Appl. Comput. Harmon. Anal. 37 (2014) 52–68

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Efficient adaptive operator application on wavelet expansions

J. Houdayer *

Institut de Physique Théorique, CEA, F-91191 Gif-sur-Yvette, France

ARTICLE INFO

Article history:
Received 14 September 2012
Received in revised form 11 June 2013
Accepted 18 September 2013
Available online 25 September 2013
Communicated by Albert Cohen

Keywords:
Wavelets
Partial differential equations
Adaptive wavelet methods
Numerical stability

ABSTRACT

One key step in solving partial differential equations using adaptive wavelet methods is the ability to efficiently apply an operator to a wavelet expansion. Whereas this problem has been generally solved in theory, the known solution is still a little slow and hard to implement. Here, we propose a more practical algorithm for a useful set of linear operators containing in particular all linear differential operators. Our algorithm is *general* as it works for many wavelet systems. It is *fast* as it is linear with a small constant factor. It is *exact* as coefficients are computed without approximation. It is *simple* since the matrix entries of the operator need to be known only for wavelets at the same scale.

1. Introduction

1.1. Background

Since a few years, there has been a lot of progress made on the use of wavelets for solving partial differential equations (PDE). A key step has been the article by Cohen et al. [1] which shows that solving such PDE using adaptive wavelet methods was possible and efficient. For an overview of the subject, see [2–4].

One of the key steps of the method (called "APPLY") computes the result of the adaptive application of the operator on some wavelet expansion. For this to work, they devised an approximation scheme able to compute an N-term expansion of the result in time $O(N)$ with sufficient accuracy. They supposed that matrix elements of the operator could be computed in time $O(1)$. This was later proven by Gantumur and Stevenson [5] for differential operators on piecewise polynomial wavelets (and generalizations thereof).

While this is a big step forward, there are some difficulties left. Here, we want to deal with three of them: (i) the prefactor in $O(N)$ is large (see below), (ii) the Gantumur result does not apply to all wavelet systems such as Daubechies wavelets for which one must rely on the refinement equation to compute matrix elements, (iii) the algorithm is non-local on the wavelet coefficients tree, which slows and complexifies the implementation (the "curse of data structure" of A. Cohen in [2, p. 318]). This last point is particularly true with sophisticated wavelet systems such as domain adapted wavelets [6–8].

An interesting progress was made by Barinka et al. [9] by cleverly using the refinement equation in a very general setting. Here, we develop on this idea in a slightly less general setting. For a restricted set of linear operators, which we call finite width operators (which nevertheless contains among others all linear differential operators), we devised an algorithm to compute the coefficients of the adaptive application of the operator to a wavelet expansion. Here, we are only focused on the way to compute those coefficients, *not* on the way to choose which coefficients to compute. This algorithm is general, fast, fully exact as no approximations are made, and relatively easy to implement and use. One of the key ingredients is the

* Fax: +33 1 69 08 81 20.
E-mail address: jerome.houdayer@cea.fr.

1063-5203/$ – see front matter
http://dx.doi.org/10.1016/j.acha.2013.09.004

fact that the matrix entries need to be known only level by level. That is, we do not need the matrix entries for wavelets at different scales.

1.2. Road map

In the first section, we set up a very general framework in a similar way as Barinka et al. [9] and make some definitions, in particular we define what we call finite width operators.

In the following section, we present three algorithms. The first is inspired by [9], it is fast and exact but only works on a restricted set of operators. It is moreover strongly unstable in some cases. Here, it is important to note, that one is confronted to the same kind of instability when directly computing entries of the operator matrix at very different wavelet scales using the refinement equation.

To cure these problems, we develop another algorithm which is still exact, faster, more stable and works on our full set of finite width operators. We think this algorithm should become the standard in the future. Unfortunately, this algorithm is also unstable in some extreme cases.

We try to cure this instability in a third experimental algorithm. This algorithm tracks an estimated error with each computed quantity and neglect them when too much signal is lost.

In the last section, we present some numerical experiments with examples of instability. We show how our second algorithm outperforms the first and how our last algorithm is indeed able to prevent instability in some cases.

2. Wavelets, expansions and operators

2.1. Wavelets

There are many ways to build a system of wavelets. In order to stay general, we will not commit to one system, but we will rather state properties we want our wavelet system to possess. Most relevant wavelet systems have the properties we describe below. This includes for example: (i) $[0,1]$ with periodic boundary conditions equipped with Daubechies wavelets or bi-orthogonal spline wavelets, (ii) $[0,1]$ with open boundaries with Daubechies or spline wavelets, with edge wavelets at the extremities, (iii) any product of the previous cases, (iv) more involved constructions to adapt wavelets on more general domains such as [6–8].

We will work in a finite domain $\Omega \subset \mathbb{R}^d$ equipped with a bi-orthogonal system of compact wavelets on Ω with the following properties.

2.1.1. Structure of the wavelet system

We start with an infinite index set $\mathcal{I}$ to represent the scale and position of the wavelets. Every $\lambda \in \mathcal{I}$ has a *level* noted $|\lambda| \in \mathbb{N}$ and we note

$$\mathcal{I}_j = \{\lambda \mid \lambda \in \mathcal{I},\ |\lambda| = j\}, \tag{1}$$

and we suppose that all $\mathcal{I}_j$ are finite sets of cardinal $\sharp\mathcal{I}_j$ with

$$c2^{jd} \leqslant \sharp\mathcal{I}_j \leqslant C2^{jd}, \tag{2}$$

where c and C are strictly positive constants (independent of j). To simplify notations, we will rewrite such constraints by rewriting $x \leqslant Cy$ for some constant C as $x \lesssim y$ and if $x \lesssim y$ and $y \lesssim x$ then we write $x \sim y$. Moreover, to shorten many equations, we also set

$$\alpha = 2^{\frac{d}{2}}. \tag{3}$$

We thus have

$$\sharp\mathcal{I}_j \sim \alpha^{2j}. \tag{4}$$

Finally we note

$$\mathcal{I}_j^\infty = \bigcup_{j' \geqslant j} \mathcal{I}_{j'}. \tag{5}$$

Every $\lambda \in \mathcal{I}_j$ (with $j > 0$) has a unique *parent* $\lambda^\downarrow \in \mathcal{I}_{j-1}$. We note $\mu \prec \lambda$ when μ is an *ancestor* of λ (i.e. $\mu = \lambda^\downarrow, \lambda^{\downarrow\downarrow}, \ldots$) and $\mu \preccurlyeq \lambda$ when equality is possible. Likewise we note $\lambda^\uparrow$ the set of *children* of λ and we suppose that this set is finite with

$$\sharp\lambda^\uparrow \sim \alpha^2. \tag{6}$$