\mathbf{v} - \mathbf{f}$. For this computation, we introduce the following procedure, which involves two tolerance parameters η_1, η_2 reflecting the desired accuracy of the computation of $\mathbf{A}\mathbf{v}$ and of $\mathbf{f}$, rspectively.

NRESIDUAL$[\mathbf{v}, \Lambda, \mathbf{f}, \eta_1, \eta_2] \to (\mathbf{r}, \tilde{\Lambda})$:

(i) **APPLYA** $[\mathbf{v}, \eta_1] \to (\mathbf{w}, \Lambda_1)$.

(ii) **NCOARSE** $[\mathbf{f}, \eta_2] \to (\mathbf{g}, \Lambda_2)$.

(iii) *Set* $\mathbf{r} := \mathbf{w} - \mathbf{g}$ *and* $\tilde{\Lambda} := \text{supp } \mathbf{r} \subseteq \Lambda_1 \cup \Lambda_2$.

Note that, due to the various approximations, the output $\mathbf{r}$ is not necessarily supported in $\nabla \setminus \Lambda$, in contrast to the exact residual $\mathbf{r}_\Lambda = \mathbf{f} - \mathbf{A}\mathbf{u}_\Lambda$.

Properties 6.5 *The output* $\mathbf{r}$ *of* **NRESIDUAL** *satisfies*

$$\|\mathbf{r} - \mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \leq \eta_1 + \eta_2 + c_2\|\mathbf{v} - \mathbf{u}_\Lambda\|_{\ell_2(\nabla)}. \tag{6.19}$$

Furthermore, if $\mathbf{u} \in \ell^w_\tau(\nabla)$, *with* $\tau = (s+1/2)^{-1/2}$ *and* $0 < s < s^*$ *(which in particular implies* $\mathbf{f} \in \ell^w_\tau(\nabla)$, *see Proposition 3.8), then the following holds:*

(i) *The support size of the output is bounded by*

$$\#(\tilde{\Lambda}) \leq \#(\Lambda_1) + \#(\Lambda_2) \leq C(\eta_1^{-1/s}\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} + \eta_2^{-1/s}\|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}). \tag{6.20}$$

(ii) *The number of* **arithmetic operations** *used in* **NRESIDUAL** *does not exceed* $C\left(\eta_1^{-1/s}\|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^{1/s} + \eta_2^{-1/s}\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}^{1/s}\right) + 2N$ *with* $N := \#(\Lambda)$

(iii) *The number of* **sorts** *needed in the computation of* $\mathbf{r}$ *does not exceed* $CN\log N$.

(iv) *The output* $\mathbf{r}$ *satisfies*

$$\|\mathbf{r}\|_{\ell_\tau^w(\nabla)} \leq C(\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}, +\|\mathbf{v}\|_{\ell_\tau^w(\nabla)}). \tag{6.21}$$

Proof: The estimate (6.19) follows from

$$\|\mathbf{r} - \mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \leq \|\mathbf{f} - \mathbf{g}\|_{\ell_2(\nabla)} + \|\mathbf{A}\mathbf{v} - \mathbf{w}\|_{\ell_2(\nabla)} + \|\mathbf{A}(\mathbf{v} - \mathbf{u}_\Lambda)\|_{\ell_2(\nabla)},$$

and (2.22). All other properties are direct consequences of the Properties 6.2 and 6.4 of **NCOARSE** and **APPLY A**. □

6.6 A Sparse Galerkin Solver

This subsection will be concerned with the computation of a numerical approximation $\bar{\mathbf{u}}_\Lambda$ of $\mathbf{u}_\Lambda$ for any given set $\Lambda \subset \nabla$. We shall discuss this issue in the context of gradient methods. A similar discussion applies to conjugate gradient methods. Given a set Λ, we thus wish to solve

$$\mathbf{P}_\Lambda \mathbf{A}\mathbf{u}_\Lambda = \mathbf{P}_\Lambda \mathbf{f}. \tag{6.22}$$

Suppose that we are provided with a current known approximation $\mathbf{v}$ to $\mathbf{u}_\Lambda$ with $\mathbf{v}$ supported on Λ, and that we want to produce an approximation $\bar{\mathbf{u}}_\Lambda$, supported on Λ, such that $\|\mathbf{u}_\Lambda - \bar{\mathbf{u}}_\Lambda\|_{\ell_2} \leq \eta$ for some prescribed tolerance η.

The gradient method (or damped Richardson iteration) takes as the next approximation

$$\mathbf{v}' := \mathbf{v} - \alpha_\Lambda(\mathbf{A}_\Lambda \mathbf{v} - \mathbf{P}_\Lambda \mathbf{f}) \tag{6.23}$$

where α_Λ is to be chosen. Then, $\mathbf{v}'$ is also supported on Λ and using (6.22), we have

$$\|\mathbf{u}_\Lambda - \mathbf{v}'\|_{\ell_2(\nabla)} \leq \theta_\Lambda \|\mathbf{u}_\Lambda - \mathbf{v}\|_{\ell_2(\nabla)}. \tag{6.24}$$

where

$$\theta_\Lambda := \|\mathbf{P}_\Lambda(\mathbf{I} - \alpha_\Lambda \mathbf{A})\| \tag{6.25}$$

with $\mathbf{I}$ the identity matrix.

To turn this into a numerical algorithm, we need to provide: (i) a value for α_Λ, (ii) an approximation for $\mathbf{A}_\Lambda \mathbf{v} - \mathbf{P}_\Lambda \mathbf{f}$. We shall take

$$\alpha_\Lambda := \alpha := \frac{1}{c_2}, \tag{6.26}$$

where c_2 is our bound for $\|\mathbf{A}\|$ given in (2.22). With this choice, it follows that

$$\theta_\Lambda \leq 1 - \frac{1}{2\kappa}, \tag{6.27}$$

with $\kappa = c_2/c_1$ the estimated condition number.

We next discuss the computation of $\mathbf{A}_\Lambda \mathbf{v} - \mathbf{P}_\Lambda \mathbf{f}$ which we call the "internal residual". In contrast to the full residual $\mathbf{A}\mathbf{v} - \mathbf{f}$ of the full equation, the internal residualcan be computed exactly at finite cost. However, this cost remains too large for the purpose of obtaining a computationally optimal algorithm, so that in practice, we shall need to replace the internal residual by a numerical approximation $\mathbf{r}$. We next examine the properties we shall want for the numerical approximation $\mathbf{r}$ in order that that the modified iterations still converge. Suppose for a moment that our initial approximation $\mathbf{v}$ satisfies

$$\|\mathbf{u}_\Lambda - \mathbf{v}\|_{\ell_2(\nabla)} \leq \delta \tag{6.28}$$

for some $\delta > 0$. We shall show in a moment how to compute an $\mathbf{r}$ such that

$$\|\mathbf{r} - (\mathbf{A}_\Lambda \mathbf{v} - \mathbf{P}_\Lambda \mathbf{f})\|_{\ell_2(\nabla)} \leq \frac{c_1 \delta}{3}. \tag{6.29}$$

Given such an $\mathbf{r}$, we define

$$\bar{\mathbf{v}}' := \mathbf{v} - \alpha \mathbf{r}. \tag{6.30}$$

Since by (6.23), (6.26) and (6.29), $\|\mathbf{v}' - \bar{\mathbf{v}}'\|_{\ell_2(\nabla)} = \alpha \|\mathbf{r} - (\mathbf{A}_\Lambda \mathbf{v} - \mathbf{P}_\Lambda \mathbf{f})\|_{\ell_2(\nabla)} \leq \frac{\delta}{3\kappa}$ we conclude that

$$\|\mathbf{u}_\Lambda - \bar{\mathbf{v}}'\|_{\ell_2(\nabla)} \leq \|\mathbf{u}_\Lambda - \mathbf{v}'\|_{\ell_2(\nabla)} + \|\mathbf{v}' - \bar{\mathbf{v}}'\|_{\ell_2(\nabla)} \leq (1 - \frac{1}{2\kappa})\delta + \frac{1}{3\kappa}\delta = \bar{\theta}\delta \tag{6.31}$$

with

$$\bar{\theta} := 1 - \frac{1}{6\kappa}. \tag{6.32}$$

The vector $\bar{\mathbf{v}}'$ is our numerical computation of one step of the gradient algorithm with a given initial approximation $\mathbf{v}$ and error estimate δ. Notice that (6.31) gives an error estimate which allows us to iterate this algorithm. For example, at the next iteration, we would replace $\mathbf{v}$ by $\bar{\mathbf{v}}'$, and δ by $\bar{\theta}\delta$.

