[6] A. Joshi, W. Bangerth, K. Hwang, J. C. Rasmussen, and E. M. Sevick-Muraca. Fully adaptive fem based fluorescence optical tomography from time-dependent measurements with area illumination and detection. *Med. Phys.*, 33(5):1299–1310, 2006.
[7] A. Joshi, W. Bangerth, K. Hwang, J. C. Rasmussen, and E. M. Sevick-Muraca. Plane wave fluorescence tomography with adaptive finite elements. *Optics Letters*, 31:193–195, 2006.
[8] A. Joshi, W. Bangerth, and E. M. Sevick-Muraca. Adaptive finite element modeling of optical fluorescence-enhanced tomography. *Optics Express*, 12(22):5402–5417, November 2004.
[9] A. Joshi, W. Bangerth, and E. M. Sevick-Muraca. Non-contact fluorescence optical tomography with scanning patterned illumination. *Optics Express*, 14(14):6516–6534, 2006.
[10] R. Rannacher. The dual-weighted-residual method for error control and mesh adaptation in finite element methods. In J.R. Whiteman, editor, *Proc. MAFELAP 1999*, pages 97–116. Elsevier, Amsterdam, 2000.

Adaptive Methods and Near-Best Tree Approximation

Peter Binev

Large scale problems require fast and efficient ways of processing. Computational tractability can often be guaranteed only by methods that are nonlinear in nature and at the same time support efficient data structures. Tree approximation is a paradigm that takes up both issues. It provides an efficient way of storing the coefficients in wavelet expansions while essentially preserving the optimal rate of the N-term approximation. It is also a natural way of representing the partitions generated by adaptive solvers for PDEs.

Previously known results (see [3]) provide optimality of asymptotic rates over a whole class of functions. The corresponding algorithms which realize such an approximation have only theoretical significance. The question is: *Can we achieve instance optimality in the sense of recovering the best N-term rate for each individual function?* To answer this question we outline a specific approximation scheme which achieves just that.

We represent the adaptively generated partitions as proper subtrees of an infinite *master* tree $\mathcal{T}$. The master tree $\mathcal{T}$ is fixed and corresponds to a particular subdivision procedure used in the adaptive process which produces the partitions. Here "proper" means that T corresponds to a partition, namely that it contains the subtree T_0 which corresponds to the initial partition and if a node Δ of T is not a leaf, then the set $\mathcal{C}(\Delta)$ of all its children is contained in T. We assume that at each node Δ of $\mathcal{T}$ it is given an error $e(\Delta)$ which can be calculated using a fixed number of operations. The total error $E(T)$ of the partition corresponding to the tree T can be calculated as the sum of the errors at the leaves $\mathcal{L}(T)$ of T

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

Often in practice it is impossible to calculate the actual error and one has to use some local quantities to represents the local errors $e(\Delta)$. A necessary condition to these quantities is the *subadditivity* property

$$e(\Delta) \geq \sum_{\Delta'\in\mathcal{C}(\Delta)} e(\Delta') \ . \tag{1}$$

Sometimes it is desirable to relax the above condition by requiring that there exists a constant $C_0 > 0$ such that

$$e(\Delta) \geq C_0 \sum_{\Delta' \in \mathcal{L}(T_\Delta)} e(\Delta') \tag{2}$$

for every subtree $T_\Delta \subset \mathcal{T}$ rooted at Δ.

Let N_0 be the number of leaves of the initial tree T_0 and let $N(T)$ be the number of subdivisions performed consecutively to T_0 in order to receive T. Evidently, $N(T) = \#(T) - \#(\mathcal{L}(T)) - \#(T_0) + N_0$ is the number of new internal nodes of T. Using this notation we can introduce the *best n-term tree approximation* by

$$\sigma_n = \sigma_n(\mathcal{T}) := \min_{T: N(T) \leq n} E(T)$$

and ask the question whether there exists an $\mathcal{O}(n)$ algorithm that finds a tree which complexity and error are within (small) multiplicative constants from the ones of σ_n. The answer to this question is given in [1]. The algorithm proposed there defines *modified* errors by the following rules

$$\tilde{e}(\Delta) := e(\Delta) \text{ for } \Delta \in T_0 \qquad \text{and} \qquad \tilde{e}(\Delta) := \frac{\sum_{\Delta'' \in \mathcal{C}(\Delta')} e(\Delta'')}{e(\Delta') + \tilde{e}(\Delta')} \text{ for } \Delta \in \mathcal{C}(\Delta') \,. \tag{3}$$

Then the tree T is received adaptively from T_0 by subdividing at each step the current leaf node Δ with the largest modified error. (Alternatively, to avoid sorting we can group the modified errors into bins based on their values and choose one which is within a small multiplicative constant from the largest.) Here we shall refer to the above algorithm as *old tree algorithm*. The main result for it is the following

Theorem 1 [1] *Let the number of the children at each node of $\mathcal{T}$ be limited by a constant K. Then there exists a constant $C > 0$ such that at each step the output tree T of the* old tree algorithm *satisfies*

$$E(T) \leq C\sigma_n(\mathcal{T})$$

whenever $n \leq N(T)/(2K+2)$. To create T the algorithm uses less than $C(N(T)+N_0)$ arithmetic operations and computations of e.

