Report NASA Grant NAG-1-703

David Gottlieb
Brown University
October 1992

1 Stable and Asymptotocally Stable Compact Schemes

Recently, higher-order compact schemes have seen increasing use in the DNS (Direct Numerical Simulations) of the Navier-Stokes equations. Although they do not have the spatial resolution of Spectral methods, they offer significant increases in accuracy over conventional second order methods. They can be used on any smooth grid, and do not have an overly restrictive CFL dependence as compared with the $O(N^{-2})$ CFL dependence observed in Chebyshev Spectral methods on finite domains. In addition, they are generally more robust and less costly than Spectral methods. The issue of the relative cost of higher-order schemes (accuracy weighted against physical and numerical cost) is a far more complex issue, depending ultimately on what features of the solution are sought and how accurately they must be resolved. In any event, the further development of the underlying stability theory of these schemes is important.

It turns out that this schemes are very sensitive to boundary treatments. In particular *all* of the boundary conditions , currently used , allow non physical time growth of the solution. Recently, the stability characteristics of various compact fourth- and sixth-order spatial operators were assessed in reference [1], using the theory of Gustafsson, Kreiss and Sundstrom (G-K-S) for the semi-discrete Initial Boundary Value Problem (IBVP). The results were then generalized to the fully discrete case with Runge-Kutta time advancement using a recently developed theory by Kreiss. In all cases, favorable comparisons were obtained between G-K-S theory, eigenvalue determination, and numerical simulation. The conventional definition of stability is then sharpened to include only those spatial discretizations that are asymptotically stable (bounded, Left Half-Plane eigenvalues). It is shown that many of the higher-order schemes which are G-K-S stable are not asymptotically stable. It was concluded that in practical calculations, only those schemes which satisfied both definitions of stability were of any great usefulness.

It was shown in the above work of Carpenter et al. that conventional (optimal) finite difference closures at the boundaries of order greater than four are not G-K-S (or asymptotically) stable. Since fifth-order boundary closures possessing both stability properties were needed for sixth-order inner schemes, an alternate method for closing the boundaries was sought. The so-

lution was to parametrize the fifth-order difference formula at several points at each end of the spatial domain, thereby creating adjustable coefficients in the spatial operator. The asymptotic properties of the operator were established by the numerical determination of the eigenvalue spectrum, and the parameters were then adjusted until the desired spectrum was obtained. The resulting scheme was then tested for G-K-S stability, and if stable, satisfied both the desired criteria for a numerical discretization.

Several technical difficulties were encountered in trying to determine stable formulations in this manner. In general, a large number of free parameters were needed to find a combination which resulted in a stable formulation. This results from trying to achieve a high-order discretizations at the inflow boundary where the stencils are dramatically downwind, and mostly unstable. Although a stable closure condition was found for the sixth-order compact scheme, $(5^2, 5^2-6-5^2, 5^2)$ it was apparent that if schemes of higher accuracy were to be obtained, a systematic procedure was required to constrain the parameter space over which the search was performed. Another difficulty was that the numerical eigenvalue determination did not yield the exact eigenvalues of the spatial operator, but rather depended on numerical round off and the condition number of the resulting spatial operator. This was not found to be a significant problem for the schemes determined in the study, but it was found that many of the high-order schemes were not well conditioned.

The fundamental difficulty with determining a spatial operator based on the results from an eigenvalue analysis, is that it uses as a basis for the method the the spatial matrix resulting from discretization of the scalar wave equation $U_t + aU_x = 0$. **While G-K-S stability of a discretization on the scalar wave equation implies G-K-S stability on a system of hyperbolic equations, (if the boundary conditions are imposed in characteristic form)[2], the same is not in general true for asymptotic stability.** Therefore, there is no guarantee that the numerical scheme determined in this manner will be stable for an arbitrary hyperbolic problem. . An obvious remedy for the analysis presented by Carpenter et al.[1] would to have used the system not the scalar eigenvalue determination as a basis for devising stable closure formula. This would further constrain an already difficult search procedure to isolate the parameters at the boundaries which would produce an strictly asymptotically stable scheme. An altogether different procedure must be used if an arbitrarily high-order scheme is sought.