We next discuss how we shall compute an approximation $\mathbf{r}$ to the internal residual which will satisfy (6.29). For this, we shall use a variant of the routine **NRESIDUAL** from §6.5, in which we shall confine all vectors to be supported in Λ. We shall denote this new subroutine by **INRESIDUAL**. It is obtained by replacing $\mathbf{f}$ by $P_\Lambda \mathbf{f}$ in the **NCOARSE** step and $\mathbf{A}$ by $\mathbf{A}_\Lambda$ in the **APPLY A** step.

INRESIDUAL $[\mathbf{v}, \Lambda, \mathbf{f}, \eta_1, \eta_2] \to \mathbf{r}$

(i) APPLY $\mathbf{A}_\Lambda$ $[\mathbf{v}, \eta_1] \to \mathbf{w}$*;*

(ii) NCOARSE $[\mathbf{P}_\Lambda \mathbf{f}, \eta_2] \to \mathbf{g}$.

(iii) *Set* $\mathbf{r} := \mathbf{g} - \mathbf{w}$.

Here **APPLY** $\mathbf{A}_\Lambda$ means that $\mathbf{A}$ is replaced by $\mathbf{A}_\Lambda$ in the fast matrix vector multiplication. From Properties 6.2 and 6.4 we know that the output $\mathbf{r}$ of $[\mathbf{v}, \Lambda, \mathbf{f}, \eta_1, \eta_2]$ satisfies

$$\|\mathbf{r} - (\mathbf{A}_\Lambda \mathbf{v} - \mathbf{P}_\Lambda \mathbf{f})\|_{\ell_2(\nabla)} \leq \eta_1 + \eta_2 \tag{6.33}$$

Thus the choice

$$\eta_1 = \eta_2 = \frac{c_1 \delta}{6} \tag{6.34}$$

suffices to ensure the validity of (6.29).

Obviously the number of iterations needed to guarantee a target accuracy η of the approximate Galerkin solution depends on the error bound δ of the initial approximation $\mathbf{v}$ of $\mathbf{u}_\Lambda$. In fact, the number K of iterations necessary to reach this accuracy is bounded by

$$K \leq K(\delta, \eta) := \left\lceil \left| \log \frac{\eta}{\delta} \right| / \left| \log \bar{\theta} \right| \right\rceil + 1. \tag{6.35}$$

While the above analysis gives an upper bound for the number of iterations we shall need to achieve our target accuracy, it will also be important for our analysis to note that this target accuracy may be reached before this number of iterations if the currently computed approximaton $\mathbf{r}$ to the internal residual is small enough. The following remark (which follows from (6.33)) makes this statement more precise.

Remark 6.6 *If we choose* $\eta_1 = \eta_2 := c_1\eta/6$, *where* η *is the target accuracy, and if* $\mathbf{r}$ *is the corresponding output of* **INRESIDUAL** $[\mathbf{v}, \Lambda, \mathbf{f}, \eta_1, \eta_2]$, *we then have*

$$\|\mathbf{u}_\Lambda - \mathbf{v}\|_{\ell_2(\nabla)} \leq c_1^{-1} \|\mathbf{r}\|_{\ell_2(\nabla)} + \eta/3, \tag{6.36}$$

so that

$$\|\mathbf{u}_\Lambda - \mathbf{v}\|_{\ell_2(\nabla)} \leq \eta \quad \textit{if} \quad \|\mathbf{r}\|_{\ell_2(\nabla)} \leq 2c_1\eta/3. \tag{6.37}$$

Note that conversely, since we also have by (6.33)

$$\|\mathbf{r}\|_{\ell_2(\nabla)} \leq c_1\eta/3 + \|\mathbf{A}_\Lambda \mathbf{v} - P_\Lambda \mathbf{f}\|_{\ell_2(\nabla)},$$

we are ensured that

$$\|\mathbf{r}\|_{\ell_2(\nabla)} \leq 2c_1\eta/3 \quad \text{if} \quad \|\mathbf{u}_\Lambda - \mathbf{v}\|_{\ell_2(\nabla)} \leq c_1 c_2^{-1} \eta/3 = \eta/3\kappa, \tag{6.38}$$

i.e., the criterion will be met when the exact internal residual is small enough.

Proof: To prove (6.36), we write

$$\mathbf{A}_\Lambda(\mathbf{u}_\Lambda - \mathbf{v}) = (\mathbf{A}_\Lambda \mathbf{u}_\Lambda - P_\Lambda \mathbf{f}) - (\mathbf{A}_\Lambda \mathbf{v} - P_\Lambda \mathbf{f}) = \mathbf{A}_\Lambda \mathbf{v} - P_\Lambda \mathbf{f} - \mathbf{r} + \mathbf{r}.$$

Since (2.22) and (2.25) imply $\|\mathbf{v} - \mathbf{u}_\Lambda\|_{\ell_2(\nabla)} \leq c_1^{-1} \|\mathbf{A}_\Lambda(\mathbf{v} - \mathbf{u}_\Lambda)\|_{\ell_2(\nabla)}$, (6.36) follows. Clearly, (6.36) implies (6.37). The rest of the claim follows from (2.22). □

After these considereations, we are now in a position to give our numerical algorithm for computing Galerkin approximations. Given a set Λ, an initial approximation $\mathbf{v}$ to $\mathbf{u}_\Lambda$, an estimate $\|\mathbf{v} - \mathbf{u}_\Lambda\|_{\ell_2(\nabla)} \leq \delta$ and a target accuracy η, with $0 < \eta < \delta$, the approximate Galerkin solver is defined by the following:

GALERKIN $[\Lambda, \mathbf{v}, \delta, \eta] \to \bar{\mathbf{u}}_\Lambda$:

(i) *Apply* **INRESIDUAL** $[\mathbf{v}, \Lambda, \mathbf{f}, \frac{c_1\eta}{6}, \frac{c_1\eta}{6}] \to \mathbf{r}$. *If* $\min\left\{\bar{\theta}\delta, c_1^{-1}\|\mathbf{r}\|_{\ell_2(\nabla)} + \eta/3\right\} \leq \eta$, *define the output* $\bar{\mathbf{u}}_\Lambda$ *to be* $\mathbf{v}$ *and STOP, else go to* (ii).

(ii) *Set*

$$\bar{\mathbf{v}}' := \mathbf{v} - \alpha\mathbf{r}.$$

Since $\eta < \delta$, *we know that* $\|\mathbf{u} - \bar{\mathbf{v}}'\|_{\ell_2(\nabla)} \leq \bar{\theta}\|\mathbf{u} - \mathbf{v}\|_{\ell_2(\nabla)}$. *Replace* $\mathbf{v}$ *by* $\bar{\mathbf{v}}'$, δ *by* $\bar{\theta}\delta$ *and go to* (i).

The relevant properties of **GALERKIN** can be summarized as follows.

