Numer Algor (2008) 48:161–188
DOI 10.1007/s11075-008-9164-0

ORIGINAL PAPER

A wavelet-based nested iteration–inexact conjugate gradient algorithm for adaptively solving elliptic PDEs

Carsten Burstedde · Angela Kunoth

Received: 4 October 2007 / Accepted: 4 January 2008 /
Published online: 1 February 2008

Abstract In Cohen et al. (Math Comput 70:27–75, 2001), a new paradigm for the adaptive solution of linear elliptic partial differential equations (PDEs) was proposed, based on wavelet discretizations. Starting from a well-conditioned representation of the linear operator equation in infinite wavelet coordinates, one performs perturbed gradient iterations involving approximate matrix–vector multiplications of finite portions of the operator. In a bootstrap-type fashion, increasingly smaller tolerances guarantee convergence of the adaptive method. In addition, coarsening performed on the iterates allow one to prove asymptotically optimal complexity results when compared to the wavelet best N-term approximation. In the present paper, we study adaptive wavelet schemes for symmetric operators employing inexact conjugate gradient routines. Inspired by fast schemes on uniform grids, we incorporate coarsening and the adaptive application of the elliptic operator into a nested iteration algorithm. Our numerical results demonstrate that the runtime of the algorithm is linear in the number of unknowns and substantial savings in memory can be achieved in two and three space dimensions.

This work was supported by the Deutsche Forschungsgemeinschaft (SFB 611) at Universität Bonn.

C. Burstedde
ICES, The University of Texas at Austin,
Austin, TX, USA
e-mail: carsten@ices.utexas.edu

A. Kunoth (✉)
Institut für Mathematik, Universität Paderborn,
Paderborn, Germany
e-mail: kunoth@math.uni-paderborn.de

Keywords Elliptic PDEs · Biorthogonal spline-wavelets · Optimal preconditioning · Adaptive method · Nested iteration · Inexact conjugate gradient (CG) method

Mathematics Subject Classifications (2000) 65N99 · 65K10

1 Introduction

Wavelets as a discretization tool for partial differential equations (PDEs) received their first attention in the context of *preconditioning* linear elliptic PDEs. When iteratively solving the linear system resulting from discretization, the convergence speed of any iterative method depends on the spectral condition number of the system matrix. Numerous studies over the past decades were concerned with developing (asymptotically) optimal preconditioners such that the linear system can be solved in an amount of arithmetic operations that is *linear* in the number of unknowns. Optimal preconditioners typically exhibit a multiscale structure like multigrid [9, 10, 30] or multilevel schemes [8, 21, 33]. Specifically, discretizations based on (appropriately scaled) wavelets yield *well-conditioned* systems in ℓ_2 [21], entailing that the convergence speed of iterative solvers like the conjugate gradient (CG) method for symmetric systems does not depend on the discretization. Assuming nested discretization spaces on uniform grids, the system matrix can be applied in an amount of arithmetic operations which is of optimal *linear complexity* by employing the fast wavelet transform. In addition, one can benefit from a technique known from multigrid as *nested iteration*: starting on some coarse refinement level, one iterates the approximate solution up to matching discretization error accuracy on each level before prolongating the final iterate to the next higher level. Combining wavelet preconditioners with nested iteration, it was shown, e.g., in [22], that the solution of the linear system with respect to a fine uniform grid can be obtained up to discretization error accuracy with optimal linear complexity in the number of unknowns on the final grid. Optimized constructions of wavelets on bounded domains developed in [11] yield small absolute iteration numbers in the numerical experiments in up to three spatial dimensions. Recall that, without nested iteration, an additional log-factor comes into play.

A further most substantial feature offered by wavelets for the efficient numerical solution of PDEs is the built-in potential to locally *adapt* discretizations to singularities in the data, the coefficients or the domain. The first such adaptive scheme for a linear elliptic PDE which was proven to converge was developed in [12]. It contains a number of new concepts when compared to standard finite element discretizations: first, the linear operator equation is discretized in the underlying Sobolev space in terms of a (scaled) wavelet basis, leading to a well-conditioned but still infinite system in ℓ_2. Second, the iterative solution of the system is based on perturbed gradient iterations where the application of the (infinite) system matrix is executed approximately

up to a certain precision. Third, the adaptive 'refinement' of the solution in wavelet coordinates is achieved by dynamically updated accuracy tolerances in the application of the operator. In this way, the elliptic PDE can be solved up to an arbitrary accuracy and, consequently, convergence of the adaptive algorithm can be proved using a perturbation argument. An additional feature of the scheme is *coarsening* the iterates: after a certain number of iterations, small wavelet coefficients are discarded in a systematic fashion in order not to spoil the overall complexity while not perturbing the iterates too much. In this way, it was possible to prove optimal *convergence rates* when compared to the wavelet-best N-term approximation of the solution. Here, techniques from nonlinear approximation theory come into play. The theoretical results are confirmed by according numerical experiments in [1, 2]. In [28], a variant of the original scheme developed in [12] was proposed which does not require coarsening of the iterates but still allows for optimal complexity estimates. A somewhat different type of adaptive wavelet scheme treating problems on more general domains was developed in [6]. Further aspects on employing wavelets for the solution of PDEs, including stability issues for stationary saddle point problems and evolution problems such as conservation laws, and related mathematical concepts may be found in the surveys [3, 13, 15, 18, 19, 32].

The focus of this paper is the adaptive solution of linear elliptic PDEs in wavelet discretization using *conjugate gradient (CG)* schemes. Moreover, deviating from the original scheme developed in [12], we use in addition a *nested iteration* strategy. In the context of PDE-constrained elliptic control problems with distributed control, a first such scheme was proposed in [11]. In this paper, we numerically study in a systematic way for a single elliptic PDE the performance and the convergence history of an iterative scheme consisting of an inexact CG method in adaptive wavelet coordinates. In particular, we compare our scheme to a CG method applied to the original one from [12]. We show in a set of experiments that our adaptive algorithm inherits the runtime-optimality of the original scheme and leads to overall better results than the scheme without nested iteration.

The remainder of this paper is structured as follows. In Section 2, we formulate the elliptic PDE in weak form in wavelet coordinates. Section 3 contains a description of the nested iteration algorithm combined with inexact conjugate gradient iterations which is shown to be asymptotically optimal for uniform discretizations in the number of unknowns on the finest uniform grid. Section 4 recalls basic terminology from nonlinear approximation theory and the relevant ingredients of adaptive schemes, and proposes the new adaptive wavelet algorithm combined with nested iteration. Numerical results discussing iteration numbers, convergence histories and complexity are provided in Section 5. Our findings are summarized in Section 6.

Throughout the paper, we use the relation $a \sim b$ to express $a \lesssim b$ and $a \gtrsim b$, which means that a can be estimated from above and below by a constant multiple of b independent of all parameters on which a or b may depend.

2 An elliptic PDE in wavelet coordinates

2.1 Problem formulation

The prototype of a linear elliptic boundary value problem is as follows. Let $a(\cdot,\cdot) : Y \times Y \to \mathbb{R}$ be a symmetric, continuous and Y–elliptic bilinear form, i.e., $a(v,v) \sim \|v\|_Y^2$, $v \in Y$. Employing the dual form $\langle\cdot,\cdot\rangle = \langle\cdot,\cdot\rangle_{Y'\times Y}$, it defines a linear operator $A : Y \to Y'$ by $\langle Av, w\rangle := a(v,w)$ which is *boundedly invertible* as an operator from Y to Y',

$$\|Av\|_{Y'} \sim \|v\|_Y, \quad v \in Y. \tag{1}$$

For a bounded Lipschitz domain $\Omega \subset \mathbb{R}^n$, a Dirichlet problem can be formulated for $a(v,w) := \int_\Omega \nabla v \cdot \nabla w\, dx$, $Y := H_0^1(\Omega)$ implying $Y' = H^{-1}(\Omega) = (H_0^1(\Omega))'$, which corresponds to the standard weak form of the elliptic boundar value problem

$$-\Delta y = f \quad \text{in } \Omega, \qquad y = 0 \quad \text{on } \partial\Omega.$$

Its weak form can be written as a linear operator equation: given $f \in Y'$, find $y \in Y$ which solves

$$Ay = f. \tag{2}$$

2.2 Wavelet bases

We assume that we have at our disposal a *wavelet basis*

$$\Psi = \{\psi_\lambda : \lambda \in \mathcal{I}\} \subset Y \tag{3}$$

for the Hilbert space Y. It is indexed by elements $\lambda = \lambda(j, \mathbf{k}, \mathbf{e})$ from an infinite set $\mathcal{I}$ which comprise the *refinement scale* or *level of resolution* $j := |\lambda|$, the spatial location $\mathbf{k} = \mathbf{k}(\lambda) \in \mathbb{Z}^n$ and the type of wavelet $\mathbf{e} \in \{0,1\}^n$ for $n > 1$.

The wavelet basis Ψ is supposed to have the following properties. First, it is a *Riesz basis* for Y: every $v \in Y$ possesses a unique expansion in terms of Ψ,

$$v = \sum_{\lambda\in\mathcal{I}} v_\lambda\, \psi_\lambda =: \mathbf{v}^T\, \Psi, \qquad \mathbf{v} := (v_\lambda)_{\lambda\in\mathcal{I}}, \tag{4}$$

and its expansion coefficients $\mathbf{v}$ satisfy a *norm equivalence*

$$\|\mathbf{v}\| \sim \|\mathbf{v}^T\Psi\|_Y, \qquad \mathbf{v} \in \ell_2(\mathcal{I}). \tag{5}$$

We always abbreviate ℓ_2 norms as $\|\cdot\| := \|\cdot\|_{\ell_2(\mathcal{I})}$. Second, Ψ is *local*, that is, $\operatorname{diam}(\operatorname{supp}\psi_\lambda) \sim 2^{-|\lambda|}$.

We will view Ψ both as a *collection* of functions as in (3) as well as a (possibly infinite) column *vector*. For a countable collection of functions Θ and some single function σ, we denote by $\langle\Theta,\sigma\rangle$ the column vector with entries $\langle\theta,\sigma\rangle$, $\theta \in \Theta$. For two collections Θ, Σ, the quantity $\langle\Theta,\Sigma\rangle$ is then a matrix with entries $(\langle\theta,\sigma\rangle)_{\theta\in\Theta,\,\sigma\in\Sigma}$.

By duality, (5) is equivalent to the existence of a set of functions which is *dual* or *biorthogonal* to Ψ,

$$\tilde{\Psi} = \{\tilde{\psi}_\lambda : \lambda \in I\!I\} \subset Y', \qquad \langle \Psi, \tilde{\Psi} \rangle = \mathbf{I} \tag{6}$$

(employing the infinite identity matrix $\mathbf{I}$) which is a Riesz basis for Y', that is, for any $\tilde{v} = \tilde{\mathbf{v}}^T \tilde{\Psi} \in Y'$, one has

$$\|\tilde{\mathbf{v}}\| \sim \|\tilde{\mathbf{v}}^T \tilde{\Psi}\|_{Y'}, \tag{7}$$

see [17, 19].