The constant C in the above theorem is given explicitly and depends on K and C_0. It is interesting to know whether this constant can be arbitrarily close to 1. In the case $C_0 = 1$ we have the following result

Theorem 2 *If $\mathcal{T}$ is a binary tree (i.e. $K = 2$) and the errors $e(\Delta)$ in it satisfy the subadditivity condition (1), then the output tree T of the* old tree algorithm *satisfies*

$$E(T) \le \left(1 + \frac{2(n + N_0)}{N(T) - n + 1}\right)\ \sigma_n(\mathcal{T})$$

whenever $n \le N(T)$.

Using the above theorem the following variant of Corollary 5.4 from [1] can be derived

Corollary 3 *Let $\mathcal{T}$ be a binary tree and the errors $e(\Delta)$ satisfy the subadditivity condition (1). If the tree T_μ is the first tree from the sequence of outputs of* old tree algorithm *that satisfies $E(T_\mu) \le \mu$, then*

$$N(T_\mu) \le \frac{2(1 + c_1)}{1 - c_1}\ N(T)$$

for $0 < c_1 < 1$ and any proper subtree T of $\mathcal{T}$ with $E(T) \le c_1\mu$.

The performance of old tree algorithm can be improved by replacing the definition of the modified errors in (3) with the following one

$$(4)\quad \tilde{e}(\Delta) := e(\Delta) \text{ for } \Delta \in T_0 \quad \text{and} \quad \tilde{e}(\Delta) := \left(\frac{1}{e(\Delta)} + \frac{1}{\tilde{e}(\Delta')}\right)^{-1} \text{ for } \Delta \in \mathcal{C}(\Delta').$$

We shall refer to that algorithm as *new tree algorithm*. The following theorem shows that the constant C no longer depends on K. However, the complexity of the algorithm still depends on K via the total number of nodes in T which can be estimated by $K\,N(T)$.

Theorem 4 *Let the errors $e(\Delta)$ in $\mathcal{T}$ satisfy the subadditivity condition (1). Then at each step of the* new tree algorithm *the output tree T satisfies*

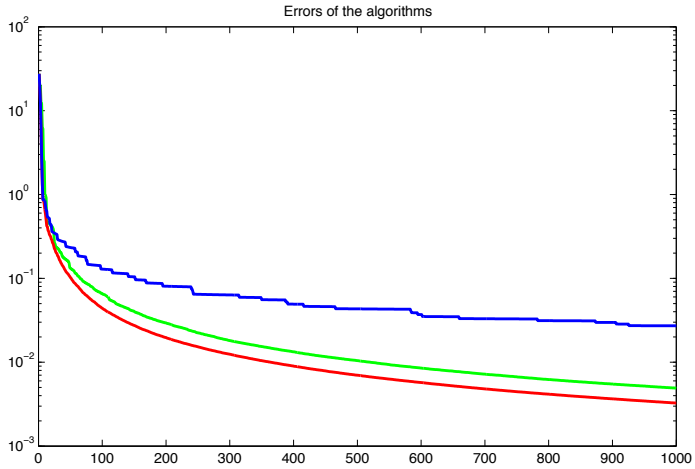
$$E(T) \le \left(1 + \frac{n + \min\{n, N_0\}}{N(T) - n + 1}\right)\ \sigma_n(\mathcal{T})$$

whenever $n \le N(T)$.

To illustrate the fact that the greedy approach cannot give a near-best performance, we consider a simple example of finding the best L_2-approximation via piecewise constants on dyadic partitions for the function $f(x) = (\sqrt{x}\ln(x/1.02))$. For this example the performance of the greedy algorithm is worse by a logarithmic factor.

In the figure the errors of the described algorithms for the first 1000 iterations are presented: the greedy algorithm is the top graph (in blue), in the middle (in green) is the algorithm proposed in [1], while the best performance has the new tree algorithm represented by the bottom graph (in red).