Properties 6.7 *Given as input a set* Λ, *an initial approximation* $\mathbf{v}$ *to the exact Galerkin solution* $\mathbf{u}_\Lambda$ *which is supported on* Λ, *an initial error estimate* δ *for* $\|\mathbf{u}_\Lambda - \mathbf{v}\|_{\ell_2(\nabla)}$ *and a target accuracy* η, *the routine* **GALERKIN** *produces an approximation* $\bar{\mathbf{u}}_\Lambda$ *to* $\mathbf{u}_\Lambda$ *which is supported on* Λ *and satisfies*

$$\|\mathbf{u}_\Lambda - \bar{\mathbf{u}}_\Lambda\|_{\ell_2(\nabla)} \leq \eta. \tag{6.39}$$

Moreover, if K *is the number of modified gradient iterations which have been used in* **GALERKIN** *to produce* $\bar{\mathbf{u}}_\Lambda$, *one also has*

$$\|\mathbf{u}_\Lambda - \bar{\mathbf{u}}_\Lambda\|_{\ell_2(\nabla)} \leq \bar{\theta}^K\delta \tag{6.40}$$

with $\bar{\theta}$ *defined by* (6.32). *Consequently, the number of iterations* K *is always bounded by*

$$K \leq K(\delta, \eta) = \left|\log\frac{\eta}{\delta}\right| / \left|\log\bar{\theta}\right|. \tag{6.41}$$

Moreover, if $\mathbf{u} \in \ell_\tau^w(\nabla)$, *with* $\tau = (s + 1/2)^{-1}$ *and* $0 < s < s^*$, *then the following are true:*

(i) *The output* $\bar{\mathbf{u}}_\Lambda$ *of* **GALERKIN** $[\Lambda, \mathbf{v}, \delta, \eta]$ *satisfies*

$$\|\bar{\mathbf{u}}_\Lambda\|_{\ell_\tau^w(\nabla)} \leq C(K)\left(\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} + \|\mathbf{u}\|_{\ell_\tau^w(\nabla)}\right), \tag{6.42}$$

where the constant $C(K)$ *depends only on the number of iterations* K.

(ii) *The number of* **arithmetic operations** *used in* **GALERKIN** $[\Lambda, \mathbf{v}, \delta, \eta]$ *is less than*

$$\tilde{C}(K)\left(\|\mathbf{v}\|_{\ell_\tau^w(\nabla)}^{1/s} + \|\mathbf{u}\|_{\ell_\tau^w(\nabla)}^{1/s}\right)\eta^{-1/s} + CK(\#\Lambda),$$

where the constant $\tilde{C}(K)$ *depends only on the number of iterations* K. *The number of* **sorts** *does not exceed* $K(\#\Lambda)\log(\#\Lambda)$.

Proof: The first part of the assertion has been already established in the course of the preceding discussion. In particular, the bound on the maximal number K of iterations clearly follows from (6.35).

As for property (i), we simply remark that (iv) in Properties (6.5) of **NRESIDUAL** also applies in the case of the modified procedure **INRESIDUAL**, so that after one modified gradient iteration we have

$$\|\bar{\mathbf{v}}'\|_{\ell^w_\tau(\nabla)} \leq C \max\left\{\|\mathbf{v}\|_{\ell^w_\tau(\nabla)}, \|\mathbf{u}\|_{\ell^w_\tau(\nabla)}\right\}.$$

The assertion (i) follows therefore by iterating this argument: denoting by $\mathbf{v}^k$ the current approximation after k iterations, we obtain that $\|\mathbf{v}^k\|_{\ell^w_\tau(\nabla)} \leq C(k)(\|\mathbf{v}\|_{\ell^w_\tau(\nabla)} + \|\mathbf{u}\|_{\ell^w_\tau(\nabla)})$.

To estimate the number of arithmetic operations in this algorithm, we can use the bound on the number of operations for **NRESIDUAL** ((ii) in Properties (6.5)), which also applies to **INRESIDUAL**. According to this property, at the k-th iteration, the application of **INRESIDUAL** to $\mathbf{v}^k$ requires at most $C\left(\|\mathbf{v}^k\|^{1/s}_{\ell^w_\tau(\nabla)} + \|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}\right)\eta^{-1/s} + 2(\#\Lambda)$ arithmetic operations. We add each of these estimates for operation count over $k = 0, 1, \ldots, K$ and use the estimate on $\|\mathbf{v}^k\|_{\ell^w_\tau(\nabla)}$ to obtain the estimate in (ii).

Finally, at each iteration, the number of sorts is clearly bounded by $\#\Lambda \log(\#\Lambda)$, which implies the bound in $K\#\Lambda\log(\#\Lambda)$ for the global procedure. □

The possible growth of the constants $C(K)$ in (6.42) shows the importance of controlling the number of iteration K. The estimate (6.41) expresses that this is feasible if the initial accuracy bound δ is within a fixed factor of the desired target accuracy η in each application of **GALERKIN**. In the setting of **Algorithm III** below this will indeed be the case.

7 Numerical Realization: The Adaptive Algorithm

We now have collected all the ingredients that are needed to construct an optimal adaptive algorithm, both in terms of memory size and computational cost. The purpose of this section is to describe this algorithm and to prove its optimality.

7.1 General principles of the Algorithm

Recall from §6.1 that we start with an estimate $\|\mathbf{f}\|_{\ell_2(\nabla)} \leq F$. Introducing the sequence of tolerances

$$\epsilon_j := 2^{-j} F c_1^{-1}. \tag{7.1}$$

we see that $\Lambda_0 := \emptyset$ and $\bar{\mathbf{u}}_{\Lambda_0} = \mathbf{0}$ are an admissible initialization in the sense that $\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda_0}\|_{\ell_2(\nabla)} \leq \epsilon_0$.

Algorithm III conceptually parallels the idealized version **Algorithm II**. Its core ingredient is a routine called **NPROG** that associates to a triplet $(\bar{\mathbf{u}}_\Lambda, \Lambda, \delta)$ such that $\bar{\mathbf{u}}_\Lambda$ is supported in Λ and $\|\bar{\mathbf{u}}_\Lambda - \mathbf{u}\|_{\ell_2(\nabla)} \leq \delta$, a new pair $(\bar{\mathbf{u}}_{\tilde{\Lambda}}, \tilde{\Lambda})$ such that $\bar{\mathbf{u}}_{\tilde{\Lambda}}$ is supported in $\tilde{\Lambda}$ and $\|\bar{\mathbf{u}}_{\tilde{\Lambda}} - \mathbf{u}\|_{\ell_2(\nabla)} \leq \delta/2$.

Iterating this procedure thus builds a sequence $(\bar{\mathbf{u}}_{\Lambda_j}, \Lambda_j)_{j\geq 0}$ with $\bar{\mathbf{u}}_{\Lambda_j}$ supported in Λ_j such that

$$\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda_j}\|_{\ell_2(\nabla)} \leq \epsilon_j. \tag{7.2}$$

If ϵ is the target accuracy, the algorithm thus stops after J steps where J is the smallest integer such that $\epsilon_J \leq \epsilon$.

As in **Algorithm II** the routine **NPROG** itself will consist of possibly several applications of a procedure **NGROW** described below, which parallels **GROW** in **Algorithm II**, followed by **NCOARSE** for exactly the same reasons that came up in §5.

In contrast to **Algorithm II**, the selection of the next larger index set done by **NGROW** will have to be based on an approximate residual obtained by **NRESIDUAL** rather than on the exact one. We shall also use the approximate Galerkin solver defined by **NGALERKIN** to derive the intermediate approximations of the solution after each growing steps. Thus, the error reduction in this growing procedure requires a more refined analysis, involving the various tolerances in these procedures. We shall first address this analysis which will result in several constraints on the tolerance parameters.