For the problem posed in Section 2.1, Y is the Sobolev space $H_0^1(\Omega)$, and Ψ and its dual $\tilde{\Psi}$ can be obtained from a dual wavelet pair for $L_2(\Omega)$ by scaling with a diagonal matrix. Here we have been using the following construction, starting with B–splines of order d on the primal side. Biorthogonal spline-wavelets on the real line based on Fourier techniques [14] have been adapted to the interval in [23]. As their L_2 stability constants directly affect the spectral condition number of the elliptic operator A in wavelet coordinates, this construction has been modified by introducing cheap linear transformations in order to achieve small absolute values of the condition number. Among these are sophisticated diagonal scalings involving the structure of the energy inner product $a(\cdot, \cdot)$ [1, 11] and inexpensive local basis transformations in terms of singular value decompositions of selected parts of the representation of the elliptic operator [11]. Combined with taking tensor products of one-dimensional wavelets, this enables the efficient solution of problems in arbitrary spatial dimensions on domains $\Omega = (0, 1)^n$ only limited by storage. For domains which can be represented as a union of parametric mappings of cubes, a very general construction based on topological isomorphisms without restrictions on the range of Sobolev smoothness indices has been proposed in [26] and optimized in [31]. Further constructions of composite wavelets have been developed in [16, 25]. Our preference for biorthogonal spline-wavelets is motivated by the fact that we can work computationally with local linear combinations of continuous piecewise polynomials while still being able to exploit the full functional analytic background provided by the theory.

2.3 The problem in ℓ_2

The discretized but yet infinite-dimensional formulation of problem (2) in wavelet coordinates is derived now as follows. Expanding $y = \mathbf{y}^T \Psi$ and testing with the elements of Ψ, (2) attains the form

$$\mathbf{A}\mathbf{y} = \mathbf{f}, \tag{8}$$

where

$$\mathbf{A} := a(\Psi, \Psi), \qquad \mathbf{f} := \langle \Psi, f \rangle . \tag{9}$$

The ellipticity of A (1) together with the Riesz basis property (5) imply the following well known fact [21, 33].

Theorem 2.1 *The matrix* $\mathbf{A}$ *is a boundedly invertible mapping from* $\ell_2(\mathbb{I})$ *onto itself, that is, there exists constants* $0 < c_{\mathbf{A}} \le C_{\mathbf{A}} < \infty$ *such that for any* $\mathbf{v} \in \ell_2(\mathbb{I})$

$$c_{\mathbf{A}}\|\mathbf{v}\| \le \|\mathbf{A}\mathbf{v}\| \le C_{\mathbf{A}}\|\mathbf{v}\|. \tag{10}$$

Thus, the spectral condition number of $\mathbf{A}$ *satisfies* $\kappa_2(\mathbf{A}) \le \frac{C_{\mathbf{A}}}{c_{\mathbf{A}}}$.

In particular, (10) entails that any finite section of $\mathbf{A}$ is also uniformly well-conditioned.

3 A nested-iteration inexact-CG algorithm

3.1 A conjugate gradient (CG) method

The basic building block of the nested iteration solver considered in this paper is the classical method of conjugate gradients (CG method). We formulate it in infinite wavelet coordinates. For the computations, it will be specified later to finite vectors and applications of finite portions of the system matrix in (8). We abbreviate by

$$\textsc{Res}(\tilde{\mathbf{v}}) := \mathbf{A}\tilde{\mathbf{v}} - \mathbf{f}. \tag{11}$$

the residual using an approximation $\tilde{\mathbf{v}}$ to the exact solution $\mathbf{y}$ of the linear system of equations (8). The CG method iteratively computes an approximate solution $\mathbf{y}_K$ to (8) with given initial guess $\mathbf{y}_0$ and prescribed tolerance $\varepsilon > 0$ such that

$$\|\mathbf{A}\mathbf{y}_K - \mathbf{f}\| = \|\textsc{Res}(\mathbf{y}_K)\| \le \varepsilon, \tag{12}$$

where K denotes the number of iterations used. Later ε will be specified depending on the discretization level for which (8) is set up. The following scheme CG contains a routine $\textsc{Apply}(\mathbf{A}, \mathbf{d}_k, \eta_k)$ which on a full uniform grid is simply the matrix–vector multiplication $\mathbf{A}\mathbf{d}_k$. Otherwise, it approximately computes this product up to a tolerance $\eta_k = \eta_k(\varepsilon)$ depending on ε.

Subroutine CG $[\mathbf{A}, \mathbf{f}, \mathbf{y}_0, \varepsilon] \to \mathbf{y}_\varepsilon$: Computes $\mathbf{y}_\varepsilon$ such that $\|\mathbf{A}\mathbf{y}_\varepsilon - \mathbf{f}\| \le \varepsilon$.

(1) Set $\mathbf{d}_0 := \mathbf{f} - \textsc{Apply}(\mathbf{A}, \mathbf{y}_0, \eta_0(\varepsilon))$ and $\mathbf{r}_0 := -\mathbf{d}_0$. Set $k := 0$.

(2) While $\|\mathbf{r}_k\| > \varepsilon$

$$\begin{aligned}
\mathbf{h}_k &:= \textsc{Apply}(\mathbf{A}, \mathbf{d}_k, \eta_k(\varepsilon)) & \alpha_k &:= \tfrac{(\mathbf{r}_k)^T\mathbf{r}_k}{(\mathbf{d}_k)^T\mathbf{h}_k} \\
\mathbf{y}_{k+1} &:= \mathbf{y}_k + \alpha_k\mathbf{d}_k & \mathbf{r}_{k+1} &:= \mathbf{r}_k + \alpha_k\mathbf{h}_k \\
\beta_k &:= \tfrac{(\mathbf{r}_{k+1})^T\mathbf{r}_{k+1}}{(\mathbf{r}_k)^T\mathbf{r}_k} & \mathbf{d}_{k+1} &:= -\mathbf{r}_{k+1} + \beta_k\mathbf{d}_k \\
k &:= k+1
\end{aligned} \tag{13}$$

(3) Accept $\mathbf{y}_k \to \mathbf{y}_\varepsilon$

The iteration is terminated when the residual is below the tolerance ε. By consequence of (10), this entails that the error in the solution is multiplied by $\|\mathbf{A}^{-1}\| = c_{\mathbf{A}}^{-1}$, that is,

$$\|\mathbf{y} - \mathbf{y}_K\| = \|\mathbf{A}^{-1}(\mathbf{f} - \mathbf{A}\mathbf{y}_K)\| \leq \|\mathbf{A}^{-1}\| \, \|\mathrm{Res}(\mathbf{y}_K)\| \leq \varepsilon \, c_{\mathbf{A}}^{-1}. \tag{14}$$

Since, in practice, the system is solved only approximately, it may be viewed as an *inexact CG method*. The effect of approximate applications of $\mathbf{A}$ in CG and more general Krylov subspace schemes with respect to convergence has been investigated in a numerical linear algebra context for a given linear system (8) in a number of papers, see e.g. [5, 29, 37]. In particular, the results in [37] yield that one can estimate the difference between the actually computed residual $\mathbf{r}_k$ in CG $[\mathbf{A}, \mathbf{f}, \mathbf{y}_0, \varepsilon]$ and $\mathrm{Res}(\mathbf{y}_k) = \mathbf{A}\mathbf{y}_k - \mathbf{f}$ as

$$\|\mathbf{r}_k - \mathrm{Res}(\mathbf{y}_k)\| \leq C_{\mathbf{A}} \sum_{i=0}^{k-1} \eta_i |\alpha_i| \, \|\mathbf{d}_i\|, \tag{15}$$

with $C_{\mathbf{A}}$ from (10) and $\alpha_i, \mathbf{d}_i$ defined in (13). For the iterations in $\mathrm{Apply}(\mathbf{A}, \mathbf{d}_i, \eta_i)$, the bounds $\eta_i = \eta_i(\varepsilon)$ are chosen as $\eta_i = \varepsilon(|\alpha_i| \, \|\mathbf{d}_i\|)^{-1}$. Thus, they may be relaxed as $\|\mathbf{d}_i\|$ becomes smaller. Nevertheless, here we have selected η_i to be proportional to the outer accuracy ε incorporating a safety factor, accounting for the values of α_i and $\|\mathbf{d}_i\|$. In agreement with the literature cited above, we get a linear dependence of the total computing time on the number of unknowns.

3.2 The CG algorithm on nested uniform grids

We now embed algorithm CG into a nested iteration strategy, first on uniform grids. Finite-dimensional trial spaces for Y or, in wavelet coordinates, for $\ell_2(\mathcal{I})$, corresponds in the wavelet setting to pick the set of all indices up to some *highest refinement level* J, i.e., $\mathcal{I}_J := \{\lambda \in \mathcal{I} : |\lambda| \leq J\}$ so that $N_J := \#\mathcal{I}_J < \infty$. The representation of operators and vectors is then derived as in Section 2.3 with respect to this truncated index set which corresponds to deleting all entries referring to indices λ satisfying $|\lambda| > J$. By construction of wavelet bases on bounded domains, there is a *coarsest level* of resolution denoted by j_0.

Remark 3.1 On uniform grids, the computational realization of (8) in wavelet coordinates consists in first setting up the operator and right hand side in terms of the *generator basis* on the finest level J, which consists on $(0, 1)^n$ of tensor products of B-Splines (or linear combinations of these near the boundaries). The application of an operator in the *wavelet basis* is then performed by applying this sparse matrix together with the fast wavelet transform (FWT) and its transpose as a pyramid scheme which in this multiplicative form needs $\mathcal{O}(N_J)$ arithmetic operations and is therefore of optimal linear complexity, see, e.g., [7, 11, 18, 23].

A full nested iteration strategy employing a sequence of CG routines can be described as follows. We start at the coarsest level of resolution j_0 with the

exact $\mathbf{y}^{j_0}$, obtained by using the conjugate gradient method as a direct solver for this very small problem. We prolongate the solution to the next higher level j to obtain an appropriate initial guess. We iteratively solve (8) until the norm of the current residual is below the discretization error on that level. Since, in wavelet coordinates, $\|\cdot\|$ corresponds to the energy norm, and since we employ tensor products of linear combinations of B-splines of order d to construct the primal wavelet basis, the discretization error is expected to be proportional to $2^{-(d-1)j}$ for smooth solutions. Prolongations are accomplished in wavelet coordinates by simply padding the coefficient vector with zeros. (Recall that wavelet coefficients have the character of differences. Thus, this prolongation corresponds to the exact representation in higher resolution wavelet coordinates.) The refinement level is successively increased until on some finest level J a prescribed tolerance proportional to the discretization error $2^{-(d-1)J}$ is met.

We can formulate a *Nested-Iteration Inexact-Conjugate-Gradient* (NIICG) Algorithm as follows. Here, superscripts on vectors denote the refinement level on which this term is computed. The right hand side $\mathbf{f}^j$ is supposed to be accessible on all levels.

Algorithm NIICG $[\mathbf{A}, \mathbf{f}, J] \to \mathbf{y}^J$: Solves $\mathbf{A}\mathbf{y}^J = \mathbf{f}$ to accuracy $\varepsilon_J := 2^{-(d-1)J}$.

(1) Initialization for coarsest level $j := j_0$

(a) Call CG $[\mathbf{A}, \mathbf{f}^{j_0}, \mathbf{0}^{j_0}, 0] \to \mathbf{y}^{j_0}$

(2) While $j < J$

(a) Prolongate $\mathbf{y}^j \to \mathbf{y}_0^{j+1}$ by padding with zeros, set $j := j+1$.

(b) Call CG $[\mathbf{A}, \mathbf{f}^j, \mathbf{y}_0^j, c_{\mathbf{A}}\, 2^{-(d-1)j}] \to \mathbf{y}^j$

In view of Theorem of 2.1, finite portions of the system matrix $\mathbf{A}$ have uniformly bounded condition numbers, entailing that each CG routine employed in the process reduces the error in each iteration step by a fixed amount $\rho < 1$. Let $N_J \sim 2^{nJ}$ be the total number of unknowns on the highest level J.