We conclude with indicating some consequences of the results about near-best tree approximation. In particular, we compare the adaptive schemes for solving elliptic PDEs and consider the conditions needed to establish instance optimality of the solution. The overwhelming majority of these methods is based on local



error estimates which sum gives reliable information about the total error. The method is often described with the iterated scheme

$$\text{MARK} \rightarrow \text{REFINE} \rightarrow \text{SOLVE} \quad \hookleftarrow . \tag{5}$$

The most popular among the marking strategies is the *bulk chasing* [4]. However, these strategies apply the greedy approach in choosing the elements for subdivision which does not give much hope for instance optimality results. A convergence result together with an error reduction property was first derived in [5] for a variant of (5) with the bulk chasing strategy in the case of Poison equation and the newest vertex bisection subdivision procedure. Optimal convergence rates were first received in [2] based on the error reduction property from [5]. The paper [2] establishes the theoretical justification of all the steps in the algorithm. It also shows that the sparsity of the solution is necessary for the optimality of the algorithm and to secure it adds the COARSENING step to (5). The inclusion of this step gives the freedom to use any error reduction procedure (and in particular the one consisting of several loops of (5)). Then, the obtained approximation should be checked for sparsity and a certain action should be taken so that the approximate solution will stay sparse. In general, the idea of the algorithm can be summarized in the scheme

$$\text{ERROR REDUCTION} \rightarrow \text{SPARSITY ADJUSTMENT} \quad \hookleftarrow . \tag{6}$$

This approach is able to give instance optimality results in the following sense: given the differential equation $Lu = f$, if the differential operator L and the function f are approximated with tolerance $\varepsilon > 0$, then the approximate solution is near-best or the error is below ε. In comparison, it was shown in [6] that the optimal rates of convergence can be achieved using a variant of the bulk chasing method with carefully selected (small) bulk constant. In some sense this method combines the steps in (6). However, its optimality is only in terms of asymptotic behavior. Thus, the performance can be very poor in terms of an individual functions and a fixed number of degrees of freedom. Note that in the example for tree approximation the greedy approach also has optimal rates, but the constants are growing as $\log n$.

In conclusion, the tools provided by near-best adaptive tree approximation can be used to obtain methods fitting the scheme (6) which can give instance optimality results for a wide range of problems.

References

[1] P. Binev, and R. DeVore, *Fast Computation in Adaptive Tree Approximation*, Numer. Math. **97** (2004), 193–217.

[2] P. Binev, W. Dahmen, and R. DeVore, *Adaptive Finite Element Methods with Convergence Rates* , Numer. Math. **97** (2004), 219–268.

[3] A. Cohen, W. Dahmen, I. Daubechies, R. DeVore, *Tree approximation and optimal encoding*, Appl. Comp. Harm. Anal. **11** (2001), 192–226.

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

[5] P. Morin, R. Nochetto, and K. Siebert, *Data Oscillation and Convergence of Adaptive FEM*, SIAM J. Numer. Anal. **38** (2000), 466–488.

[6] R. Stevenson, *Optimality of a standard adaptive finite element method*, Found. Comput. Math. **7** (2007), 245–269.

Unified a posteriori error analysis of nonstandard finite element methods

Carsten Carstensen

General strategies are discussed to derive a posteriori error estimates for conforming, mixed, and nonconforming finite element methods in energy norms which also cover discontinuous Galerkin schemes or Mortar finite elements for second order elliptic problems. The unifying approach provides reliable error estimates which can be shown to be efficient as well. The goal is to prove that *all* nonstandard schemes allow for (some) a posteriori error control, there is no finite element method known to the author where there is no error control.

Surprisingly, there remains one type of residuals R for different problems, such as, the Laplace problem, the Stokes problem, and Navier-Lamé problem, with conforming, nonconforming, and mixed finite element method. One key observation is that

$$\mathrm{Res}(v) := \int_\Omega g \cdot v dx + \int_{\cup\mathcal{E}} g_\mathcal{E} \cdot v ds \quad \text{for } v \in V := H^1_0(\Omega;\mathbb{R}^m)$$

is the same (or at least very similar) for all those schemes. Based on some Galerkin property, the kernel of the residual $\mathrm{Res} \in V^* = H^{-1}(\Omega;\mathbb{R}^m)$ includes a space V_h^{NC}. This space V_h^{NC} can be mapped onto some first- order conforming finite element space $V_h^C \subset V := H^1_0(\Omega;\mathbb{R}^m)$. In all cases the author is aware of, there exists some operator $\Pi : V_h^C \to V_h^{NC}$ with some elementwise properties such that a unifying analysis is possible.

Some nonconforming elements [1,2,3] for the Poisson problem, the Stokes problem, and linear elasticity in two dimensions are depicted in the following tables. The point is that the analysis simultaneously discusses all of those (and even more).