The approach of devising suitable boundary closures and then testing them with various stability techniques (such as finding the norm) is entirely the wrong approach when dealing with high-order methods. Very seldom are high-order boundary closures stable, making them difficult to isolate. An alternative approach is to begin with a norm which satisfies all the stability criteria for the hyperbolic system, and look for the boundary closure forms which will match the norm exactly. This method was used recently by Strand[4] to isolate stable boundary closure schemes for the explicit central fourth- and sixth-order schemes. The norm used was an energy norm mimicking the norm for the differential equations. Further research should be devoted to BC for high order schemes in order to make sure that the results obtained are reliable. The compact 4-th order and sixth order finite difference scheme had been incorporated into the a code to simulate flow past circular cylinders. This code will serve as a verification of the full spectral codes. A detailed stability analysis by Carprnter (from the fluid Mechanics Division) and Gottlieb gave analytic conditions for stability as well as asymptotic stability.This had been incorporated in the code in form of stable boundary conditions.

Effects of the cylinder rotations has been studied. The results differ from the known theoretical results. We are in the middle of analyzing the results.

A detailed analysis of the effects of the heating of the cylinder on the shedding frequency had been studied using the above schemes. It has been found that the shedding frequency decreases when the wire was heated. Experimental work is being carried out to affirm this result. This is carried out by Eric Voth in conjuction with D. Rudy from the Fluid Mechanics Division.

References

[1] Mark H. Carpenter, David Gottlieb and Saul Abarbanel, "The stability of numerical boundary treatments for compact high-order finite difference schemes", ICASE report 91-71, september 1991.

[2] Gottlieb D., Gunzburger M., and Turkel E. " ON numerical boundary treatment of hyperbolic systems for finite difference and finite element methods" SIAM Journal on Numerical Analysis 19,4 (1982).

[3] K. R. Sreenivasan, "Transition and turbulent wakes and chaotic dynamical systems", in S.H. Davis and J.L. Lumley, eds, Frontiers in Fluid Mechanics (Springer, New York, 1985), pp. 41-67.

[4] B. Strand- Ph.D thesis Uppsala University, to appear.

2 Wavelets

A major effort to adapt wavelets to the solution of PDE's is under investigation. It has been found by L. Jameson (agraduate student in the program) that using wavelets as a basis function for differentiation is equivalent to the use of finite difference schemes. The result is suppose to give a clue of how to implement boundary conditions. We attach a paper by him on the subject.

0.1 Introduction

The numerical solution of a partial differential equation requires an easily manipulated spatial approximation to the derivative of the unknown function as well as some method to march forward in time. In general one starts from given values of the unknown function, then a finite dimensional approximation, based on those values, is constructed. This approximation is differentiated and the result are read at the gridpoints. For example,in the psuedospectral Chebyshev method for the disretization of the equation

$$\frac{\partial U(x,t)}{\partial t} = \frac{\partial F[U(x,t)]}{\partial x}$$

One assumes that at a given time the values of $U(x_j,t)$ are given for some points $x_j = \cos(\frac{\pi j}{N})$, $(j = 0, N)$. Then one constructs the interpolation polynomial through those points and differentiate *this polynomial* to get approximate values for $\frac{\partial F[U(x,t)]}{\partial x}$ at the point x_j. this procedure can be viewed as a transformation from N given values (of the function) to new N values (approximating the derivative. This is the Chebyshev *Differentiation Matrix.* The numerical algorithm therefore is simple and the boundary conditions can be easily applied. It is natural to ask whether one gain by using wavelets instead of Chebyshev polynomials. . Since wavelets are well localized functions it is reasonable to conjecture that they might represent steep gradients or the development of a shock with a relatively small number of terms. consider a periodic function $f(x)$ given on an equally spaced mesh. Expand it in wavelet expansion and use the derivative of this expansion as an approximation to the derivative of $f(x)$. Lee Jameson, a student of D. Gottlieb has recently proved that approximation of a periodic function $f(x)$ in a wavelet basis and the differention of this approximation *yields nothing more than a finite difference approximation to a derivative.* The following is an outline of the proof:

i) Given a periodic function $f(x)$ let $\vec{s}$ represent the periodic scaling function coefficients of this function on the finest scale. This requires approximating the inner product of $f(x)$ with the scaling function on the finest scale. The matrix representation of this approximation is circulent in form: $C : \vec{f} \rightarrow \vec{s}$, where $\vec{f}$ represents $f(x)$ sampled on an evenly spaced grid.

ii) Let D be the mapping from the scaling function coefficients of $f(x)$ to the set of scaling function coefficients that represents the derivative of $f(x)$:

$D : \vec{s} \rightarrow \vec{\dot{s}}$. Since $f(x)$ is periodic then the matrix form of D is circulent in form.

iii) All circulent martrices of the same size commute, therefore we can apply the operator D directly to $\vec{f}$. The operator D has the effect of a finite difference operator, and the proof will be complete.

