Wavelet compression for unsteady CFD data

M. Giles
Oxford University Computing Laboratory

October 7, 1997

Abstract

This report presents an introduction to the ideas of wavelet compression for unsteady CFD data and gives an overview of the MSc dissertation written by Graham Berridge on this topic with sponsorship from Rolls-Royce.

In summary, wavelet compression has a number of advantages relative to the Fourier compression which has been used previously. The wavelet transforms for a periodic time series have been implemented in FORTRAN, and this code could be used as the basis for future work in compressing the periodic data arising from 3D stator/rotor and flutter applications.

1 Fourier compression

Suppose one has data f_n at N equally spaced intervals in time, and that the data is periodic so that one may write

$$f_n = f_{n+N}.$$

If N is even, this data can be represented by a combination of cosine and sine functions,

$$f_n = \sum_{m=0}^{\frac{N}{2}} a_m \cos\left(\tfrac{2\pi mn}{N}\right) + \sum_{m=1}^{\frac{N}{2}-1} b_m \sin\left(\tfrac{2\pi mn}{N}\right),$$

where

$$a_m = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} f_n \cos\left(\frac{2\pi mn}{N}\right), & m = 0, \frac{N}{2} \\ \frac{2}{N}\sum_{n=1}^{N} f_n \cos\left(\frac{2\pi mn}{N}\right), & m = 1, 2, \ldots, \frac{N}{2}-1 \end{cases}$$

$$b_m = \frac{2}{N}\sum_{n=1}^{N} f_n \sin\left(\tfrac{2\pi mn}{N}\right), \quad m = 1, 2, \ldots, \tfrac{N}{2}-1.$$

In unsteady CFD applications, f_n is the flow data at a particular grid point. If the data consists simply of a mean component plus unsteadiness at the fundamental frequency, the Fourier transform will result in non-zero values for a_0, a_1, b_1; the remaining coefficients will be zero and so need not be stored. More generally, the idea of Fourier compression is to first compute the full set of a_m, b_m coefficients and then decide how many of these need to be retained in order to reconstruct the original signal to within an acceptable accuracy.

For subsonic flow applications in which only the first two harmonics need to be stored, the compression is excellent, resulting in much smaller disk files. For example, if $N = 50$ then a 10:1 compression ratio would be achieved. However, in transonic flow applications with shocks, there are discontinuities in the unsteady data at a particular grid point due to the passing of shocks. This results in a very full Fourier spectrum (i.e. most of the a_m, b_m are significant) and so the compression ratio is very poor.

There are other problems as well due to the global nature of the cosine/sine functions. The entire data is required before one can calculate any of the Fourier coefficients. Therefore, the Fourier compression can only be performed as a post-processing step. Also, the reconstruction at a particular instant in time requires all of the Fourier coefficients. If one considers the Fourier coefficients needed for each point in an entire 3D CFD grid, the memory of the workstations being used for visualisation will usually be inadequate. Therefore, the reconstruction cannot be performed 'on-the-fly' as part of the visualisation. Instead, it would be necessary to re-construct the data by working on one piece of the grid at a time, storing the reconstructed data in a disk file, and then perform the visualisation by reading back in the reconstructed data as needed.

It was the above concerns which led to the MSc project by Graham Berridge to investigate the use of wavelets to transform and compression periodic time series data.

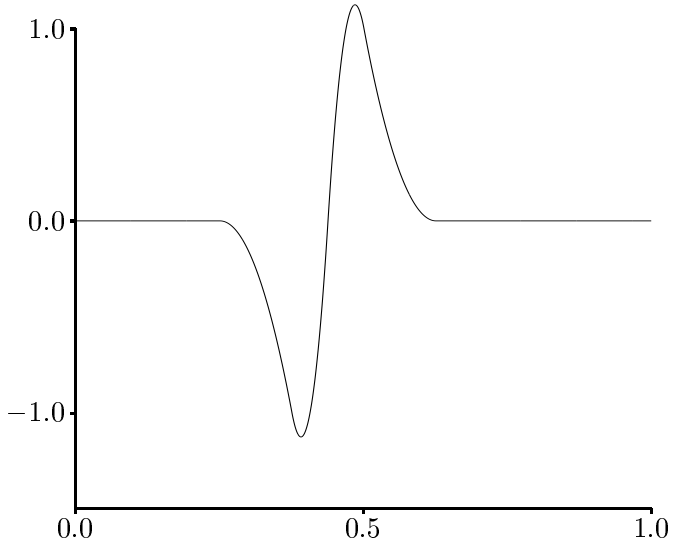

Figure 1: CDF 3,1 wavelet

2 Wavelet compression

2.1 General ideas

The Fourier transform and its inverse reconstruction are a particular example of a general transform pair which may be written as

$$f_n = \sum_{m=1}^{N} a_m w_{n,m}, \quad a_m = \sum_{n=1}^{N} f_n \widetilde{w}_{m,n}.$$