Remark 3.2 Employing the CG method *only* on the highest level, one needs $\mathcal{O}(J) \sim \mathcal{O}(-\log \varepsilon_J)$ iterations to achieve a prescribed discretization error accuracy $\varepsilon_J \sim 2^{-(d-1)J}$. As each application of $\mathbf{A}$ by Remark 3.1 requires $\mathcal{O}(N_J)$ operations, the solution of (8) by CG iterating only on the finest level, thus, requires $\mathcal{O}(J\, N_J)$ arithmetic operations.

Proposition 3.3 *If the residual* (11) *is computed on each level j up to discretization error proportional to $2^{-(d-1)j}$ and the corresponding solutions are taken as initial guesses for the next higher level,* NIICG *is an asymptotically optimal method: it provides the solution $\mathbf{y}^J$ up to discretization error on level J in an overall amount of $\mathcal{O}(N_J)$ arithmetic operations and is, therefore, of optimal linear complexity.*

The argumentation which was briefly sketched in [22] is as follows. Continuing in the line of Remark 3.2, nested iteration allows to get rid of the factor J in the total amount of operations: starting with the exact solution on the coarsest level j_0, one needs only a fixed amount of iterations to reduce the error up to discretization error accuracy $\varepsilon_j \sim 2^{-(d-1)j}$ on each subsequent level j, taking the solution from the previous level as initial guess. Thus, on each level, one needs $\mathcal{O}(N_j)$ operations to realize discretization error accuracy ε_j. Since the spaces are nested and the number of unknowns on each level grows like $N_j \sim 2^{nj}$, by a geometric series argument the total number of arithmetic operations stays proportional to $\mathcal{O}(N_J)$.

4 Nonlinear approximation and adaptive algorithms

Up to now, it was implicitly assumed that the solution of (2) is sufficiently smooth in order to allow for an a-priori error estimate between the exact solution y and its Galerkin approximation y_J on level J,

$$\|y - y_J\|_Y \lesssim 2^{-(d-1)J} \|y\|_{H^d(\Omega)}. \tag{16}$$

In other words, one needs $N_J \sim 2^{nJ}$ degrees of freedom on a grid for $\Omega \subset \mathbb{R}^n$ with grid spacing $h \sim 2^{-J}$ to achieve an accuracy of order $N_J^{-(d-1)/n}$. *Adaptive* approximations of y from a space less smooth than $H^d(\Omega)$, allowing for singularities, aim to recover the same *approximation rate* $\sigma := (d-1)/n$ with N degrees of freedom, distributed nonuniformly and adapted to singularities. Expressed in terms of the target accuracy $\varepsilon \sim N^{-\sigma}$, one expects to need $\varepsilon^{-1/\sigma} \sim N$ arithmetic operations.

At this point, we need to recall some facts from nonlinear approximation theory in order to classify which functions or, in wavelet coordinates, which sequences allow for such rates. In fact, adaptivity in the wavelet framework is realised by choosing an index set of given cardinality N in such a way that the approximation error is minimized. This is called (wavelet-best) *N-term approximation*.

4.1 Nonlinear approximation theory

We provide a short introduction here, largely following [12].

4.1.1 Weak ℓ_τ spaces and their connection to Besov spaces

An important role in nonlinear approximation is played by a finite set of wavelet indices denoted by $\Lambda \subset \mathcal{I}$. It determines the nonlinear space of vectors with at most N nonzero entries $\Sigma_N \subset \ell_2(\mathcal{I})$. For $\sigma \geq 0$, we denote by $\mathcal{A}^\sigma$ the set of all vectors $\mathbf{v} \in \ell_2$ which can be approximated with order $\mathcal{O}(N^{-\sigma})$ by the elements of Σ_N, i.e.,

$$\|\mathbf{v}\|_{\mathcal{A}^\sigma} := \sup_{N \geq 0} (N+1)^\sigma \inf_{\mathbf{w} \in \Sigma_N} \|\mathbf{w} - \mathbf{v}\| < \infty.$$

One can characterize $\mathcal{A}^\sigma$ by a *decreasing rearrangement* $\mathbf{v}^*$ of $\mathbf{v}$ as follows. Denote by v_N^* the N-th largest of the coefficients $|v_\lambda|$, and set $\mathbf{v}^* := (v_N^*)_{N=1}^\infty$. Introducing the parameter $0 < \tau < 2$, let ℓ_τ^{w} denote the set of all $\mathbf{v} \in \ell_2$ for which $|\mathbf{v}|_{\ell_\tau^{\mathrm{w}}} := \sup_{N\geq 1} N^{1/\tau} v_N^* < \infty$. Alternatively, the vectors in ℓ_τ^{w} can be characterized by $\#\{\lambda : |v_\lambda| \geq \varepsilon\} \lesssim \varepsilon^{-\tau}$. The space ℓ_τ^{w} is called *weak* ℓ_τ which is a special case of a *Lorentz sequence space*. The terminology stems from the inclusions $\ell_\tau \subsetneq \ell_\tau^{\mathrm{w}} \subsetneq \ell_2$. Defining the quasi-norm $\|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}} := \|\mathbf{v}\| + |\mathbf{v}|_{\ell_\tau^{\mathrm{w}}}$, the following relations hold.

Proposition 4.1 [20, 27] *Given $\sigma > 0$, let τ be determined by*

$$\frac{1}{\tau} = \frac{1}{2} + \sigma. \tag{17}$$

Then $\mathbf{v}$ belongs to $\mathcal{A}^\sigma$ if and only if $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$, and we have the equivalence $\|\mathbf{v}\|_{\mathcal{A}^\sigma} \sim \|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}}$. In particular, if $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$, then

$$\inf_{\mathbf{w}\in\Sigma_N} \|\mathbf{v} - \mathbf{w}\| \lesssim N^{-\sigma} \|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}}, \qquad N \geq 1. \tag{18}$$

The limit case $\tau = 2$ corresponds to the standard ℓ_2 space. Then we have $\sigma = 0$, i.e, the approximation error does not decay at all. Thus, nonlinear approximation to $\mathbf{v}$ produces a better rate than linear approximation if and only if $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$ for $0 < \tau < 2$.

Finally, we briefly recall an important relation between sequence spaces and Besov function spaces [20] which provide a larger class than Sobolev spaces for the same smoothness index and, thereby, allow for isolated singularities. We dispense here with the formal definition of a Besov norm but only cite the most relevant norm equivalence which generalizes (5). Let $f = \mathbf{c}^T \Psi$ be the representation of a function f in terms of the L_2-stable wavelet basis Ψ. Then it was proved in, e.g., [20] that

$$\|f\|_{B_{q,p}^\alpha} \sim \left(\sum_{j=j_0-1}^{\infty} 2^{jq(\alpha+\frac{n}{2}-\frac{n}{p})} \left(\sum_{k\in\nabla_j} |c_{j,k}|^p \right)^{q/p} \right)^{1/q} \tag{19}$$

which reduces to (5) for $p = q = 2$ and $\alpha = 1$. Setting $p = q = \tau$ and $\alpha = \sigma n + \beta$, we obtain in view of (17)

$$\|f\|_{B_{\tau,\tau}^{\sigma n+\beta}} \sim \|\mathbf{D}^\beta \mathbf{c}\|_{\ell_\tau}, \tag{20}$$

where $\mathbf{D}$ is a diagonal matrix based on the scalings in (19). The condition $\mathbf{D}^\beta \mathbf{c} \in \ell_\tau^{\mathrm{w}}$ leads to slightly larger spaces: they contain precisely the functions whose best N-term wavelet approximation in the energy norm produces an error decaying like $\mathcal{O}(N^{-\sigma})$. Note that the Sobolev embedding theorem yields $B_{\tau,\tau}^{\sigma n+\beta} \subset H^\beta$ for $\tau^{-1} \leq \frac{1}{2} + \sigma$, which means that Besov spaces with $p = q = \tau < 2$ are larger than Sobolev spaces of the same smoothness index. Thus, for smaller values of τ singularities can be resolved increasingly better. Consequently, optimal convergence rates can be guaranteed using adaptive wavelet methods for a larger class of functions compared to uniform discretizations.

4.1.2 Coarsening strategy

The first ingredient for the adaptive wavelet algorithm described below is a routine which reduces the number of nonzero coefficients in an adaptive wavelet representation in a controlled fashion, called *coarsening*. Note that (20) guarantees that the error made in coarsening a vector is equivalent to the error in the corresponding function. Essential is a decreasing rearrangement of an input vector $\mathbf{v}$, i.e., reordering $\mathbf{v}$ according to the size of the moduli of its entries. Requiring that the error is bounded, the purpose of the routine Coarse, originally described in [12], is to reduce the number of nonzero entries of a vector $\mathbf{v}$ as much as possible. In a naive implementation, sorting introduces a logarithmic factor in the runtime complexity. Instead, it was proposed in [1] to perform a binary binning which still provides a sufficient rearrangement and which increases the error by at most a constant factor but is of optimal linear complexity. The amount by which the *support* $\operatorname{supp}\mathbf{v}$ of $\mathbf{v}$, the number of nonzero entries, can be reduced by this routine depends on the class ℓ_τ^{w} to which $\mathbf{v}$ belongs.

Subroutine Coarse $[\mathbf{v}, \eta] \to \mathbf{v}_\eta$: Reduces the support of $\mathbf{v}$ such that $\|\mathbf{v} - \mathbf{v}_\eta\| \le \eta$.

(1) Define $N := \#(\operatorname{supp}\mathbf{v})$ and sort the nonzero entries of $\mathbf{v}$ into decreasing order, with sorted indices $\lambda_1, \ldots, \lambda_N$.
(2) Compute $\|\mathbf{v}\|_{\ell_2}^2 := \sum_{i=1}^N |\mathbf{v}_{\lambda_i}|^2$.
(3) For $k = 1, 2, \ldots$, form $S_k := \sum_{i=1}^k |\mathbf{v}_{\lambda_i}|^2$ and find the smallest k such that $S_k \ge \|\mathbf{v}\|_{\ell_2}^2 - \eta^2$. Set $\Lambda := \{\lambda_1, \ldots, \lambda_k\}$ and $\mathbf{v}_\eta$ as the projection of $\mathbf{v}$ onto Λ.

The routine Coarse will not only be used repeatedly within the algorithm but also to process the input data. We assume that all wavelet coefficients of the right hand side $\mathbf{f}$ in (8) are known or can be computed up to any desired accuracy. Then $\mathbf{f}$ can be approximated up to arbitrary accuracy by a finite vector in a conceptually similar fashion as in Coarse . From the above theoretical results we know that for any $\mathbf{f} \in \ell_\tau^{\mathrm{w}}$, $0 < \tau < 2$, and any target accuracy $\eta > 0$, there exists an approximation $\mathbf{f}_\eta$ such that $\|\mathbf{f} - \mathbf{f}_\eta\| \le \eta$ and $\#\operatorname{supp}\mathbf{f}_\eta \lesssim \eta^{-1/\sigma}\|\mathbf{f}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma}$, where σ and τ are related by (17).