7.2 The growing procedure

At the start of the growing procedure that will define **NPROG**, we are given set Λ, an approximate solution $\bar{\mathbf{u}}_\Lambda$ supported on Λ and a known estimate $\|\mathbf{u} - \bar{\mathbf{u}}_\Lambda\|_{\ell_2(\nabla)} \leq \delta$.

We set $\Lambda^0 := \Lambda$ and $\bar{\mathbf{u}}_{\Lambda^0} := \bar{\mathbf{u}}_\Lambda$. The growing procedure will build iteratively some larger sets Λ^k, $k = 0, 1, \cdots$, and approximate solutions $\bar{\mathbf{u}}_{\Lambda^k}$, and will be stopped at some K such that we are ensured that

$$\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^K}\|_{\ell_2(\nabla)} \leq \delta/10, \tag{7.3}$$

so that applying **NCOARSE** $[\bar{\mathbf{u}}_{\Lambda^K}, 2\delta/5]$ will output the new set $\tilde{\Lambda}$ and approximate solution $\bar{\mathbf{u}}_{\tilde{\Lambda}}$ such that $\|\mathbf{u} - \bar{\mathbf{u}}_{\tilde{\Lambda}}\|_{\ell_2(\nabla)} \leq \delta/2$. The choice $\delta/10$ in (7.3) is justified by the Properties 6.3 of the thresholding procedure: it ensures the optimality of the approximate solution and the control of its $\ell^w_\tau(\nabla)$ norm (see (i) and (ii) in Properties 6.3).

As in **Algorithm II**, the growing procedure will ensure a geometric reduction of the error in the energy norm $\|\mathbf{u} - \mathbf{u}_{\Lambda^k}\|$ where $\mathbf{u}_{\Lambda^k}$ is the exact Galerkin solution. Although it will not ensure such a reduction for $\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^K}\|_{\ell_2(\nabla)}$, we shall still reach (7.3) after a controlled number of steps.

The procedure **NGROW** generating the sets Λ^k can be described as follows: given a set Λ and an approximation $\bar{\mathbf{u}}_\Lambda$ supported on Λ, we compute an approximate residual $\mathbf{r}$ and select the new set $\tilde{\Lambda} \supset \Lambda$ as small as possible such that

$$\|\mathbf{P}_{\tilde{\Lambda}/\Lambda}\mathbf{r}\|_{\ell_2(\nabla)} \geq \gamma \|\mathbf{r}\|_{\ell_2(\nabla)}, \tag{7.4}$$

for some fixed γ in $(0, 1]$. This can be done by taking $\tilde{\Lambda} := \Lambda \cup \Lambda^c$ where

$$(\Lambda^c, \mathbf{P}_{\Lambda^c}\mathbf{r}) = \mathbf{NCOARSE}\,[\mathbf{r}, \sqrt{1-\gamma^2}\|\mathbf{r}\|_{\ell_2(\nabla)}].$$

This procedure can thus be summarized as follows.

NGROW $[\Lambda, \bar{\mathbf{u}}_\Lambda, \xi_1, \xi_2, \mathbf{f}, \gamma] \to (\tilde{\Lambda}, \mathbf{r})$
Given an initial approximation $\bar{\mathbf{u}}_\Lambda$ *to the Galerkin solution* $\mathbf{u}_\Lambda$ *supported on* Λ *the procedure* **NGROW** *consists of the following steps:*

(i) *Apply* **NRESIDUAL** $[\bar{\mathbf{u}}_\Lambda, \Lambda, \mathbf{f}, \xi_1, \xi_2] \to (\Lambda^r, \mathbf{r})$.

(ii) *Apply* **NCOARSE** $[\mathbf{r}, \sqrt{1-\gamma^2}\|\mathbf{r}\|_{\ell_2(\nabla)}] \to (\Lambda^c, \mathbf{P}_{\Lambda^c}\mathbf{r})$ *and define* $\tilde{\Lambda} := \Lambda \cup \Lambda^c$.

It is interesting to note that we allow the situation where $\gamma = 1$, in which case we simply have $\tilde{\Lambda} = \Lambda \cup \Lambda^r$. This was not possible with **GROW** in **Algorithm II** since $\tilde{\Lambda}$ could then be the full infinite set ∇.

Properties 7.1 *The residual computed by* **NGROW** *satisfies the estimate*

$$\|\mathbf{r} - \mathbf{r}_\Lambda\|_{\ell_2(\nabla)} \le \xi_1 + \xi_2 + c_2\|\bar{\mathbf{u}}_\Lambda - \mathbf{u}_\Lambda\|_{\ell_2(\nabla)}. \tag{7.5}$$

If $\mathbf{u} \in \ell^w_\tau(\nabla)$, *with* $\tau = (s + 1/2)^{-1}$ *and* $0 < s < s^*$, *then the following are true:*

(i) *The cardinality of the output* $\tilde{\Lambda}$ *of* **NGROW** *can be bounded by*

$$\#(\tilde{\Lambda}) \le \#(\Lambda) + C\xi^{-1/s}\left(\|\bar{\mathbf{u}}_\Lambda\|^{1/s}_{\ell^w_\tau(\nabla)} + \|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}\right), \tag{7.6}$$

where $\xi := \min\{\xi_1, \xi_2\}$.

(ii) *The number of* **arithmetic operations** *used in* **NGROW** *is less than*

$$M(\xi) := C\left(\xi^{-1/s}(\|\bar{\mathbf{u}}_\Lambda\|^{1/s}_{\ell^w_\tau(\nabla)} + \|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}) + \#(\Lambda)\right). \tag{7.7}$$

(iii) *The number of* **sorts** *does not excede* $CM(\xi)\log M(\xi)$.

Proof: The first part of the assertion follows from (6.19). The claims (i), (ii) and (iii) follow from (i), (ii) and (iii) in the Properties 6.5 of **NRESIDUAL**. □

In our growing procedure, the tolerance parameters ξ_1 and ξ_2 will be related to the initial accuracy δ by $\xi_1 = q_1\delta$ and $\xi_2 = q_2\delta$ where q_1 and q_2 are fixed parameters that we shall specify below through our analysis. Similarly, we shall always set the tolerance parameter in the applications of **NGALERKIN** in such a way that the approximate solutions $\bar{\mathbf{u}}_{\Lambda^k}$ will always satisfy

$$\|\mathbf{u}_{\Lambda^k} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} \le q_3\delta/c_2, \tag{7.8}$$

where q_3 is another parameter to be specified later and $\mathbf{u}_{\Lambda^k}$ is the exact Galerkin solution.

Note that (7.8) is not ensured for $k = 0$, so that the very first step of our growing procedure should be to replace $\bar{\mathbf{u}}_{\Lambda^0}$ by the output of **NGALERKIN** $[\Lambda^0, \bar{\mathbf{u}}_{\Lambda^0}, \delta, q_3\delta/c_2]$.

The growing procedure will then proceed as follows: for $k > 0$, we shall define Λ^k as the first output of $\mathbf{NGROW}\,[\Lambda^{k-1}, \bar{\mathbf{u}}_{\Lambda^{k-1}}, q_1\delta, q_2\delta, \mathbf{f}, \gamma]$. We then define $\bar{\mathbf{u}}_{\Lambda^k}$ as the output of $\mathbf{NGALERKIN}\,[\Lambda^k, \bar{\mathbf{u}}_{\Lambda^{k-1}}, q_0\delta, q_3\delta/c_2]$, with the constant q_0 still to be specified. It follows that (7.8) will automatically be satisfied by (6.39).

