Implementation of Complex Wavelet representation of background error covariances in ALADIN CY35
Report of work done in Toulouse, 29/06-10/07/2009

Alex Deckmyn
Royal Meteorological Institute of Belgium, Brussels, Belgium
e-mail: alex.deckmyn@oma.be

Abstract

This report describes the implementation of a complex wavelet Jb in ALADIN. The work is a continuation of the work that started in 2007 at the Hungarian Meteorological Service and continued during several stays at Météo France.

The 3d-Var code of ALADIN (CY35 bf1) was adapted to a complex wavelet representation of the (unbalanced part of the) background error covariances. During the 2009 stay, some new theoretical results were obtained (non-periodic boundaries) and also the technical implementation was further debugged.

Contents

1 Introduction

The work during the present two-week stay (29 June – 10 July 2009) can be divided into 2 parts. On the one hand, the theoretical background of the complex wavelet transforms was further developped. We can now replace the periodic boundary conditions by **symmetric boundary conditions**. This means we can avoid artificial continuation of structure functions at the opposite side of the domain. This will be explained in the next section.

Using the periodic boundary conditions that were implemented last year, I also continued validation and testing of the present code.

2 The complex wavelet transform: adding symmetric boundary conditions

This wavelet transform was originally proposed by Kingsbury [2, 3]. A very detailed overview can be found in [4].

In 1D, the transform actually consists of 2 separate orthogonal wavelet transforms. By a careful choice of the wavelet filter, these 2 transforms can be considered as the real and imaginary part of a complex wavelet (although this is in fact only approximately correct). In a very simplified form, one may think of these 2 wavelets as being "shifted by half a grid interval".

There are several possible choices of wavelets that give such a **Dual Tree Complex Wavelet Transform**. In 2 existing (matlab) implementations, the following choices are made:

1. Periodic boundary conditions, orthogonal wavelets at all scales.
2. Symmetric boundary conditions, bi-orthogonal wavelets at the smallest scale, orthogonal at larger scales.

For our work, we much prefer to have orthogonal basis functions. One reason is that the positiveness of $\mathbf{B}$ is only preserved under orthogonal transforms. However, it turns out that it is possible to have both symmetric boundaries and orthogonal wavelets at all scales.

Both the above implementations rely on a first stage transform where the real and imaginary part are the same wavelet *shifted by 1 grid point*. However, I propose to use a transform where the "real" wavelet is the *reverse* of the "imaginary" wavelet. If this wavelet is **almost symmetric**, the difference is very small.

Many basic applications implement periodic boundaries for the wavelet transform. These are simple to code, and there are many possible wavelets that form orthogonal bases. However, to have a perfectly invertible transform under symmetric (mirror) boundary conditions, the wavelet itself must be (anti-)symmetric. It has been proven that such wavelets can never be orthogonal, except for the very basic Haar wavelet. Therefore, one usually has to resort to bi-orthogonal wavelets when using symmetric boundaries.

With dual tree transforms, we can use the fact that we have 2 transforms (so we have a redundant transform, and the wavelets form what is called a *tight frame*). If one filter is the exact reverse of the other, the combination of both makes it possible to have "perfect reconstruction" with symmetric boundary conditions.

A simple way to understand this (and in fact also a possible way to implement it) is by considering the double domain $x_1 \dots x_N x_N \dots x_1$. This is a periodic domain. If we now apply any wavelet filter to this domain, the transform of the mirror part is identical to the transform of teh original domain using an reversed filter. In the case of an almost-symmetric filter, the reversed filter is almost identical to the original, up to a shift of 1 grid point. So it can be interpreted as the imaginary part of a compex wavelet.

In [1], a general approach is given to construct orthogonal wavelets that are almost (anti-)symmetric. One of the examples is exactly suited for our needs. This filter is relatively short, which does lead to some small distortions at the first few stages of the transform, but I think this a very minor effect (and common to all symmetric filters). The anti-symmetric (even-length) wavelet that can be used with periodic boundaries, gives smoother results, but the difference becomes invisible after about 4 scales.

The calculations in [1] were re-done to have more precise wavelet coefficients (this required working with multiple-precision mathematical software). The fully symmetric complex wavelet transform was coded in fortran with an R interface for testing. It was not yet implemented in the most efficient way (like the previous periodic version).

3 Implementation issues & experiments

Several coding errors were corrected and new experiments were run. Also the external access to Olive was fixed, so that experimental work can now be continued from outside Météo France.

3.1 Fixing coding errors

The implementation in Aladin 3d-Var was not yet working correctly. During my stay (and in the weeks directly following), several problems were fixed:

- Correction of a factor 2 error in the adjoint transform, which caused the minimisation to use many more iterations before stabilising.
- Fixes in numerical precision of some parameters.
- Fixes for very small variances ($< 10^{-16}$) in some scales. Using 10^{-15} as an error margin is obviously wrong in such cases.

3.2 Fourier noise

The main issue remaining was that the analysis increments of 1-obs experiments were very noisy. The origins of this noise were not immediately clear, but it was suspected that it was related to the transform between Fourier and wavelet space (via grid point space). The great problem is that at this stage, the wavelet domain size must have several powers of 2 (e.g. $320 = 2^6 * 5$) and so may be different to the original domain size in spectral space (e.g. 300). The conversion at every iteration causes Gibbs-type noise.

No final solution for this problem was found. However, a new data ensemble was created with domain size $320 = 2^8$ that is suitable for both coordinate bases. The data was created from an existing set off differences for Aladin-France, by taking a larger extension zone (31 in stead of 11) and re-running the bi-periodicisation (biper).

When running wavelet 3d-Var on this adapted dom,ain, the noise is gone. This clearly showes that it is indeed due to the different domain sizes (and the resulting padding at the borders). So at this stage **we can avoid the problem, but have no full solution**.

TThe 320**y**320 domain can now be used for a whole series of 1-obs experiments. The first results look realistic and give us hope that the code is more or less stable and bug-free. But further validations must now be started.

4 Discussion

The work is not finished. While using a larger extension zone (and thus re-running FESTAT as well) eliminates (almost) all the Gibbs noise, this is not an ideal solution for the future. However, it will allow us to at last run realistic experiments with the Wavelet 3d-Var implementation.

Single-observation experiments for various parameters and location (e.g. temperature, humidity at various heights and grid locations) have now been initiated and will continue in the future. This can lead to some publications.

Some further topics that were discussed during the stay and may be revisited in the future:

- Is it possible to have odd-sized domains (or, more general, to avoid the powers of 2 requirements of the wavelet transform)? Kingsbury offers the solution of repeating 1 boundary point every time the domain size is odd at a certain scale. This is a possibility, but data organiosation becomes more difficult and also it is not clear how to treat this in the adjoint code.
- Should the (enlarged) extension zone be changed? For the wavelet transform it now doesn't have to be periodic any more, so it may be used to e.g. *improve the behaviour of gradients at the border.* **However, if we were to use non-periodic boundary conditions with the current code, we could re-introduce the Gibbs-noise!**

References

[1] M. F. Abdelnour. *Wavelet design using Gröbner basis*. PhD thesis, Polytechnic Institute, New York University, 2003.

[2] Nick Kingsbury. The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters. In *Proc. 8th IEEE DSP Workshop, Bryce Canyon*, 1998.

[3] Nick Kingsbury. Complex wavelet analysis for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10:234–253, 2001.

[4] Ivan W. Selesnick, Richard G. Baraniuk, and Nick G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, pages 123 – 151, November 2005.