4.1.3 Quasi-sparse matrices and approximate matrix–vector products

Recall that on uniform grids, the computation of matrix–vector products within the iterative routine CG can be achieved in linear time using a product of uniformly sparse matrices. To ensure optimal computational complexity for our adaptive computations we accordingly only accept a cost of $\mathcal{O}(N)$. Here $N \le N_J$ is the current size of the support of the adaptive wavelet representation of a function. Moreover, the size of the support of the output vector must be controlled. The following two properties are essential to achieve this

goal. First, the stiffness matrix in wavelet representation $\mathbf{A}$ is quasi-sparse: it exhibits a decay of entries away from the diagonal which can be quantified a-priori. Second, the computation of the exact matrix–vector product would result due to the infinite size of $\mathbf{A}$ in an infinite number of nonzero entries in the output vector. Instead, the matrix–vector product will only be computed approximately for which one can, on one hand, guarantee the required accuracy while, on the other hand, ensure optimal linear complexity. This requires some sophisticated ingredients based on increasingly better finite approximations of $\mathbf{A}$ applied to finite portions of a vector, firstly developed in [12]. To describe these tools, note that for a large class of elliptic operators B over Sobolev spaces H^α the decay property

$$2^{-(|\lambda'|+|\lambda|)\alpha}|\langle B\psi_\lambda, \psi'_\lambda\rangle| \lesssim 2^{-||\lambda|-|\lambda'||\gamma}(1+r(\lambda,\lambda'))^{-\beta}, \tag{21}$$

holds, with parameters $\gamma > \frac{n}{2}$, $\beta > n$ and

$$r(\lambda,\lambda') := 2^{\min\{|\lambda|,|\lambda'|\}}\operatorname{dist}(\operatorname{supp}(\psi_\lambda), \operatorname{supp}(\psi_{\lambda'})). \tag{22}$$

Here γ depends on the smoothness of the wavelets, whereas β is related to the approximation order of the dual multiresolution and the order of the operator B. Property (21) has been confirmed for different partial differential and integral operators, see, e.g., [4, 18, 36, 38].

A matrix $\mathbf{B} = (b_{\lambda,\lambda'})_{\lambda,\lambda'}$ is said to be *quasi-sparse* if it belongs to the class $\mathcal{A}_{\gamma,\beta}$, defined as

$$\mathcal{A}_{\gamma,\beta} := \left\{\mathbf{B} : |b_{\lambda,\lambda'}| \lesssim 2^{-||\lambda|-|\lambda'||\gamma}(1+r(\lambda,\lambda'))^{-\beta}\right\}, \tag{23}$$

with $r(\lambda,\lambda')$ defined as in (22). It follows from (21) that $\mathbf{B} \in \mathcal{A}_{\gamma,\beta}$ holds for the expansion of an operator B in a scaled wavelet basis for H^α like in (4). Matrices from $\mathcal{A}_{\gamma,\beta}$ are bounded operators on ℓ_2. They are also *compressible*, see [4, 24, 34, 35, 38] for details on matrix compression. Essential for the reasoning in [12] is the following result.

Proposition 4.2 *For each $\gamma > \frac{n}{2}$ and $\beta > n$, let*

$$\sigma^* := \min\left\{\frac{\gamma}{n} - \frac{1}{2}, \frac{\beta}{n} - 1\right\} \tag{24}$$

and assume that $\mathbf{B} \in \mathcal{A}_{\gamma,\beta}$. Then, for any given $\sigma < \sigma^*$ and every $j \in \mathbb{N}$, there exists a matrix $\mathbf{B}_j$ containing at most 2^j nonzero entries in each row and column which satisfies

$$\|\mathbf{B} - \mathbf{B}_j\| \lesssim 2^{-\sigma j}. \tag{25}$$

Remark 4.3 For the representation of the operator A from (1) in the wavelet basis Ψ (4), the matrices $\mathbf{B}_j$ can be defined as

$$(\mathbf{B}_j)_{\lambda,\lambda'} = \begin{cases} b_{\lambda,\lambda'} & \text{for } ||\lambda| - |\lambda'|| \le \frac{j}{n}, \\ 0 & \text{else,} \end{cases}$$

for any $\sigma \le \frac{\gamma}{n} - \frac{1}{2}$; the parameter β is not relevant in this case.

For an optimal adaptive matrix–vector multiplication, the following slightly more general characterization will be useful.

Definition 4.4 A matrix $\mathbf{B}$ is in $\mathcal{B}_\sigma$ if there are two positive sequences $(\alpha_j)_{j\geq 0}$ and $(\beta_j)_{j\geq 0}$ that are both summable, and if for every $j \geq 0$ there is a matrix $\mathbf{B}_j$ with at most $2^j\alpha_j$ nonzero entries per row and column such that

$$\|\mathbf{B} - \mathbf{B}_j\| \leq 2^{-\sigma j}\beta_j. \tag{26}$$

With σ^* defined by (24), we have $\mathcal{A}_{\gamma,\beta} \subset \mathcal{B}_\sigma$ for every $0 \leq \sigma < \sigma^*$. For $\mathbf{B} \in \mathcal{A}_{\gamma,\beta}$, the sequences $(\alpha_j)_{j\geq 0}$ and $(\beta_j)_{j\geq 0}$ can be chosen to decay exponentially. This means that the stiffness matrix in wavelet representation belongs to $\mathcal{B}_\sigma$. We finally arrive at the main result.

Theorem 4.5 *For $\mathbf{B} \in \mathcal{B}_\sigma$, $\mathbf{B}$ is continuous on ℓ_τ^{w}. For any $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$ and $\varepsilon > 0$, there exists a vector $\mathbf{w}_\varepsilon$ such that*

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_\varepsilon\| \leq \varepsilon, \tag{27}$$

and $\#\operatorname{supp}\mathbf{w}_\varepsilon \lesssim N_\tau(\mathbf{v},\varepsilon) := \varepsilon^{-1/\sigma}\|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma}$ with σ and τ related by (17). Moreover, the number of arithmetic operations to compute $\mathbf{w}_\varepsilon$ is also bounded by a constant multiple of $N_\tau(\mathbf{v},\varepsilon)$.

We explain next how $\mathbf{w}_\varepsilon$ can actually be computed. Let $\mathbf{v}_{[j]} \in \Sigma_{2^j}$ be a best 2^j-term approximation to $\mathbf{v}$ in ℓ_2 as introduced in Section 4.1.1. Set $\mathbf{v}_{[-1]} := 0$ and

$$\mathbf{w}_j := \mathbf{B}_j\left(\mathbf{v}_{[0]} - \mathbf{v}_{[-1]}\right) + \cdots + \mathbf{B}_0\left(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\right). \tag{28}$$

Then the error can be computed as

$$\mathbf{B}\mathbf{v} - \mathbf{w}_j = \mathbf{B}\left(\mathbf{v} - \mathbf{v}_{[j]}\right) + (\mathbf{B} - \mathbf{B}_0)\left(\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\right) + \cdots + (\mathbf{B} - \mathbf{B}_j)\left(\mathbf{v}_{[0]} - \mathbf{v}_{[-1]}\right). \tag{29}$$

Applying the triangle inequality, this sum can be estimated by known quantities: the matrix norms abbreviated as $b := \|\mathbf{B}\|$, $b_j := \|\mathbf{B} - \mathbf{B}_j\|$ and the vector norms $v_j := \|\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\|$ and $\bar{v}_j := \|\mathbf{v} - \mathbf{v}_{[j]}\|$. This yields

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_j\| \leq R_j := b\,\bar{v}_j + b_0 v_j + \cdots + b_j v_0. \tag{30}$$

Employing (18) and (26), we arrive at

$$\|\mathbf{B}\mathbf{v} - \mathbf{w}_j\| \lesssim 2^{-\sigma j}\|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}}, \tag{31}$$

i.e., the ℓ_2 error of the approximate matrix–vector product decays with j for $\sigma > 0$.

The corresponding algorithm Ad-Apply runs through the levels j and constructs the vectors $\mathbf{w}_j$ and norms $\bar{v}_j$ and v_j: starting with $j = 0$, it accepts thefirst j for which the right hand side of (30) is less or equal to ε. For efficiency, we employ the trivial relations $\bar{v}_j^2 = \|\mathbf{v}\|^2 - \|\mathbf{v}_{[j]}\|^2$ and $\|\mathbf{v}_{[j]}\|^2 = \sum_{\ell=0}^{j} v_\ell^2$. The subroutine Ad-Apply thus calculates an approximate matrix–vector product within given error bounds (27). If the input vector satisfies $\mathbf{v} \in \ell_\tau^{\mathrm{w}}$ with $0 < \sigma < \sigma^*$ and (17) holds, then the size of the output vector is bounded by a constant multiple of $N_\tau(\mathbf{v}, \varepsilon)$. Moreover, the number of arithmetic operations is bounded by $CN_\tau(\mathbf{v}, \varepsilon) + 2N$, and the output vector satisfies $\|\mathbf{w}_\varepsilon\|_{\ell_\tau^{\mathrm{w}}} \lesssim \|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}}$.

Subroutine Ad-Apply $[\mathbf{B}, \mathbf{v}, \varepsilon] \to \mathbf{w}_\varepsilon$: Computes $\mathbf{w}_\varepsilon$ such that $\|\mathbf{B}\mathbf{v} - \mathbf{w}_\varepsilon\| \le \varepsilon$.

(1) Define $N := \#(\operatorname{supp} \mathbf{v})$ and sort the nonzero entries of $\mathbf{v}$ into decreasing order. Form $\mathbf{v}_{[0]}$ and $\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}$, $j = 0, \ldots, \lceil \log(N) \rceil$.
(2) Compute $v_j := \|\mathbf{v}_{[j]} - \mathbf{v}_{[j-1]}\|$, $\bar{v}_j := \|\mathbf{v} - \mathbf{v}_{[j]}\|$, $j = 0, \ldots, \lceil \log(N) \rceil$.
(3) For $j = 0, 1, \ldots$, compute the right hand side R_j of (30) and find the smallest j such that $R_j \le \varepsilon$.
(4) For this j, compute $\mathbf{w}_j$ as in (28) and accept $\mathbf{w}_j \to \mathbf{w}_\varepsilon$

The main convergence and complexity result of the original adaptive wavelet algorithm using Coarse and Ad-Apply within a simple Richardson iteration is the following.

Theorem 4.6 *Assume that $\mathbf{A} \in \mathcal{B}_\sigma$ with $0 < \sigma < \sigma^*$. Then for any $\varepsilon > 0$ and $\mathbf{f} \in \ell_\tau^{\mathrm{w}}$, the procedures yield an approximation $\mathbf{y}_\varepsilon$ to $\mathbf{y}$ with $N_\varepsilon := \#\operatorname{supp} \mathbf{y}_\varepsilon < \infty$ degrees of freedom satisfying $\|\mathbf{y} - \mathbf{y}_\varepsilon\| \le \varepsilon$. If the exact solution satisfies $\mathbf{y} \in \ell_\tau^{\mathrm{w}}$, where σ and τ are related by (17), then the algorithm is optimal in the sense that $N_\varepsilon \lesssim \varepsilon^{-1/\sigma} \|\mathbf{y}\|_{\ell_\tau^{\mathrm{w}}}^{1/\sigma}$ and the number of arithmetic operations is bounded by a constant multiple of N_ε.*

4.2 The adaptive nested-iteration CG algorithm

The subroutines Ad-Apply and Coarse constitute the two main ingredients in a fully adaptive wavelet algorithm combined with CG called Ad-CG which is described next. All matrix–vector products used before are now replaced by approximate matrix–vector products performed in the fashion of Ad-Apply . This yields an inexact CG iteration since the conjugate directions are slightly perturbed. In addition, calls to inner routines Coarse are executed after each CG step, which introduces an additional error. This is done in order to reduce the sizes of the supports of all vectors employed in Ad-CG . By a slight abuse of notation, the outputs of these inner Coarse procedures are denoted like the inputs for simplicity.

Subroutine AD-CG $[\mathbf{A}, \mathbf{f}, \mathbf{y}_0, \varepsilon] \to \mathbf{y}_\varepsilon$: Computes $\mathbf{y}_\varepsilon$ such that $\|\mathbf{A}\mathbf{y}_\varepsilon - \mathbf{f}\| \le \varepsilon$.