Regarding the parameter q_0, we need to choose its value so that at each iteration we have

$$\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} \leq q_0\delta \tag{7.9}$$

because we are using $\bar{\mathbf{u}}_{\Lambda_k}$ as the input for the next application of **NGALERKIN**. Now, for each $k > 0$, we have

$$\begin{aligned}
\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} &\leq \|\mathbf{u} - \mathbf{u}_{\Lambda^k}\|_{\ell_2(\nabla)} + \|\mathbf{u}_{\Lambda^k} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} \\
&\leq c_1^{-1/2}\|\mathbf{u} - \mathbf{u}_{\Lambda^k}\| + q_3\delta/c_2 \\
&\leq c_1^{-1/2}\|\mathbf{u} - \mathbf{u}_{\Lambda^0}\| + q_3\delta/c_2 \\
&\leq \kappa^{1/2}\|\mathbf{u} - \mathbf{u}_{\Lambda^0}\|_{\ell_2(\nabla)} + q_3\delta/c_2 \\
&\leq (\kappa^{1/2} + q_3/c_2)\delta,
\end{aligned}$$

where we have used the monotonicity of the error $\|\mathbf{u} - \mathbf{u}_{\Lambda^k}\|$ as the sets Λ^k are growing. Hence, we see that we can take $q_0 := \kappa^{1/2} + q_3/c_2$. With this choice of q_0 and with any fixed choice of q_3, (6.41) the Properties 6.7 shows that the number of iterations within each application of **NGALERKIN** is uniformly bounded independently of k and δ.

Note also that in terms of the parameters q_1, q_2, q_3, from (7.5) and (7.8) we deduce

$$\|\mathbf{r}^k - \mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} \leq (q_1 + q_2 + q_3)\delta, \tag{7.10}$$

where $\mathbf{r}^k$ is the second output of $\mathbf{NGROW}\,[\Lambda^k, \bar{\mathbf{u}}_{\Lambda^k}, q_1\delta, q_2\delta, \mathbf{f}, \gamma]$.

In order to analyze the error reduction in our growing procedure, we shall need to relate the property (7.4) that defines **NGROW** with the property (4.8) which is known to ensure a fixed reduction of the error $\|\mathbf{u} - \mathbf{u}_\Lambda\|$. Using our error estimate (7.10) we obtain

$$\begin{aligned}
\|\mathbf{P}_{\Lambda^{k+1}}\mathbf{r}_{\Lambda_k}\|_{\ell_2(\nabla)} &\geq \|\mathbf{P}_{\Lambda^{k+1}}\mathbf{r}^k\|_{\ell_2(\nabla)} - \|\mathbf{P}_{\Lambda^{k+1}}(\mathbf{r}^k - \mathbf{r}_{\Lambda^k})\|_{\ell_2(\nabla)} \\
&\geq \gamma\|\mathbf{r}^k\|_{\ell_2(\nabla)} - \|\mathbf{r}^k - \mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} \\
&\geq \gamma\|\mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} - (1+\gamma)\|\mathbf{r}^k - \mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} \\
&\geq \gamma\|\mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} - (1+\gamma)(q_1 + q_2 + q_3)\delta.
\end{aligned} \tag{7.11}$$

Of course, we wish to ensure that our above choice of the expanded set Λ^{k+1} which was based on the *approximate* residual $\mathbf{r}^k$ does capture also a sufficient bulk of the *true* residual $\mathbf{r}_{\Lambda^k}$. This can be indeed inferred from the above estimate provided that the perturbation on the right hand side is small compared with the first summand. If this is not the case the choice of the parameters q_i should ensure that the residual itself and hence the error is already small enough. The following observation describes this in more detail.

Remark 7.2 *Given any $q_4 > 0$, suppose that the parameters q_1, q_2, q_3 are chosen small enough that*

$$\left(\frac{q_3}{c_2} + \frac{2(1+\gamma)(q_1+q_2+q_3)}{\gamma c_1}\right) \leq q_4. \tag{7.12}$$

Then, for Λ^{k+1} constructed from $\bar{\mathbf{u}}_{\Lambda^k}$ as explained above, one either has

$$\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} \leq q_4\delta, \tag{7.13}$$

or

$$\|\mathbf{u} - \mathbf{u}_{\Lambda^{k+1}}\| \leq \theta\|\mathbf{u} - \mathbf{u}_{\Lambda^k}\|, \tag{7.14}$$

where

$$\theta := \sqrt{1 - \frac{c_1}{4c_2}\gamma^2}. \tag{7.15}$$

Proof: let $q := (1+\gamma)(q_1+q_2+q_3)$. We distinguish two cases. If $\gamma\|\mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} \leq 2q\delta$, then by (2.22) we have $\gamma c_1\|\mathbf{u} - \mathbf{u}_{\Lambda^k}\|_{\ell_2(\nabla)} \leq 2q\delta$. Combining this with the estimate (7.8), we obtain

$$\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} \leq \left(\frac{q_3}{c_2} + \frac{2q}{\gamma c_1}\right)\delta,$$

which, in view of (7.12) proves (7.13). Alternatively, when $\gamma\|\mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} > 2q\delta$, we infer from (7.11) that

$$\|\mathbf{P}_{\Lambda^{k+1}}\mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)} \geq \frac{\gamma}{2}\|\mathbf{r}_{\Lambda^k}\|_{\ell_2(\nabla)}, \tag{7.16}$$

which is the desired prerequisite for error reduction in the energy norm. In fact, we can invoke Lemma 4.1 to conclude that (7.14) holds for θ defined in (7.15). □

It remains to adjust the various parameters q_1, q_2, q_3. To this end, one should keep in mind that the growing procedure aims to achieve the accuracy in (7.3) after a finite number of steps K.

In view of Remark 7.2, a first natural choice seems to be $q_4 = 1/10$ since the occurrence of case one in Remark 7.2 would then imply (7.3). However, with such a choice, it could still happen that at the i-th stage of the growing procedure, case one comes up but is not discovered by any error control. In this case, we will need to make sure that subsequent steps still satisfy (7.3). For $k > i$, we have

$$\begin{aligned}
\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} &\leq \|\mathbf{u} - \mathbf{u}_{\Lambda^k}\|_{\ell_2(\nabla)} + \|\mathbf{u}_{\Lambda^k} - \bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} \\
&\leq \frac{1}{\sqrt{c_1}}\|\mathbf{u} - \mathbf{u}_{\Lambda^k}\| + q_3\delta/c_2 \leq \frac{1}{\sqrt{c_1}}\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda^i}\| + q_3\delta/c_2 \\
&\leq \left(q_4\sqrt{\kappa} + \frac{q_3}{c_2}\right)\delta,
\end{aligned}$$

where we have again made standard use of (2.21), the best approximation property of Galerkin solutions and (7.8). Thus our first requirement is

$$\left(q_4\sqrt{\kappa} + \frac{q_3}{c_2}\right) \leq 1/10. \tag{7.17}$$

We next have to make sure that if case one never occurs a uniformly bounded finite number of steps suffices to reach (7.3). In fact, we infer from (7.14) that

$$\begin{aligned}\|\mathbf{u}-\bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)} &\leq \|\mathbf{u}-\mathbf{u}_{\Lambda^k}\|_{\ell_2(\nabla)}+\|\mathbf{u}_{\Lambda^k}-\bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)}\\ &\leq \frac{1}{\sqrt{c_1}}\|\mathbf{u}-\mathbf{u}_{\Lambda^k}\|+q_3\delta/c_2 \leq \frac{\theta^k}{\sqrt{c_1}}\|\mathbf{u}-\bar{\mathbf{u}}_{\Lambda^0}\|+q_3\delta/c_2\\ &\leq \left(\theta^k\sqrt{\kappa}+\frac{q_3}{c_2}\right)\delta. \end{aligned} \tag{7.18}$$

Thus our second requirement in order to achieve (7.3) is that for sufficiently large K,

$$\left(\theta^K\sqrt{\kappa}+\frac{q_3}{c_2}\right)\leq 1/10, \tag{7.19}$$

but this is always implied by our first requirement (7.17) for K sufficiently large but fixed.