Using matrix notation, these can be written as

$$f = Wa, \quad a = \widetilde{W} f,$$

which reveals that $\widetilde{W} = W^{-1}$. Hence $W\widetilde{W} = \widetilde{W}W = I$ and so

$$\sum_m w_{n1,m} \widetilde{w}_{m,n2} = \begin{cases} 1, & n1 = n2 \\ 0, & n1 \neq n2 \end{cases}$$

$$\sum_n \widetilde{w}_{m1,n} w_{n,m2} = \begin{cases} 1, & m1 = m2 \\ 0, & m1 \neq m2 \end{cases}$$

In wavelet transforms, the function $w_{n,m}$ for a particular value of m is a 'wavelet' which is non-zero on only part of the period, as illustrated in Figure 1. Furthermore, the wavelets for different values of m have a variety of widths, and are centred on different parts of the period. The narrow wavelets represent the high-frequency part of the data; the wide wavelets represent the low-frequency components.

The dual wavelet $\widetilde{w}_{m,n}$ is non-zero on the same interval as the primary wavelet $w_{n,m}$. Thus, the local nature of wavelets means that a discontinuity in the data only affects the amplitudes

a_m of the relatively few wavelets whose span includes the discontinuity. This is the feature which makes wavelets better suited to handling data which is discontinuous.

The local nature of wavelets also makes it possible to perform 'on-the-fly' compression and reconstruction, but to understand how this is done one needs to understand the recursive way in which wavelet transforms are applied in practice. This is explained first for the simple Haar wavelets, before then discussing the CDF 3,1 wavelets which Graham Berridge found to be most effective of those he tested.

2.2 Haar wavelets

For simplicity it will be assumed that N is a power of 2. Starting with the original data f_n one defines the transformed data as a half of the sum and difference of each pair of values,

$$\begin{aligned} s_n &= \tfrac{1}{2}(f_{2n}+f_{2n-1}), \quad n=1,2,\ldots,N/2, \\ d_n &= \tfrac{1}{2}(f_{2n}-f_{2n-1}), \quad n=1,2,\ldots,N/2. \end{aligned}$$

The inverse transform, reconstructing the original data, is given by

$$\begin{aligned} f_{2n} &= s_n + d_n, \quad n=1,2,\ldots,N/2, \\ f_{2n-1} &= s_n - d_n, \quad n=1,2,\ldots,N/2. \end{aligned}$$

The full wavelet transform comes from applying the above recursively, treating the $N/2$ values s_n as the function values at the next level, again computing one half of the sum and difference of each pair of values.

Using the notation $s_n^{(m)}, d_n^{(m)}$ to denote the appropriate values at level m, with the level 0 values being the original data,

$$s_n^{(0)} \equiv f_n,$$

then the values at the higher levels are defined recursively by

$$\begin{aligned} s_n^{(m)} &= \tfrac{1}{2}(s_{2n}^{(m-1)}+s_{2n-1}^{(m-1)}), \\ d_n^{(m)} &= \tfrac{1}{2}(s_{2n}^{(m-1)}-s_{2n-1}^{(m-1)}). \end{aligned}$$

Note that what is stored at each level are the values of the differences $d_n^{(m)}$. The number of these is halved at each level until finally at the topmost level, $M = \log_2 N$, there is a single difference coefficient $d_1^{(M)}$ and a single summation coefficient $s_1^{(M)}$ which must also be stored.

The inverse transform is given recursively by

$$\begin{aligned} s_{2n}^{(m)} &= s_n^{(m+1)} + d_n^{(m+1)}, \\ s_{2n-1}^{(m)} &= s_n^{(m+1)} - d_n^{(m+1)}, \end{aligned}$$

starting with $d_1^{(M)}$ and $s_1^{(M)}$ and working downwards to $m=0$, adding in the contributions from the difference coefficients at each level.

Although this wavelet transform is very simple to apply, it is not very effective in compressing the data. If f_n is a constant, then the difference values will be zero at all levels and need not be retained. On the other hand, if f_n is linear then the difference coefficients will be non-zero at all levels and will probably have to be kept. The CFD 3,1 wavelets described in the next section

have the property that if f_n is locally quadratic the difference coefficients will be zero, and so need not be kept. This makes it much more effective than the Haar wavelets in generating a lot of very small difference coefficients which can be omitted in the compression phase.