(1) SET $\mathbf{d}_0 := \mathbf{f} - \mathbf{A}\mathbf{y}_0$ AND $\mathbf{r}_0 := -\mathbf{d}_0$. SET $k := 0$.
(2) WHILE $\|\mathbf{r}_k\| > \varepsilon$
  (A) CALL AD-APPLY $[\mathbf{A}, \mathbf{d}_k, \frac{\eta_k}{2}] \to \mathbf{h}_k$, CALL COARSE $[\mathbf{h}_k, \frac{\eta_k}{2}]$
  (B) $\alpha_k := \frac{(\mathbf{r}_k)^T \mathbf{r}_k}{(\mathbf{d}_k)^T \mathbf{h}_k}$ $\mathbf{y}_{k+1} := \mathbf{y}_k + \alpha_k \mathbf{d}_k$ $\mathbf{r}_{k+1} := \mathbf{r}_k + \alpha_k \mathbf{h}_k$
  (C) $\beta_k := \frac{(\mathbf{r}_{k+1})^T \mathbf{r}_{k+1}}{(\mathbf{r}_k)^T \mathbf{r}_k}$ $\mathbf{d}_{k+1} := -\mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$
  (D) CALL COARSE $[\mathbf{y}_{k+1}, \delta \frac{\eta_k}{C_\mathbf{A}}]$, COARSE $[\mathbf{r}_{k+1}, \delta\eta_k]$, COARSE $[\mathbf{d}_{k+1}, \delta\eta_k]$
  (E) $k := k + 1$
(3) ACCEPT $\mathbf{y}_k \to \mathbf{y}_\varepsilon$

The complete nested iteration algorithm proposed here is then of the form NIICG with the subroutines CG replaced by AD-CG . This leads to an overall scheme which we call AD-NIICG whose precise listing we dispense with here. This scheme is conceptually different to the scheme proposed in [12] in that new degrees of freedom are only allowed on the next finer level introduced by nested iteration. Otherwise, the structure of NIICG remains unchanged. The tolerances ε in each call of AD-CG take on the form $\varepsilon = \varepsilon(j) := c_\mathbf{A} 2^{-(d-1)j}$ and for each j the inner tolerances are set as $\eta_k := \varepsilon(j)$. Moreover, we have introduced an additional heuristic coarsening parameter $0 \le \delta < 1$ in the inner COARSE routines to enforce more compressed inner variables. Replacing the matrix–vector multiplication on uniform grids by its adaptive approximate counterpart, the cost of AD-NIICG is still of optimal linear complexity $\mathcal{O}(N)$.

A few remarks to compare AD-NIICG with the original adaptive wavelet scheme from [12] (but with Richardson iteration replaced by CG) which we abbreviate as AD-ORIGCG are in order. In AD-ORIGCG , the outer loop runs over the target accuracy which is tightened from iteration to iteration. For each accuracy, indices for the wavelet coefficients which are largest in magnitude are added to the set Λ, irrespective of the level. Consequently, the accuracy dominates the selection of indices in Λ and, hence, the refinement level the coefficients stem from. This algorithm is designed to execute the process of nonlinear approximation. In our algorithm AD-NIICG , the level determines the accuracy. We have two competing goals: the scheme of nested iteration to keep the iteration numbers independent of the level, and the routines COARSE and AD-APPLY to control the number of nonzero coefficients.

In order to achieve a statement comparable to Theorem 4.6, we expect that somewhat different arguments need to be employed. The proof of convergence of the original scheme is based on a perturbation of the infinite linear system of equations which would also apply to finite systems derived adaptively within a nested iteration algorithm. To prove optimal complexity is more involved in the present context: even without the nested-iteration framework, a complexity analysis for the conjugate gradient scheme with intermediate iteration vectors appropriately coarsened is to our knowledge not yet available.

Therefore, also the issue whether coarsening of the iterands can be avoided employing the techniques from [28] remains in the present context unsolved.

Our approach focusses on a practically fast solution in the framework of the nested iteration conjugate gradient solver. Consequently, the efficiency of our method is best studied numerically. To this end, we dedicate the next section to an extensive presentation and discussion of numerical experiments.

5 Numerical results

All numerical experiments have been executed on a 3.2GHz Pentium IV computer (family 15, model 4, stepping 1, with 1MB L2 Cache). The computations have been performed within the newly contrived programme framework `BWP` which is written in pure `C` and has been designed to be self-consistent, non-redundant and modular.

We have tested different cases of the weak formulation of the problem $\Delta y + y = f$ with natural Neumann boundary conditions on the domain $\Omega = (0, 1)^n$ for $n \in \{1, 2, 3\}$ with different right hand sides f. The choice of f includes smooth right hand sides as well as isolated singularities. Since there are no singularities in the domain, in the case of a smooth f, uniform discretizations should yield uniformly decaying errors in accordance with (16).

In this section, we will first provide evidence that our algorithm Ad-NIICG is generally much more effective than the original adaptive wavelet scheme Ad-OrigCG from [12] (with CG iterations). Thereafter, we will concentrate on the numerical properties of Ad-NIICG and compare the behaviour for different right hand sides. Recall from (10) that the stiffness matrix in wavelet representation has a uniformly bounded condition number independently of the level of resolution so that in every iteration in Ad-CG the error in the energy norm is reduced by a constant factor. Moreover, the prolongation to a higher level is a trivial operation in the wavelet framework. These facts already provide a basic motivation for the nested iteration principle, since iterations on lower levels are significantly cheaper than on the finest level of resolution, and hence the computation of good starting values is significantly accelerated.

The implementation of an adaptive wavelet strategy brings in another perspective to the subject of numerical efficiency. The computational work required by the matrix–vector operations used in the CG method is determined by the sparsity of the vectors holding the adaptive coefficients. So a nearly full vector causes about the same cost as for a uniform discretization, while a vector with a smaller fraction of nonzero entries can be processed faster. As we will see below, the fill-in characteristics of the nested iteration approach are clearly more favorable than those for Ad-OrigCG . Note that the purpose of the numerical experiments in [2] was to confirm theoretical reduction rates for nonlinear approximation, while here we concentrate on computational efficiency.

The constants $c_{\mathbf{A}}$ and $C_{\mathbf{A}}$ are independent of the resolution and can be approximated by truncated power and inverse power iterations on a moderate level and stored for later reference.

5.1 1D computations

The first tests that we performed systematically concentrate on the behaviour of the new scheme AD-NIICG compared to AD-ORIGCG . The solution on the coarsest level $j_0 = 3$ was always obtained with a direct solver, an operation of negligible cost, and the numbers from this level were left out in the tables. The results can be seen in Table 1 through Table 6. In the top table in each of these, the results for AD-ORIGCG are displayed, in the bottom the results for AD-NIICG . Both tables show a hierarchy of levels j up to the highest level $J = 16$. For the top tables, the first column displays the highest possible refinement level j which is allowed in AD-ORIGCG , that is, each row corresponds to a separate solution of the system at that fixed finest level. The results in the bottom table have been obtained in one run of AD-NIICG , consistently reusing the solution from the previous level as a starting value for the current level.

In both tables the column named #It contains the number of CG iterations which was needed to reach discretization error accuracy. We employ biorthogonal wavelets based on splines of order $d = 2$ on the primal side and the error is measured in ℓ_2 in wavelet coordinates corresponding to the error in the energy

Table 1 1D, $f(x) \equiv 1, \delta = 0$

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
AD-ORIGCG								
4		7	0.0137			97.5	16	3.425e−04
5		9	0.0137			93.2	30	3.245e−04
6		10	0.0137			93.2	60	1.513e−04
7		11	0.0137			94.2	121	6.317e−05
8		12	0.0137			94.2	242	1.615e−05
9		13	0.0137			92.2	473	1.019e−05
10		14	0.0137			94.8	971	5.980e−06
11		15	0.0137			91.1	1,865	2.040e−06
12		16	0.0137			91.6	3,752	3.181e−07
13		17	0.0137			91.2	7,473	1.271e−07
14		18	0.0137			89.8	14,718	5.169e−08
15		19	0.0137			85.9	28,141	1.900e−08
16		20	0.0137			81.3	53,291	6.079e−09
AD-NIICG								
4	0.00e−00	0	0.0137	0.0137	0.0259	52.9	9	7.69e−16
5	0.00e−00	0	0.0137	0.0137	0.0504	27.2	9	7.69e−16
6	0.00e−00	0	0.0137	0.0137	0.0992	13.8	9	7.69e−16
7	0.00e−00	0	0.0137	0.0137	0.197	6.95	9	7.69e−16
8	0.00e−00	0	0.0137	0.0137	0.392	3.49	9	7.69e−16
9	0.00e−00	0	0.0137	0.0137	0.783	1.75	9	7.69e−16
10	0.00e−00	0	0.0137	0.0137	1.56	0.878	9	7.69e−16
11	0.00e−00	0	0.0137	0.0137	3.13	0.438	9	7.69e−16
12	0.00e−00	0	0.0137	0.0137	6.25	0.219	9	7.69e−16
13	0.00e−00	0	0.0137	0.0137	12.5	0.110	9	7.69e−16
14	0.00e−00	0	0.0137	0.0137	25	0.0548	9	7.69e−16
15	0.00e−00	0	0.0137	0.0137	50	0.0274	9	7.69e−16
16	0.00e−00	0	0.0137	0.0137	100	0.0137	9	7.69e−16

norm $\|\cdot\|_{H^1(\Omega)}$. Thus, the discretization error accuracy is here proportional to 2^{-j} (16).

The percentages in the columns labeled P, V, M and S are related to the adaptive efficiency of the method as follows. The column P contains the number of nonzero coefficients in the final solution vector N in relation to the total number of degrees of freedom on the highest uniform level N_J, that is, $P := N/N_J$. In Table 1, this is $(2^3+1)/(2^{16}+1) \approx 0.0137\%$. However, to obtain a valid and fair estimate for the actual cost of the computations, the operation count within the conjugate gradient iterations is predominant. Since different vectors are involved in these computations which have different patterns of nonzero coefficients (possibly changing between CG iterations), we resort to the *average* of the number of nonzero coefficients over the whole solution cycle, denoted by N_{av}. The quotient of this average computational cost with respect to the cost of a uniform implementation on that level is denoted by $S := N_{\text{av}}/N_j$. The columns $V := N_{\text{av}}/N_J$, measuring the average adaptive efficiency in relation to the overall highest level J, and $M := N_j/N_J$, the relative quantity of total unknowns, are included for comparison. The last column contains the error proportional to the error in the energy norm between the exact solution and the result of the iteration on this level, $\varepsilon(\mathbf{y}) := \|\mathbf{y} - \mathbf{y}^j\|$. Finally, $\|\mathbf{r}_j\|$ for AD-NIICG displays the norm of the residual $\|\mathbf{A}\mathbf{y}^j - \mathbf{f}\|$ when the iteration on level j is stopped. Comparing the values for $\|\mathbf{r}_j\|$ and $\varepsilon(\mathbf{y})$ indicate like in (14) the impact of the extreme eigenvalues of $\mathbf{A}$.