Finally, we wish to install intermediate error controls to avoid unnecessarily many steps in the above growing procedure. To this end, we write

$$\mathbf{u}-\bar{\mathbf{u}}_{\Lambda_k}=\mathbf{u}-\mathbf{u}_{\Lambda_k}+\mathbf{u}_{\Lambda_k}-\bar{\mathbf{u}}_{\Lambda_k}.$$

and deduce from (7.8) that at any intermediate stage

$$\|\mathbf{u}-\bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)}\leq c_1^{-1}\left(\|\mathbf{r}^k\|_{\ell_2(\nabla)}+\|\mathbf{r}^k-\mathbf{r}_{\Lambda_k}\|_{\ell_2(\nabla)}\right)+\frac{q_3\delta}{c_2}$$

Therefore, using (7.10), we find

$$\|\mathbf{u}-\bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_2(\nabla)}\leq c_1^{-1}\left(\|\mathbf{r}^k\|_{\ell_2(\nabla)}+(q_1+q_2+q_3)\delta\right)+\frac{q_3\delta}{c_2} \tag{7.20}$$

Thus, imposing the requirement

$$(q_1+q_2+q_3+\kappa^{-1}q_3)\leq\frac{c_1}{20}, \tag{7.21}$$

we can stop the iteration if the following test of the *current approximate residual* is answered affirmatively

$$\|\mathbf{r}^k\|_{\ell_2(\nabla)}\leq\frac{c_1\delta}{20}. \tag{7.22}$$

Choice of parameters: *In summary, possible choices for these parameters are limited by* (7.12), (7.17) *and* (7.21). *A simple possibility is to take*

$$q_4:=\frac{1}{20\kappa} \tag{7.23}$$

Then choose $q_1=q_2=q_3$ *such that* (7.12), (7.17) *and* (7.21) *hold. We then define* $q_0:=\kappa^{1/2}+q_3/c_2$.

Thus one finally sees from (7.18) that the maximal number of steps needed to achieve (7.3) is bounded by

$$K:=K(\kappa,\theta):=\left[\frac{\log\ 20\kappa}{|\log\ \theta|}\right]+1. \tag{7.24}$$

7.3 Description of the algorithm

We are now in a position to describe the main step **NPROG** in our algorithm. We fix a value of γ with $0 < \gamma \leq 1$ and we choose parameters q_0, q_1, q_2, q_3, q_4 as in the **Choice of parameters** of the previous subsection. We fix these values. The analysis of the previous subsection shows that a uniformly bounded finite number of applications of **NGROW** suffices to reduce the initial error by the desired amount. The **NPROG** can thus be summarized as follows.

NPROG $[\Lambda, \mathbf{v}, \delta, \mathbf{f}] \to (\hat{\Lambda}, \hat{\mathbf{v}}, \hat{\mathbf{r}})$
Given a set Λ, an approximation $\mathbf{v}$ to the exact solution $\mathbf{u}$ of (4.1) whose support is contained in Λ and such that $\|\mathbf{v} - \mathbf{u}\|_{\ell_2(\nabla)} \leq \delta$, the procedure **NPROG** *consists of the following steps:*

(i) *Apply* **GALERKIN** $[\Lambda, \mathbf{v}, \delta, q_3\delta/c_2] \to \bar{\mathbf{u}}_\Lambda$. *Set $\Lambda^0 := \Lambda$, $\bar{\mathbf{u}}_{\Lambda^0} := \bar{\mathbf{u}}_\Lambda$, $k := 0$.*

(ii) *Apply* **NGROW** $[\Lambda^k, \bar{\mathbf{u}}_{\Lambda^k}, q_1\delta, q_2\delta, \mathbf{f}, \gamma] \to (\Lambda^{k+1}, \mathbf{r}^k)$.

(iii) *If $\|\mathbf{r}^k\|_{\ell_2(\nabla)} \leq c_1\delta/20$ or $k = K$ defined in (7.24) go to* (iv)*, otherwise apply* **GALERKIN** $[\Lambda^{k+1}, \bar{\mathbf{u}}_{\Lambda^k}, q_0\delta, q_3\delta/c_2] \to \bar{\mathbf{u}}_{\Lambda^{k+1}}$. *Replace k by $k+1$, Λ^k by Λ^{k+1}, $\bar{\mathbf{u}}_{\Lambda^k}$ by $\bar{\mathbf{u}}_{\Lambda^{k+1}}$ and go to* (ii).

(iv) *Apply* **NCOARSE** $[\bar{\mathbf{u}}_{\Lambda^k}, 2\delta/5] \to (\hat{\Lambda}, \hat{\mathbf{v}})$, *set $\hat{\mathbf{r}} := \mathbf{r}^k$ and STOP.*

The relevant properties of **NPROG** can be summarized as follows.

Properties 7.3 *The output $\hat{\mathbf{v}}$ of* **NPROG** *satisfies*

$$\|\mathbf{u} - \hat{\mathbf{v}}\|_{\ell_2(\nabla)} \leq \delta/2. \tag{7.25}$$

Moreover, if $\mathbf{u} \in \ell^w_\tau(\nabla)$, with $\tau = (s + 1/2)^{-1}$ and $0 < s < s^*$, then the following are true:

(i) *One has the bound*

$$\|\hat{\mathbf{v}}\|_{\ell^w_\tau(\nabla)} \leq C\|\mathbf{u}\|_{\ell^w_\tau(\nabla)}, \tag{7.26}$$

and the cardinality of $\hat{\Lambda}$ is bounded by

$$\#(\hat{\Lambda}) \leq C\delta^{-1/s}\|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}. \tag{7.27}$$

(ii) *The cardinality of all intermediate sets Λ^k produced by* **NGROW** *can be bounded by*

$$\#(\Lambda) + C\delta^{-1/s}\left(\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} + \|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}\right). \tag{7.28}$$

(iii) *The number of* **arithmetic operations** *used in* **NPROG** $[\Lambda, \mathbf{v}, \delta, \mathbf{f}]$ *is less than*

$$G := C\left(\delta^{-1/s}(\|\mathbf{v}\|^{1/s}_{\ell^w_\tau(\nabla)} + \|\mathbf{u}\|^{1/s}_{\ell^w_\tau(\nabla)}) + \#(\Lambda)\right). \tag{7.29}$$

The number of **sorts** *does not excede $CG \log G$.*

Proof: By our choice of parameters $q_i, i = 1, 2, 3, 4$, Remark 7.2 and the subsequent discussion show that after at most K steps, K given by (7.24), the reduction (7.3) is achieved. The estimate (7.25) is then an immediate consequence of (6.6) and (6.7) in Properties 6.3. Moreover, when $\mathbf{u} \in \ell_\tau^w(\nabla)$, with $\tau = (s + 1/2)^{-1}$, then (i) is a a direct consequence of (ii) and (iii) in Properties 6.3. By a repeated application of (6.42) in Properties 6.7 we conclude that

$$\|\bar{\mathbf{u}}_{\Lambda^k}\|_{\ell_\tau^w(\nabla)} \leq C \left(\|\mathbf{v}\|_{\ell_\tau^w(\nabla)} + \|\mathbf{u}\|_{\ell_\tau^w(\nabla)} \right). \tag{7.30}$$

We have used here that only a uniformly bounded number of applications of **NGROW** and **GALERKIN** is used in **NPROG**. Combining (7.30) with (7.6) in Properties 7.1 yields the estimate (7.28) in (ii).