2.3 CDF 3,1 wavelets

As with the Haar wavelets, the CDF 3,1 wavelets are applied in a recursive manner. Unlike the Haar wavelets, the coefficients for the CDF 3,1 wavelets at each level of recursion depend on 4 values at the level below.

$$\begin{aligned} s_n^{(m)} &= \sum_{k=0}^{3} \tilde{h}_k s_{2n-k}^{(m-1)}, \\ d_n^{(m)} &= \sum_{k=0}^{3} \tilde{g}_k s_{2n-k}^{(m-1)}. \end{aligned}$$

The recursion is applied until there are at most two elements on the top level, at which point the $s_n^{(m)}$ values are stored in addition to the $d_n^{(m)}$ coefficients.

The inverse transform is given by

$$\begin{aligned} s_{2n}^{(m)} &= h_0 s_n^{(m+1)} + h_2 s_{n+1}^{(m+1)} + g_0 d_n^{(m+1)} + g_2 d_{n+1}^{(m+1)}, \\ s_{2n-1}^{(m)} &= h_1 s_n^{(m+1)} + h_3 s_{n+1}^{(m+1)} + g_1 d_n^{(m+1)} + g_3 d_{n+1}^{(m+1)}. \end{aligned}$$

The elements of the vectors $\tilde{g}, \tilde{h}, g, h$ are

$$\begin{aligned} \tilde{g} &= \{-\tfrac{1}{8}, \tfrac{3}{8}, -\tfrac{3}{8}, \tfrac{1}{8}\} \\ \tilde{h} &= \{-\tfrac{1}{4}, \tfrac{3}{4}, \tfrac{3}{4}, -\tfrac{1}{4}\} \\ g &= \{\tfrac{1}{2}, \tfrac{3}{2}, -\tfrac{3}{2}, -\tfrac{1}{2}\} \\ h &= \{\tfrac{1}{4}, \tfrac{3}{4}, \tfrac{3}{4}, \tfrac{1}{4}\} \end{aligned}$$

By explicitly writing out the matrices W and $\widetilde{W}$ (as defined earlier) for a single level of the recursive process, it is possible to confirm that $W\widetilde{W} = \widetilde{W}W = I$ and so the inverse transform does indeed restore the original data.

Since the elements of the vector $\tilde{g}$ correspond to the coefficients of a third difference operator, it can be seen that the difference coefficients will be zero if the underlying data is locally quadratic. More generally, the difference coefficients at the first level will be proportional to

$$\Delta t^3 \frac{d^3 f}{dt^3},$$

which will be very small if the data is smooth and the timestep is small. Thus it is likely that a high degree of compression will be possible.

Threshold	Sawtooth	Sinusoidal
0.01	22	8
0.001	23	16
0.0001	25	44

Table 1: Compressed number of wavelet coefficients for 2 test cases

3 Compression results

A FORTRAN program has been written to perform the CDF 3,1 wavelet transformation and reconstruction. The compression capability is assessed here by considering two simple test cases, a sawtooth wave which is a combination of a linear function plus a single discontinuity, and a single sinusoidal wave. For each function, 1024 data values were generated. Table 1 lists for each test case the number of wavelet coefficients exceeding certain threshold values; these threshold levels correspond approximately to the error in the reconstructed data. Figure 2 presents for the sawtooth wave the original data and the reconstructed data using the 0.01 threshold; Figure 3 shows the corresponding results for the sinusoidal test function. In both cases, the reconstruction error is barely visible.

For the sawtooth test case, theory shows that the number of non-zero wavelet coefficients should be no more than $3 \log_2 N$. The numerical results are consistent with this. The compression this provides is excellent, especially in comparison to Fourier transforms which offer almost no compression.

In the sinusoidal test case, the number of coefficients which need to be retained is strongly dependent on the accuracy required. For 1% accuracy, which is quite sufficient for plotting and visualisation, only 8 coefficients are needed. This is very good but not as good as Fourier compression in which only 1 coefficient would be needed. If the phase of the sinusoidal wave is shifted by an arbitrary amount, and a constant is added, the number of non-zero Fourier coefficients would increase to 3, while the wavelet compression would still require 8 coefficients. Thus, the wavelet compression would still be worse than the Fourier compression, but not by such a large margin.