In one dimension, we begin with the right hand side $f \equiv 1$, corresponding to the exact solution $y \equiv 1$. This means that the solution can be represented exactly on the coarsest level $j_0 = 3$, at a cost of only 9 degrees of freedom. So we would like the adaptive algorithm to produce this minimal number of active variables. Examining the column S more closely shows a dramatic difference between the top and bottom listings in Table 1. Namely, the fill-in for AD-ORIGCG is almost 100% (that is, there is virtually no adaptation), while AD-NIICG shows perfect adaptation, retaining the minimum amount of nonzero

Table 2 1D, $f(x) \equiv 1$, $\delta = 0.2$

j	#It	P (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
4	7	0.0137	86.3	14	4.550e−04
5	9	0.0137	78.9	26	3.483e−04
6	10	0.0137	80.3	52	1.193e−04
7	11	0.0137	80.5	103	5.384e−05
8	12	0.0137	79.6	204	1.314e−05
9	13	0.0137	79.2	406	6.529e−06
10	14	0.0137	78.4	803	2.511e−06
11	15	0.0137	77.9	1,596	8.475e−07
12	16	0.0137	77.0	3,152	3.914e−07
13	17	0.0137	76.2	6,245	1.190e−07
14	18	0.0137	75.3	12,344	4.351e−08
15	19	0.0137	74.1	24,274	1.571e−08
16	20	0.0137	72.0	47,172	5.583e−09

Top: AD-ORIGCG , bottom: like in Table 1

coefficients $N = 9$. As can be seen in the top table in Table 1, the increasing accuracy requirements produce a logarithmic grow in iteration numbers for the computations on the finest level only. The nested iteration strategy shown in the bottom table demonstrates the perfect pattern of zero iterations, since the start value on the coarsest level is already equal to the exact result.

To counteract the fill-in of the adaptive vectors with possibly very small entries, we have expanded the algorithm with additional coarsening steps on the different variables inside the conjugate gradient loop. These coarsening operations are controlled by the threshold parameter $0 \leq \delta < 1$. Setting δ to zero reduces the algorithm to the previous version, while increasing δ applies stronger thresholding actions to the work variables. Table 2 shows the runs for $f \equiv 1$, $y \equiv 1$ with an additional coarsening parameter of $\delta = 0.2$. The table for AD-NIICG is exactly the same as in Table 1 so we dispense with it here. These results show an increased efficiency for the iterations on the finest level only by about 10%. So it provides a significant but moderate improvement. Our AD-NIICG is far superior, just as in our first example.

Now we choose the right hand side $f(x) = \cos(x)$, retaining the additional coarsening ratio of $\delta = 0.2$. The results are listed in Table 3. Here the solution

Table 3 1D, $f(x) = \cos(x)$, $\delta = 0.2$

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
AD-ORIGCG								
4		3	0.0122			66.2	11	2.085e−02
5		4	0.0213			76.3	25	1.081e−02
6		5	0.0473			74.5	48	5.098e−03
7		6	0.0886			73.5	94	2.755e−03
8		8	0.162			69.0	177	1.463e−03
9		9	0.325			69.2	355	7.402e−04
10		10	0.633			68.9	705	3.802e−04
11		10	1.24			70.3	1,440	1.941e−04
12		11	2.42			69.6	2,852	9.909e−05
13		12	4.81			68.8	5,636	5.073e−05
14		13	9.80			68.2	11,170	2.491e−05
15		14	19.9			67.7	22,191	1.244e−05
16		15	39.8			67.1	43,985	6.313e−06
AD-NIICG								
4	7.15e−04	2	0.0122	0.0191	0.0259	73.7	13	2.65e−02
5	4.99e−04	7	0.0214	0.0307	0.0504	60.9	20	1.27e−02
6	3.37e−04	3	0.0427	0.0731	0.0992	73.7	48	6.30e−03
7	9.87e−05	4	0.0809	0.128	0.197	65.0	84	3.23e−03
8	5.79e−05	4	0.160	0.246	0.392	62.8	161	1.64e−03
9	3.37e−05	4	0.319	0.456	0.783	58.2	299	8.33e−04
10	2.30e−05	4	0.638	0.863	1.56	55.3	566	4.28e−04
11	6.99e−06	7	1.23	1.39	3.13	44.4	911	2.09e−04
12	5.25e−06	3	2.63	3.92	6.25	62.7	2,569	9.83e−05
13	2.12e−06	3	5.16	6.85	12.5	54.8	4,489	5.01e−05
14	1.20e−06	3	10.4	13.6	25	54.4	8,913	2.51e−05
15	6.77e−07	3	20.7	27.5	50	55.0	18,023	1.27e−05
16	2.33e−07	7	39.0	37.7	100	37.7	24,707	5.92e−06

Reduction rates: $\sigma_1 = 1.058$, $\sigma_2 = 1.061$

has contributions on all levels, which can be seen from the values in column P. Scheme Ad-NIICG now requires work on all levels, as can be seen from the iteration counts in #It. Due to the uniformly bounded condition number of the stiffness matrix in wavelet representations, the iteration counts are constant over the levels, but still much lower than the iteration counts required in Ad-OrigCG . This is in agreement with columns S and N_{av}, showing an increase in adaptive efficiency up to a factor of 2 on the finest level from Ad-OrigCG to Ad-NIICG.

In the caption of this and the following tables, we have displayed two numbers which stand for the computational rates: σ_1 is the rate by which the residuum is reduced; it is proportional to $1/n$. The second number σ_2 is the rate by which the error $\varepsilon(\mathbf{y})$ is reduced. Due to the coarsening and the condition number of $\mathbf{A}$, this rate is expected to be slightly smaller than σ_1 in absolute value but still somewhat proportional. Numerically, these values have been computed by regression.

For Table 4, we have chosen as right hand side the function

$$f_1(x) := \sqrt{|x - 1/3|} \tag{32}$$

Table 4 1D, $f_1(x) = \sqrt{|x - 1/3|}$, $\delta = 0.2$

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
Ad-OrigCG								
4		7	0.0137			87.4	14	1.006e−02
5		9	0.0137			83.2	27	8.861e−03
6		13	0.0168			76.3	49	4.668e−03
7		14	0.0276			78.9	101	7.006e−03
8		15	0.0518			80.5	207	4.270e−03
9		16	0.108			81.6	418	6.322e−04
10		17	0.216			82.3	843	3.211e−04
11		18	0.447			82.2	1,684	1.548e−04
12		19	0.894			81.8	3,353	7.783e−05
13		20	1.84			81.5	6,679	3.917e−05
14		21	3.75			81.1	13,281	1.974e−05
15		22	7.50			80.5	26,373	9.989e−06
16		23	14.7			79.5	52,080	5.168e−06
Ad-NIICG								
4	9.93e−04	1	0.0137	0.0130	0.0259	50.2	9	7.45e−03
5	4.48e−04	3	0.0137	0.0289	0.0504	57.3	19	7.18e−03
6	2.96e−04	4	0.0153	0.0605	0.0992	61.0	40	5.93e−03
7	1.65e−04	5	0.0259	0.116	0.197	58.9	76	3.00e−03
8	8.78e−05	5	0.0488	0.236	0.392	60.2	155	1.51e−03
9	4.68e−05	7	0.0931	0.41	0.783	52.4	269	7.32e−04
10	1.97e−05	4	0.197	1.01	1.56	64.7	662	3.46e−04
11	9.70e−06	4	0.385	1.97	3.13	62.9	1,291	1.77e−04
12	5.13e−06	4	0.774	3.89	6.25	62.2	2,549	9.13e−05
13	2.97e−06	7	1.46	6.02	12.5	48.2	3,945	4.52e−05
14	9.10e−07	4	3.11	14.8	25	59.2	9,699	2.12e−05
15	4.62e−07	4	6.16	29.5	50	59.0	19,333	1.07e−05
16	2.44e−07	4	12.2	57.8	100	57.8	37,880	5.48e−06

Reduction rates: $\sigma_1 = 1.000$, $\sigma_2 = 0.934$

displaying a singularity. Compared to the results with the smooth right hand side discussed previously, the iteration counts as well as the ratios of nonzero wavelet coefficients increase. This can be explained by the worse approximation properties of f in this example. Again, the new AD-NIICG scheme shows a much improved adaptive efficiency.

The root singularity of the right hand side f_1 in the previous example satisfies $f_1 \in H^{1-\varrho}(\Omega)$ for any $\varrho > 0$. The capabilities of the wavelet method allow for right hand sides as irregular as $H^{-1}(\Omega)$. Consequently, we construct an irregular right hand side denoted by f_1^D by applying a Sobolev shift operator, $f_1^D := D^{3/2} f_1$. The operator D (which is just a diagonal matrix in wavelet coordinates) is tuned to preserve the mean value of f. This operation enforces that $f_1^D \in H^{-1/2-\varrho}(\Omega)$ for any $\varrho > 0$. The corresponding results are given in Table 5. They show only very little difference when compared to the previous example of higher regularity. We attribute this to the strong smoothing properties of the inverse elliptic operator. To illustrate the behaviour of the method in this case, Fig. 1 shows the distributions of nonzero wavelet coefficients of the solution for the original and the roughened right hand sides. In accordance with the bottom listings of Tables 4 and 5, both lead to about

Table 5 1D, $f = f^D := D^{3/2} f_1, \delta = 0.2$

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
AD-ORIGCG								
4		7	0.0137			87.4	14	1.079e−02
5		11	0.0137			73.8	24	8.483e−03
6		14	0.0168			74.2	48	5.220e−03
7		15	0.0260			76.3	98	2.564e−03
8		15	0.0474			81.2	208	1.244e−03
9		16	0.0900			82.1	421	6.285e−04
10		17	0.194			82.5	845	3.150e−04
11		18	0.416			82.6	1,691	1.492e−04
12		19	0.869			82.3	3,373	7.482e−05
13		20	1.75			82.1	6,729	3.730e−05
14		21	3.50			81.7	13,386	1.878e−05
15		22	7.05			81.1	26,586	9.631e−06
16		23	13.8			80.1	52,502	4.986e−06
AD-NIICG								
4	1.17e−03	1	0.0137	0.0130	0.0259	50.2	9	9.19e−03
5	5.24e−04	3	0.0137	0.0315	0.0504	62.5	21	8.94e−03
6	3.04e−04	4	0.0168	0.0618	0.0992	62.3	41	6.03e−03
7	1.79e−04	4	0.0244	0.128	0.197	65.0	84	3.29e−03
8	7.56e−05	8	0.0412	0.189	0.392	48.2	124	1.77e−03
9	4.51e−05	4	0.0809	0.498	0.783	63.6	326	1.02e−03
10	2.00e−05	4	0.174	0.957	1.56	61.3	627	6.18e−04
11	1.13e−05	4	0.346	1.97	3.13	62.9	1,291	3.97e−04
12	5.37e−06	4	0.706	3.79	6.25	60.6	2,484	2.61e−04
13	1.46e−06	5	1.40	6.52	12.5	52.2	4,273	1.69e−04
14	7.93e−07	5	2.82	12.6	25	50.4	8,258	1.04e−04
15	5.36e−07	5	5.65	26.0	50	52.0	17,040	5.40e−05
16	3.31e−07	5	11.2	50.4	100	50.4	33,031	5.68e−06

Reduction rates: $\sigma_1 = 1.019, \sigma_2 = 0.805$

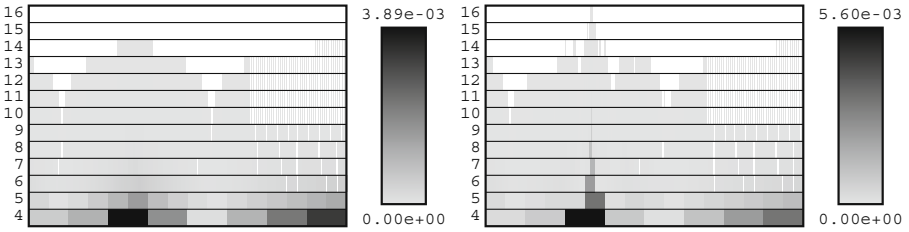

Fig. 1 These graphics show the distribution and magnitudes of the wavelet coefficients of the approximate solution $\mathbf{y}^J$. The *left image* corresponds to the right hand side f_1 defined in (32) and the *right image* to the roughened version f_1^D. The coefficients are horizontally arranged according to their location on the x-axis. The vertical layers are ordered by the increasing level of resolution j listed numerically on the *left edge*. The coarsest level j_0 is left out. From level to level, the number of coefficients doubles, which leads to the diminishing width of the *shaded rectangles from bottom to top*

the same adaptive efficiency on average. The stronger singularity at $x = 1/3$ can be identified as a sharper concentration of coefficients in the right hand picture. This is in perfect agreement with the theoretical results on nonlinear approximation. The large amount of dark (blue) coefficients—corresponding