Note that the same is true for the possibly somewhat larger sets $\Lambda \cup \Lambda^r$ generated in **NGROW**, since we accept the case $\gamma = 1$. The remaining assertion (iii) is also obtained by combining (7.30) with (7.7) in Properties 7.1. □

We are now prepared to describe

Algorithm III

(i) Initialization: *Let $\epsilon > 0$ be the target accuracy. Set $\Lambda := \emptyset$, $\mathbf{v} = \mathbf{0}$ and $\delta := F$, where F is defined at the beginning of this section. Select the parameters q_0, q_1, q_2, q_3, q_4 according to the above* **Choice of Parameters** *and fix these parameters.*

(ii) *If $\delta \leq \epsilon$, accept $\mathbf{u}(\epsilon) := \mathbf{v}$, $\Lambda(\epsilon) := \Lambda$ as the final solution and STOP. Otherwise, apply* **NPROG** $[\Lambda, \mathbf{v}, \delta, \mathbf{f}] \to (\hat{\Lambda}, \hat{\mathbf{v}}, \hat{\mathbf{r}})$.

(iii) *If $\|\hat{\mathbf{r}}\|_{\ell_2(\nabla)} + (q_1 + q_2 + (1 + \kappa^{-1})q_3)\delta \leq c_1\epsilon$ accept $\bar{\mathbf{u}}(\epsilon) := \bar{\mathbf{u}}_{\Lambda^k}$, $\Lambda(\epsilon) = \Lambda^k$ as the solution, where $\bar{\mathbf{u}}_{\Lambda^k}$, $\Lambda(\epsilon) = \Lambda^k$ are the last outputs of* **NGROW** *in* **NPROG** *before thresholding. Otherwise, replace δ by $\delta/2$, $\mathbf{v}$ by $\hat{\mathbf{v}}$ and Λ by $\hat{\Lambda}$ and go to* (ii).

Remark 7.4 *We see that the finest accuracy needed on the data $\mathbf{f}$ is $2q_2\epsilon$ in the last application of* **NPROG** *so that we can start with an estimate $\bar{\mathbf{f}}$ with $\eta = 2q_2\epsilon$ in* (6.2).

Remark 7.5 *The proper choice of $(q_0, q_1, q_2, q_3, q_4)$ is meant to ensure the convergence of* **Algorithm III**, *as well as the control of the operation count in each application of* **NPROG**. *This, in turn, allows us to prove the optimality of this algorithm, as shown below. Roughly speaking, convergence is ensured if these parameters are sufficiently small, but choosing them too small typically increases the constants that enter the optimality analysis (e.g. the number of iterations needed in* **GALERKIN** *or* **APPLY A**), *so that a proper tuning should really be effective in practice. In particular, it might be that our requirements in the* **Choice of Parameters** *are too pessimistic and that the algorithm still works with larger tolerances.*

7.4 The main Result

The convergence properties of **Algorithm III** can be summarized as follows.

Theorem 7.6 *Assume that* $\mathbf{A} \in \mathcal{B}_s$, *with* $0 < s < s^*$, *and that* $\mathbf{A}$ *is an isomorphism on* $\ell_2(\nabla)$ *and suppose that the assumptions* **(N1)**–**(N3)** *are satisfied. Let* $\mathbf{u}$ *be the solution of* (2.17), *that is,* $\mathbf{Au} = \mathbf{f}$. *Then for any* $\epsilon > 0$ *and any* $\mathbf{f} \in \ell_2(\nabla)$, **Algorithm III** *produces an approximation* $\bar{\mathbf{u}} = \bar{\mathbf{u}}(\epsilon)$ *with* $N = N(\epsilon) := \#\mathrm{supp}\,\mathbf{u}(\epsilon) < \infty$ *satisfying*

$$\|\mathbf{u} - \bar{\mathbf{u}}(\epsilon)\|_{\ell_2(\nabla)} \leq \epsilon. \tag{7.31}$$

Moreover, **Algorithm III** *is optimal in the following sense. If* $\mathbf{u} \in \ell_\tau^w(\nabla)$, $\tau = (s+1/2)^{-1}$ *for some* $0 < s < s^*$, *then* $N(\epsilon) \leq C\epsilon^{-1/s}\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}$ *and the computation of* $\bar{\mathbf{u}}(\epsilon)$ *requires at most* $CN(\epsilon)$ *arithmetic operations and at most* $CN(\epsilon)\log N(\epsilon)$ *sorts, where the constants* C *are independent of* $\mathbf{f}$ *and* ϵ.

Proof: Let $\epsilon_j := 2^{-j}F$, $j = 0, 1, \ldots$. Let k be the smallest integer such that $\epsilon_k \leq \epsilon$. The algorithm shuts down when at an iteration j either (i) $\|\mathbf{r}^j\|_{\ell_2(\nabla)} + (q_1 + q_2 + (1+\kappa^{-1}q_3) \leq c_1\epsilon$ or (ii) $\delta \leq \epsilon$. In the first case, (7.31) is satisfied because of (7.20). When case (i) is not met for any $j = 0, \ldots, k$, then (7.25) in Properties 7.3 shows that **Algorithm III** produces a sequence $(\Lambda_j, \bar{\mathbf{u}}_{\Lambda_j})$ such that $\|\mathbf{u} - \bar{\mathbf{u}}_{\Lambda_j}\|_{\ell_2(\nabla)} \leq \epsilon_j$, where $\epsilon_j = 2^{-j}F$. Hence the desired target accuracy ϵ is reached when $j = k$. In either case, the algorithm will need at most k steps to reach (7.31).

As for the complexity analysis, if $\mathbf{u} \in \ell_\tau^w(\nabla)$, $\tau = (s+1/2)^{-1}$ for some $0 < s < s^*$, we conclude from (7.26) that $\|\mathbf{u}_{\Lambda_j}\|_{\ell_\tau^w(\nabla)} \leq C\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}$ for all j and from (7.27) that $\#(\Lambda_j) \leq CG_j$ with $G_j := \epsilon_j^{-1/s}\|\mathbf{u}\|_{\ell_\tau^w(\nabla)}^{1/s}$.

Thus, on account of (ii) and (iii) in Properties 7.3, the number of arithmetic operations and the number of sorts at the jth stage of the algorithm can be bounded respectively by $C\,G_j$ and $C\,\log\,G_j$. The assertion now follows by summing these estimates over $j = 0, \ldots, k$. □

We conclude with briefly summarizing the consequences of the above theorem with regard to the original operator equation (2.5).

Corollary 7.7 *Assume that* $A : H^t \to H^{-t}$ *is an isomorphism and let* u *denote the exact solution of* $Au = f$ *for some* $f \in H^{-t}$. *Suppose that* A *and the wavelet bases* $\Psi, \tilde{\Psi}$ *satisfy assumptions* **(A1)**–**(A3)** *so that, in particular, the preconditioned wavelet representation* $\mathbf{A}$ *of* A *belongs to* $\mathcal{A}_{\sigma,\beta}$. *Let* $s^* := \min\{\frac{\sigma}{d} - \frac{1}{2}, \frac{\beta}{d} - 1\}$. *Then for any* $f \in H^{-t}$ *and every* $\epsilon > 0$, **Algorithm III** *produces a sequence* $\bar{\mathbf{u}}(\epsilon) = \{u_\lambda\}_{\lambda\in\Lambda(\epsilon)}$ *such that*

$$\|u - \sum_{\lambda\in\Lambda(\epsilon)} 2^{t|\lambda|}u_\lambda\psi_\lambda\|_{H^t} \leq \epsilon.$$

Moreover, if for some $0 < s < s^*$ *and* $\tau := (s+1/2)^{-1}$ *the solution* u *belongs to the Besov space* $B_\tau^{t+sd}(L_\tau)$ *then the number* $N(\epsilon) := \#\Lambda(\epsilon)$ *is bounded by* $C_0\epsilon^{-1/s}\|u\|_{B_\tau^{t+sd}(L_\tau)}$. *At most* $C\,N(\epsilon)$ *arithmetic operations and* $C\,N(\epsilon)\log\,N(\epsilon)$ *sorts are needed for the computation of* $u_{\Lambda(\epsilon)}$.