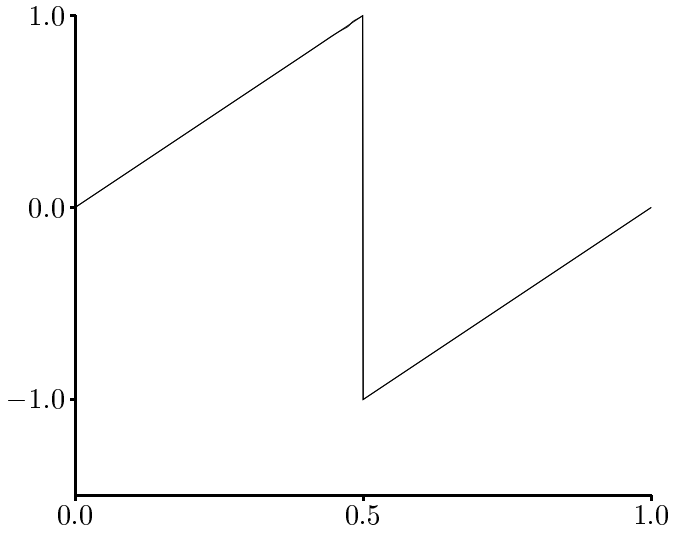

Figure 2: Sawtooth test case – original + reconstruction with 1% threshold

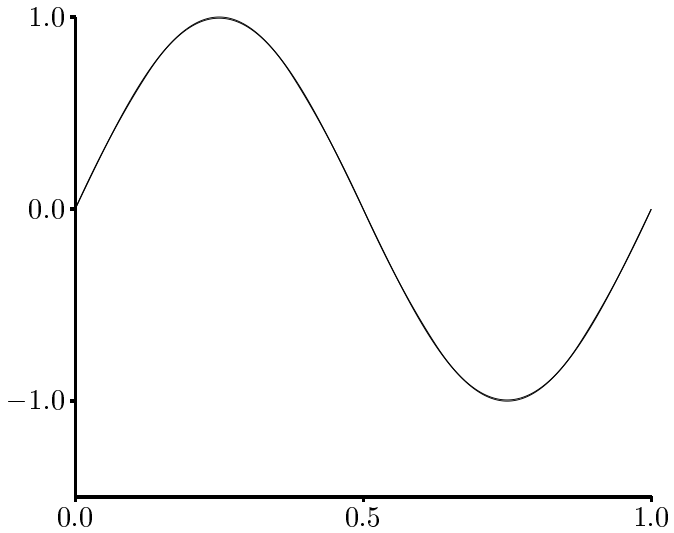

Figure 3: Sinusoidal test case – original + reconstruction with 1% threshold

4 Overview of MSc dissertation

In his dissertation Berridge presents a comprehensive introduction to the theory of wavelets, with a substantial list of references. He investigated 14 different wavelets to determine their effectiveness in compressing some test data consisting of a combination of two sinusoidal functions plus several discontinuous 'sawtooth' functions. On the basis of these tests, he found that the CDF 3,1 wavelet was the most effective.

As a final test, he applied the wavelet compression to some unsteady CFD data. This data came from a transonic wake/rotor interaction in which there were propagating shocks, and so Fourier compression was able to reduce the data by only 10%. The wavelet compression reduced it by 20%, which was better but still disappointing. Berridge attributes this to the fact that the data consisted of only 50 timesteps per period. The data had such a rich content (i.e. there was substantial variation throughout the period) that many coefficients were needed to represent that. His belief is that if the number of stored timesteps per period were increased, then the compression ratio would increase because the number of coefficients needing to be retained would not increase linearly with the total number of timesteps stored. This suggests that one should perhaps store more timesteps per period and thereby substantially reduce the error due to lack of temporal resolution at the expense of only slightly greater storage. This increased storage could be offset by storing information on a coarser grid, the error due to the loss of spatial resolution being less than the error due to the limited temporal resolution.

Berridge also discusses another approach to compressing data storage. CFD codes now often use 64-bit arithmetic, and by default will write unformatted files with 64-bit data. This is an unnecessary level of precision. If one stores the minimum and maximum values, the data can then be renormalised to lie between 0 and 1. Multiplying by 2^{16} and rounding to the nearest integer gives an integer value which can be stored in 16-bits. If necessary, two such numbers can be combined in a single 32-bit integer. This gives a compression ratio of 4:1 with a negligible loss of accuracy since the fractional error is $2^{-16} \approx 10^{-5}$. Being more aggressive, one could pack three numbers within a single 32-bit integer, giving a compression ratio of 6:1 and a fractional error of $2^{-32/3} \approx 0.6{\times}10^{-3}$ which is still not large enough to affect the visualisation of the data.

A final observation is that the benefits are not simply reduced disk usage. Since it is probable that there will still be too much data to be held within the memory of the workstation being used for visualisation, it will be necessary to read the data from disk during the visualisation. Reducing the quantity of data to be read from disk will greatly improve the speed of this operation. This may become very important since disk capacities have increased much more rapidly than disk i/o rates.