Table 6 1D, $f = f^D$

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
Reduction rates: $\sigma_1 = 1.037$, $\sigma_2 = 0.806$ ($\delta = 0.3$)								
4	1.17e−03	1	0.0137	0.0118	0.0259	45.6	8	9.19e−03
5	5.18e−04	3	0.0137	0.0276	0.0504	54.8	18	8.93e−03
6	3.29e−04	4	0.0168	0.0545	0.0992	54.9	36	6.03e−03
7	1.81e−04	4	0.0229	0.116	0.197	58.9	76	3.52e−03
8	7.30e−05	8	0.0366	0.154	0.392	39.3	101	1.93e−03
9	2.63e−05	5	0.0748	0.403	0.783	51.5	264	1.06e−03
10	1.23e−05	5	0.156	0.779	1.56	49.9	511	6.41e−04
11	6.49e−06	5	0.317	1.59	3.13	50.8	1,042	4.05e−04
12	5.67e−06	4	0.671	3.47	6.25	55.5	2,274	2.62e−04
13	1.86e−06	5	1.30	5.81	12.5	46.5	3,808	1.71e−04
14	1.11e−06	5	2.63	11.6	25	46.4	7,602	1.04e−04
15	7.17e−07	5	5.30	23.6	50	47.2	15,467	5.43e−05
16	3.18e−07	9	9.09	27.8	100	27.8	18,219	6.68e−06
Reduction rates: $\sigma_1 = 1.040$, $\sigma_2 = 0.805$ ($\delta = 0.4$)								
4	1.17e−03	1	0.0137	0.0111	0.0259	42.9	7	9.19e−03
5	6.09e−04	3	0.0137	0.025	0.0504	49.6	16	8.94e−03
6	3.76e−04	4	0.0153	0.0503	0.0992	50.7	33	6.88e−03
7	1.89e−04	5	0.0214	0.0935	0.197	47.5	61	3.79e−03
8	8.23e−05	8	0.0336	0.137	0.392	34.9	90	2.09e−03
9	3.31e−05	5	0.0687	0.381	0.783	48.7	250	1.11e−03
10	1.48e−05	5	0.142	0.761	1.56	48.8	499	6.60e−04
11	7.52e−06	5	0.287	1.51	3.13	48.2	990	4.14e−04
12	3.82e−06	5	0.592	2.77	6.25	44.3	1,815	2.66e−04
13	2.07e−06	5	1.19	5.35	12.5	42.8	3,506	1.71e−04
14	1.19e−06	5	2.39	10.8	25	43.2	7,078	1.05e−04
15	7.27e−07	5	4.86	21.8	50	43.6	14,287	5.44e−05
16	3.20e−07	10	7.71	23.1	100	23.1	15,139	8.39e−06

to very small coefficients—indicates in both cases that Ad-NIICG is quite conservative since the solution could be compressed further.

In all examples presented so far, the new Ad-NIICG algorithm yielded far lower iteration counts on the highest level and a significantly increased average adaptive efficiency. Therefore we can safely regard Ad-NIICG as superior compared to Ad-OrigCG . We have also observed this for the higher dimensional examples. Hence, all results in the remaining parts of this paper will refer exclusively to Ad-NIICG .

Our last one-dimensional results shown in Table 6 deal with the additional coarsening parameter δ. While a value of $\delta = 0.2$ has been chosen previously, we use these runs to examine the behaviour for $\delta = 0.3$ (top) and $\delta = 0.4$ (bottom). The most obvious effect of these increased coarsening tolerances can be seen in the columns S and N_{av}. They show that the fraction of nonzero wavelet coefficients is reduced by more than a factor of 2 for $\delta = 0.4$. However, the iteration counts per level rise by one, and on the highest level this number jumps to 9 or 10, respectively. We regard this as a warning that higher values of δ affect the stability of the conjugate gradient algorithm in an adverse way up to the situation of failing convergence. In view of the fact that no convergence analysis is available for the inexact CG method in the present situation, we revert to the parameter $\delta = 0.2$ for the subsequent sections of this paper, which has been found safe in all experiments conducted so far.

Table 7 2D, results for right hand sides from top to bottom: $f(x, y) \equiv 1$, $f(x, y) = f_{2,1}(x, y) := f_1(x)$, $f(x, y) := D^{3/2} f_{2,1}(x, y)$ ($\delta = 0.2$)

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
4	0.00e−00	0	0.00771	0.00771	0.0275	28.0	81	6.00e−12
5	0.00e−00	0	0.00771	0.00771	0.104	7.41	81	6.00e−12
6	0.00e−00	0	0.00771	0.00771	0.402	1.92	81	6.00e−12
7	0.00e−00	0	0.00771	0.00771	1.58	0.488	81	6.00e−12
8	0.00e−00	0	0.00771	0.00771	6.29	0.123	81	6.00e−12
9	0.00e−00	0	0.00771	0.00771	25	0.0308	81	6.00e−12
10	0.00e−00	0	0.00771	0.00771	100	0.00771	81	6.00e−12
Reduction rates: $\sigma_1 = 0.529$, $\sigma_2 = 0.358$								
4	8.89e−04	1	0.00771	0.00697	0.0275	25.3	73	6.72e−03
5	6.87e−04	2	0.00771	0.0244	0.104	23.5	256	6.52e−03
6	3.79e−04	4	0.00771	0.0716	0.402	17.8	752	6.23e−03
7	1.91e−04	5	0.00999	0.291	1.58	18.4	3,057	3.95e−03
8	8.85e−05	9	0.0208	0.894	6.29	14.2	9,393	2.00e−03
9	3.33e−05	5	0.0719	4.85	25	19.4	50,955	9.37e−04
10	1.42e−05	5	0.263	17.5	100	17.5	183,859	4.76e−04
Reduction rates: $\sigma_1 = 0.532$, $\sigma_2 = 0.381$								
4	1.04e−03	1	0.00771	0.00721	0.0275	26.2	76	8.19e−03
5	5.71e−04	3	0.00771	0.0215	0.104	20.7	226	7.86e−03
6	2.86e−04	4	0.00771	0.0714	0.402	17.8	750	7.73e−03
7	1.88e−04	6	0.0109	0.269	1.58	17.0	2,826	4.26e−03
8	9.09e−05	5	0.0263	1.21	6.29	19.2	12,713	2.22e−03
9	4.12e−05	9	0.0746	3.46	25	13.8	36,352	1.10e−03
10	1.39e−05	5	0.273	17.5	100	17.5	183,859	4.71e−04

Table 8 2D, right hand side top: $f(x, y) = f_{2,2}(x, y) = f_1(x) f_1(y)$, bottom: $f(x, y) = D^{3/2} f_{2,2}(x, y)$ ($\delta = 0.2$)

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
Reduction rates: $\sigma_1 = 0.498$, $\sigma_2 = 0.270$								
4	6.11e−04	1	0.00771	0.00593	0.0275	21.6	62	4.79e−03
5	5.13e−04	2	0.00771	0.0347	0.104	33.4	365	4.65e−03
6	3.28e−04	3	0.00771	0.136	0.402	33.8	1,429	4.52e−03
7	1.81e−04	5	0.00847	0.50	1.58	31.6	5,253	3.91e−03
8	8.28e−05	7	0.0201	1.87	6.29	29.7	19,647	1.99e−03
9	4.72e−05	5	0.0709	8.71	25	34.8	91,509	1.00e−03
10	1.89e−05	9	0.229	26.4	100	26.4	277,365	4.98e−04
Reduction rates: $\sigma_1 = 0.507$, $\sigma_2 = 0.302$								
4	8.23e−04	1	0.00771	0.00676	0.0275	24.6	71	5.90e−03
5	6.11e−04	2	0.00771	0.0359	0.104	34.5	377	5.76e−03
6	3.45e−04	3	0.00771	0.141	0.402	35.1	1,481	5.66e−03
7	1.88e−04	5	0.0098	0.521	1.58	33.0	5,474	4.09e−03
8	9.28e−05	7	0.0241	1.81	6.29	28.8	19,016	2.14e−03
9	4.71e−05	5	0.087	8.70	25	34.8	91,404	1.07e−03
10	1.86e−05	9	0.273	25.4	100	25.4	266,859	5.04e−04

5.2 2D computations

The results for more than one spatial dimension have been obtained by tensor product wavelet bases on the unit cube. The behaviour of the adaptive scheme is essentially similar to the one-dimensional case. Also in higher dimensions, the error in the H^1 norm is reduced by a factor of 2 between levels. The number of degrees of freedom on the lowest level is $(2^{j_0} + 1)^n$ in n dimensions, which amounts to 81 for our 2D computations. The highest level computed was $j = 10$, allowing for a maximum of over one million unknowns.

The first listing in Table 7 corresponds to the right hand side $f \equiv 1$, yielding a constant solution which is already represented exactly with the minimum amount of degrees of freedom, $N = 81$. The second and third table show the non-smooth examples for $f_{2,1}(x, y) := f_1(x)$ using (32), and its roughened version $D^{3/2} f_{2,1}$ as described in the previous section. Both of these functions are constant along the y direction. Since tensor product wavelets allow dimension-adaptivity, we should expect a low percentage of nonzero coefficients due to

Table 9 3D, top: $f_{3,1}$, bottom: $f_{3,1}^D$ (constant in y, z)

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
Reduction rates: $\sigma_1 = 0.392$, $\sigma_2 = 0.318$								
4	9.58e−04	1	0.0339	0.0348	0.229	15.2	747	2.82e−02
5	7.47e−04	5	0.0340	0.146	1.67	8.74	3,134	5.56e−03
6	3.60e−04	4	0.0340	0.915	12.8	7.15	19,642	5.36e−03
7	1.55e−04	7	0.0384	6.61	100	6.61	141,896	4.06e−03
Reduction rates: $\sigma_1 = 0.396$, $\sigma_2 = 0.323$								
4	1.10e−03	1	0.0339	0.0370	0.229	16.2	794	2.84e−02
5	7.67e−04	5	0.0340	0.155	1.67	9.28	3,327	6.67e−03
6	3.36e−04	4	0.0340	0.981	12.8	7.66	21,059	6.51e−03
7	1.74e−04	7	0.0421	6.69	100	6.69	143,613	4.11e−03

Table 10 3D, top: $f_{3,2}$, bottom: $f_{3,2}^D$ (constant in z)

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
Reduction rates: $\sigma_1 = 0.337$, $\sigma_2 = 0.352$								
4	6.81e−04	1	0.0338	0.0300	0.229	13.1	644	2.82e−02
5	7.25e−04	5	0.0339	0.266	1.67	15.9	5,710	1.30e−02
6	3.41e−04	6	0.0340	1.62	12.8	12.7	34,776	3.73e−03
7	1.73e−04	4	0.0340	15.5	100	15.5	332,737	3.71e−03
Reduction rates: $\sigma_1 = 0.343$, $\sigma_2 = 0.342$								
4	8.09e−04	1	0.0338	0.0344	0.229	15.0	738	2.84e−02
5	7.30e−04	5	0.0339	0.278	1.67	16.6	5,968	1.33e−02
6	3.43e−04	6	0.0340	1.66	12.8	13.0	35,635	4.62e−03
7	1.85e−04	4	0.0368	15.8	100	15.8	339,177	3.93e−03

the smoothness in one direction. Indeed, the number of nonzero coefficients stays below 20%, which is lower by more than a factor of 2 compared to the one-dimensional examples with square-root singular right hand sides in Tables 4 and 5. As in the one-dimensional examples, the roughening operator $D^{3/2}$ has almost no effect on the percentages and iteration numbers.