Acknowledgement: We are indepted to Dietrich Braess for valuable suggestions concerning the presentation of the material.

References

[1] A. Averbuch, G. Beylkin, R. Coifman, and M. Israeli, Multiscale inversion of elliptic operators, in: Signal and Image Representation in Combined Spaces, J. Zeevi, R. Coifman (eds.), Academic Press, 1995, ?-??.

[2] I. Babuška and A. Miller, A feedback finite element method with a-posteriori error estimation: Part I. The finite element method and some basic properties of the a-posteriori error estimator, Comput. Methods Appl. Mech. Engrg. **61** (1987), 1–40.

[3] I. Babuška and W.C. Rheinboldt, Error estimates for adaptive finite element computations, SIAM J. Numer. Anal. **15** (1978), 736–754.

[4] R.E. Bank, A.H. Sherman and A. Weiser, Refinement algorithms and data structures for regular local mesh refinement, in: R. Stepleman et al. (eds.), *Scientific Computing*, Amsterdam: IMACS, North-Holland, 1983, 3–17.

[5] R.E. Bank and A. Weiser, Some a posteriori error estimates for elliptic partial differential equations, Math. Comp., **44** (1985), 283–301.

[6] S. Bertoluzza, A-posteriori error estimates for wavelet Galerkin methods, Appl Math. Lett., **8** (1995), 1–6.

[7] S. Bertoluzza, An adaptive collocation method based on interpolating wavelets, in: *Multiscale Wavelet Methods for PDEs*, W. Dahmen, A. J. Kurdila, P. Oswald (eds.), Academic Press, San Diego, 1997, 109–135.

[8] G. Beylkin, R. R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I, Comm. Pure and Appl. Math., **44** (1991), 141–183.

[9] G. Beylkin and J. M. Keiser, An adaptive pseudo-wavelet approach for solving nonlinear partial differential equations, in: *Multiscale Wavelet Methods for PDEs*, W. Dahmen, A. J. Kurdila, P. Oswald (eds.), Academic Press, San Diego, 1997, 137–197.

[10] F. Bornemann, B. Erdmann, and R. Kornhuber, A posteriori error estimates for elliptic problems in two and three space dimensions, SIAM J. Numer. Anal., **33** (1996), 1188–1204.

[11] D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, Cambridge University Press, 1997

[12] S. Brenner and L.R. Scott, *The mathematical theory of finite element methods*, Springer Verlag, New York, 1994.

[13] C. Canuto and I. Cravero, Wavelet-based adaptive methods for advection-diffusion problems, Preprint, University of Torino, 1996.

[14] A. Cohen, Wavelet methods in Numerical Analysis, to appear in the Handbook of Numerical Analysis, vol. VII, 1998.

[15] A. Cohen and R. Masson, Wavelet adaptive methods for elliptic equations - Preconditioning and adaptivity, Preprint, LAN, Université Pierre et Marie Curie, Paris, 1997, to appear in SIAM J. Sci. Comp.

[16] S. Dahlke, W. Dahmen, and R. DeVore, Nonlinear approximation and adaptive techniques for solving elliptic equations, in: *Multiscale Techniques for PDEs*, W. Dahmen, A. Kurdila, and P. Oswald (eds), Academic Press, 1997, San Diego, 237–284.

[17] S. Dahlke, W. Dahmen, R. Hochmuth, and R. Schneider, Stable multiscale bases and local error estimation for elliptic problems, Applied Numerical Mathematics, **23** (1997), 21–47.

[18] S. Dahlke and R. DeVore, Besov regularity for elliptic boundary value problems, Communications in PDEs, **22**(1997), 1–16. .

[19] W. Dahmen, Wavelet and multiscale methods for operator equations, Acta Numerica **6** , Cambridge University Press, 1997, 55–228.

[20] W. Dahmen, S. Müller, and T. Schlinkmann, Multigrid and multiscale decompositions, in: Large-Scale Scientific Computations of Engineering and Environmental Problems, M. Griebel, O.P. Iliev, S.D. Margenov, and P.S. Vassilevski, eds., Notes on Numerical Fluid Mechanics, Vol. 62, Vieweg, Braunschweig/Wiesbaden, 18–41, 1998.

[21] W. Dahmen, S. Prößdorf, and R. Schneider, Multiscale methods for pseudo-differential equations on smooth manifolds, in: *Proceedings of the International Conference on Wavelets: Theory, Algorithms, and Applications*, C.K. Chui, L. Montefusco, and L. Puccio (eds.), Academic Press, 1994, 385-424.

[22] W. Dahmen and R. Schneider, Wavelets on manifolds I, Construction and domain decomposition, IGPM-Report # 149, RWTH Aachen, Jan 1998.

[23] W. Dahmen and R. Schneider, Wavelets on manifolds II, Application to boundary integral equations, in preparation.

[24] W. Dahmen, R. Stevenson, Element-by-element construction of wavelets - stability and moment conditions, IGPM-Report # 145, RWTH Aachen, Dec. 1997.

[25] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, **61**, SIAM Philadelphia, 1988.

[26] R. DeVore, Nonlinear approximation, Acta Numerica **7**, Campbridge University Press, 1998, 51-150.

[27] R. DeVore, B. Jawerth and V. Popov, Compression of wavelet decompositions, Amer. J. Math., **114** (1992), 737–785.

[28] R. DeVore, V. Popov, Interpolation spaces and nonlinear approximation, in: *Function Spaces and Approximation*, M. Cwikel et al, eds., Lecture Notes in Mathematics, vol.1302, 1998, Springer, 191–205.

[29] R. DeVore and V. Temlyakov, Some remarks on greedy algorithms, Advances in Computational Math., **5** (1996), 173–187.

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

[31] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, Introduction to adaptive methods for differential equations, Acta Numerica **4**, Cambridge University Press, (1995), 105–158.

[32] M. Frazier, B. Jawerth, and G. Weiss, *Littlewood-Paley theory and the study of function spaces*, CBMS Conference Lecture Notes 79, (AMS, Providence, RI), 1991.

[33] Y. Meyer, *Ondelettes et Operateurs, Vol 1 and 2*, Hermann, Paris, 1990

[34] E. Novak, On the power of adaptation, J. Complexity, **12**(1996), 199–237.

[35] T. von Petersdorff, and C. Schwab, Fully discrete multiscale Galerkin BEM, in: *Multiscale Wavelet Methods for PDEs*, W. Dahmen, A. Kurdila, and P. Oswald (eds.), Academic Press, San Diego, 1997, 287–346.

[36] R. Schneider, *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung groß er vollbesetzter Gleichungssysteme*, Habilitationsschrift, Technische Hochschule, Darmstadt, 1995.

[37] P. Tchamitchian, Wavelets, Functions, and Operators, in: *Wavelets: Theory and Applications*, G. Erlebacher, M.Y. Hussaini, and L. Jameson (eds.), ICASE/LaRC Series in Computational Science and Engineering, Oxford University Press, 1996, 83–181.

Albert Cohen
Laboratoire d'Analyse Numerique
Universite Pierre et Marie Curie
4 Place Jussieu, 75252 Paris cedex 05
France
e–mail: `cohen@ann.jussieu.fr`
WWW: `http://www.ann.jussieu.fr/∼cohen/`
Tel: 33-1-44277195, Fax: 33-1-44277200

Wolfgang Dahmen
Institut für Geometrie und Praktische Mathematik
RWTH Aachen
Templergraben 55
52056 Aachen
Germany
e–mail: dahmen@igpm.rwth-aachen.de
WWW: http://www.igpm.rwth-aachen.de/∼dahmen/
Tel: 49-241-803950, Fax: 49-241-8888-317

Ronald 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: 803-777-26323, Fax: 803-777-6527