For the two sets of results listed in Table 8 we have changed the right hand side to $f_{2,2}(x, y) := f_1(x) f_1(y)$ and $D^{3/2} f_{2,2}$, respectively, with f_1 defined in (32). These functions are non-smooth in both x and y directions, so we expect a higher percentage of nonzero coefficients than in the previous example. This is actually confirmed by the column S stating an average of 30%. The iteration counts and errors do not change much compared to the previous example, and again the roughness of the second experiment does not harm the convergence. Compared to the one-dimensional situation, the two-dimensional examples require a larger overall amount of degrees of freedom, while at the same time they seem to take advantage of the adaptive features more thoroughly.

5.3 3D computations

We have arranged the results of the three-dimensional computations in three sets of tables. For the first set listed in Table 9, the right hand side is constant in the y and z directions, i.e., $f_{3,1}(x, y, z) \equiv f_1(x)$, cf. (32). For the right

Table 11 3D, top: $f_{3,3}$, bottom: $f_{3,3}^D$ (depends on x, y, z)

j	$\|\mathbf{r}_j\|$	#It	P (%)	V (%)	M (%)	S (%)	N_{av}	$\varepsilon(\mathbf{y})$
Reduction rates: $\sigma_1 = 0.332$, $\sigma_2 = 0.396$								
4	3.94e−04	1	0.0333	0.0278	0.229	12.1	597	2.98e−02
5	5.77e−04	7	0.0338	0.255	1.67	15.3	5,474	1.36e−02
6	3.36e−04	6	0.0339	2.09	12.8	16.3	44,866	6.28e−03
7	1.83e−04	8	0.0340	14.3	100	14.3	306,977	2.44e−03
Reduction rates: $\sigma_1 = 0.334$, $\sigma_2 = 0.367$								
4	5.01e−04	1	0.0333	0.0293	0.229	12.8	629	2.99e−02
5	5.72e−04	7	0.0338	0.257	1.67	15.4	5,517	1.37e−02
6	3.41e−04	6	0.0339	2.15	12.8	16.8	46,154	6.52e−03
7	1.84e−04	8	0.0340	14.6	100	14.6	313,417	3.00e−03

hand sides in Tables 10 and 11, we have chosen $f_{3,2}(x, y, z) := f_1(x) f_1(y)$ and $f_{3,3}(x, y, z) := f_1(x) f_1(y) f_1(z)$. In all three examples a second table is included corresponding to the roughened version $f_{3,i}^D := D^{3/2} f_{3,i}$. We have conducted computations for the finest level $J = 7$, which allows for up to two million possible coefficients.

It can be observed in all tables that the adaptive efficiency is further increased compared to the two-dimensional examples. Table 9, corresponding to a right hand side which is constant in the y and z directions, shows the largest savings with under 10% of nonzero coefficients. The percentage for the right hand sides $f_{3,2}$ and $f_{3,3}$ is a little higher at about 15%, but consistently lower than all values obtained for the two-dimensional examples. The increasing complexity of all six right hand sided tested in three dimensions does not show a significant effect on the convergence history, since the iteration counts appear to be bounded by a constant and the H^1 errors reduce by a factor of 2 from level to level.

6 Conclusions

Our new scheme Ad-NIICG is favorable over Ad-OrigCG for several reasons:

- The number of iterations on the finest level, which accounts for the largest fraction of computational work, is reduced by a factor between 2 and 4.
- The adaptive implementation is ineffective in Ad-OrigCG due to high fill-in. The nested iteration strategy can take advantage of the adaptive wavelet formulation to reduce the number of nonzero coefficients used in the computations.
- Special cases where most wavelet coefficients of the solution belong to lower levels are accelerated significantly through the good start values computed by the nested iteration approach.

Moreover, a well-preconditioned elliptic operator as provided in the examples is highly recommended to obtain a small number of iterations per level, which reduces the fill-in of very small wavelet coefficients. Our full adaptive nested iteration method Ad-NIICG produces substantial savings in the number of nonzero coefficients compared to full uniform discretizations which becomes more relevant with increasing space dimension: from about 50% in 1D to 75% in 2D and 85% in 3D. Since the wavelet constructions used here are based on tensor products, they are inherently multi-dimensional which offers perspectives for effective adaptivity for even higher dimensional problems.

References

1. Barsch, T.: Adaptive Multiskalenverfahren für elliptische partielle Differentialgleichungen—Realisierung, Umsetzung und numerische Ergebnisse. Shaker (2001)

2. Barinka, A., Barsch, T., Charton, P., Cohen, A., Dahlke, S., Dahmen, W., Urban, K.: Adaptive wavelet schemes for elliptic problems—implementation and numerical experiments. SIAM J. Sci. Comput. **23**, 910–939 (2001)
3. Bramble, J., Cohen, A., Dahmen, W.: Multiscale problems and methods in numerical simulations. Lecture Notes in Mathematics. Springer (2003)
4. Beylkin, G., Coifman, R., Rokhlin, V.: Fast wavelet transforms and numerical algorithms I. Commun. Pure Appl. Math. **44**, 141–183 (1991)
5. Bouras, A., Fraysse, V.: A relaxation strategy for inexact matrix–vector products for Krylov methods. Technical Report TR/PA/00/15, CERFACS, France (2000)
6. Berrone, S., Kozubek, T.: An adaptive wavelet algorithm for solving elliptic boundary value problems in fairly general domains. SIAM J. Sci. Comput. **28**(6), 2114–2138 (2006)
7. Burstedde, C., Kunoth, A.: Fast iterative solution of elliptic control problems in wavelet discretization. J. Comput. Appl. Math. **196**(1), 299–319 (2006)
8. Bramble, J., Pasciak, J., Xu, J.: Parallel multilevel preconditioners. Math. Comput. **37**, 1–22 (1981)
9. Bramble, J.: Multigrid Methods. Pitman (1993)
10. Braess, D.: Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics, 2nd edn. Cambridge University Press (2001)
11. Burstedde, C.: Fast optimised wavelet methods for control problems constrained by elliptic PDEs. Ph.D. Dissertation, Institut für Angewandte Mathematik, Universität Bonn (2005) http://hss.ulb.uni-bonn.de/diss_online/math_nat_fak/2005/burstedde_carsten
12. Cohen, A., Dahmen, W., DeVore, R.: Adaptive wavelet methods for elliptic operator equations—convergence rates. Math. Comput. **70**, 27–75 (2001)
13. Cohen, A., Dahmen, W., DeVore, R.: Adaptive wavelet techniques in numerical simulation. In: Stein, E., de Borst, R., Hughes, T. (eds.) Encyclopedia of Computational Mechanics, vol. 1. John Wiley & Sons (2004)
14. Cohen, A., Daubechies, I., Feauveau, J.: Biorthogonal bases of compactly supported wavelets. Commun. Pure Appl. Math. **45**, 485–560 (1992)
15. Cohen, A.: Numerical Analysis of Wavelet Methods. Elsevier (2003)
16. Canuto, C., Tabacco, A., Urban, K.: The wavelet element method, part i: construction and analysis. Appl. Comput. Harmon. Anal. **6**, 1–52 (1999)
17. Dahmen, W.: Stability of multiscale transformations. J. Fourier Anal. Appl. **4**, 341–362 (1996)
18. Dahmen, W.: Wavelet and multiscale methods for operator equations. Acta Numer. **6**, 55–228 (1997)
19. Dahmen, W.: Wavelet methods for pdes—some recent developments. J. Comput. Appl. Math. **128**, 133–185 (2001)
20. DeVore, R.: Nonlinear approximation. Acta Numer. **7**, 51–150 (1998)
21. Dahmen, W., Kunoth, A.: Multilevel preconditioning. Numer. Math. **63**, 315–344 (1992)
22. Dahmen, W., Kunoth, A., Schneider, R.: Wavelet least square methods for boundary value problems. SIAM J. Numer. Anal. **39**, 1985–2013 (2002)
23. Dahmen, W., Kunoth, A., Urban, K.: Biorthogonal spline-wavelets on the interval—stability and moment conditions. Appl. Comput. Harmon. Anal. **6**, 132–196 (1999)
24. Dahmen, W., Prößdorf, S., Schneider, R.: Multiscale methods for pseudo-differential equations on smooth manifolds. In: Chui, C.K., Montefusco, L., Puccio, L. (eds.) Proceedings of the International Conference on Wavelets: Theory, Algorithms, and Applications, pp. 385–424. Academic Press (1994)
25. Dahmen, W., Schneider, R.: Composite wavelet bases for operator equations. Math. Comput. **68**, 1533–1567 (1999)
26. Dahmen, W., Schneider, R.: Wavelets on manifolds I: construction and domain decomposition. SIAM J. Math. Anal. **31**, 184–230 (1999)
27. DeVore, R., Temlyakov, V.: Some remarks on greedy algorithms. Adv. Comput. Math. **5**, 173–1124 (1996)
28. Gantumur, T., Harbrecht, H., Stevenson, R.: An optimal adaptive wavelet method without coarsening of the iterands. Math. Comp. **77**, 615–629 (2007)
29. Golub, G., Ye, Q.: Inexact preconditioned conjugate gradient method with inner-outer iteration. SIAM J. Sci. Comput. **21**, 1305–1320 (2000)
30. Hackbusch, W.: Elliptic Differential Equations: Theory and Numerical Treatment. Springer (1992)

31. Kunoth, A., Sahner, J.: Wavelets on manifolds: an optimized construction. Math. Comput. **75**, 1319–1349 (2006)
32. Kunoth, A.: Wavelet-based multiresolution methods for stationary PDEs and PDE-constrained control problems. In: Blowey, J., Craig, A. (eds.) Frontiers in Numerical Analysis (Durham 2004), Universitext, pp. 1–63. Springer (2005)
33. Oswald, P.: On discrete norm estimates related to multilevel preconditioners in the finite element method. In: Ivanov, K.G., Petrushev, P., Sendov, B. (eds.) Constructive Theory of Functions, Proc. Int. Conf. Varna, pp. 203–214 (1991)
34. Schneider, R.: Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme. Advances in Numerical Mathematics. Teubner Stuttgart (1998)
35. Stevenson, R.: On the compressibility of operators in wavelet coordinates. SIAM J. Math. Anal. **35**(5), 1110–1132 (2004)
36. Tchamitchian, P.: Wavelets, functions, and operators. In: Erlebacher, G., Hussaini, M.Y., Jameson, L. (eds.) Wavelets: Theory and Applications, Series in Computational Science and Engineering, pp. 83–181. Oxford University Press (1996)
37. van den Eshof, J., Sleijpen, G.: Inexact Krylov subspace methods for linear systems. SIAM J. Matrix Anal. A. **26**(1), 125–153 (2004)
38. von Petersdorff, T., Schwab, C.: Fully discrete multiscale Galerkin BEM. In: Dahmen, W., Kurdila, A., Oswald, P. (eds.) Multiscale Wavelet Methods for PDEs, pp. 287–346. Academic Press (1997)