Therefore, the effect of first approximating in a wavelet basis, then differentiating in this basis and finally converting back from the wavelet basis to the original function is equal to applying the appropriate finite difference operator directly to the equally spaced sampled values of the original function $f(x)$. The proof provides an insight into the possibility of using wavelets for solutions of PDE's.

Wavelets while not more than known finite difference schemes can provide a mechnism for automatic adaptation of the mesh.

Since the proof is unpublished we will bring it here in some detail. This proof contains five sections: i) The first is the introduction outlining the presentation. ii) The second introduces scaling functions and wavelets. iii) The third discusses the approximation of a periodic function by scaling functions on the finest scale. iv) The fourth is concerned with the derivative of the scaling function and wavelet approximation of a function. v) The fifth concludes with a statement of the thesis that spatial wavelet approximations provide nothing more than finite difference methods do for the numerical solution of partial differential equations.

0.2 Definition of Wavelets

Wavelets have been precisely defined in many places [Daubechie], [Strang], [Beylkin], and others. The following outlines the most prominent properties of wavelets.

Begin with two sets of coefficients of length L, [Daubechie] $H = \{h_k\}_{k=0}^{L-1}$, $G = \{g_k\}_{k=0}^{L-1}$ called quadrature mirror filters, i.e., H and G are related by $g_k = (-1)^k h_{L-k}$ for $k = 0, ..., L-1$, which completely determine, along with the additional restriction of normalization,

$$\int_{-\infty}^{\infty} \phi(x)dx = 1,$$

the mother scaling function $\phi(x)$ and the mother wavelet $\psi(x)$, respectively,

by the following relations:

$$\phi(x) = \sum_{k=0}^{L-1} h_k \phi(2x - k),$$

and

$$\psi(x) = \sum_{k=0}^{L-1} g_k \phi(2x - k).$$

For the remainder of this paper the following notation will adopted: $\phi_k^j(x)$ and $\psi_k^j(x)$ will denote the mother scaling function and mother wavelet, respectively, at scale j and location k, i.e.,

$$\phi_k^j(x) = \phi(2^{-j}x - k),$$

and

$$\psi_k^j(x) = \psi(2^{-j}x - k).$$

A few of the ramifications of the above definitions are, first of all, that the wavelet $\psi(x)$ has M vanishing moments

$$\int_{-\infty}^{\infty} \psi(x) x^m dx = 0$$

for $m = 0, ..., M-1$, where the number of coefficients in H and G is equal to twice the number of vanishing moments, $L = 2M$ (this is true for the usual Daubechie wavelets only). Second, define V_j and W_j as linear span of ϕ_k^j and ψ_k^j over all location parameters k with the scale j fixed, i.e.,

$$V_j \equiv span_k \phi_k^j(x),$$

and

$$W_j \equiv span_k \psi_k^j(x).$$

These definitions lead to,

$$... \subset V_1 \subset V_0 \subset V_{-1} \subset ...,$$

$$\bigcap_{j \in Z} V_j = \{0\},$$

$$\bigcup_{j \in Z} V_j = L^2(R),$$

$$V_{j+1} = V_j \bigoplus W_j,$$

and finally,

$$L^2(R) = \bigoplus_{j \in Z} W_j.$$

These are the essential definitions and properties of scaling functions and wavelets. For more discussion and details see [Daubechie], [Mallat], and [Strang].

0.3 Approximating in Wavelet Bases

Scaling functions and wavelets were introduced in the previous section. As noted, V_j is the space spanned by $\phi_k^j(x)$ over all k. Without loss of generality, let scale $j = 0$ be the finest scale. Then, for example, $V_0 = V_1 \bigoplus W_1$. The approximation of an arbitrary periodic function $f(x)$ begins by projecting $f(x)$ onto each basis function $\phi_k^0(x)$ at the finest scale:

$$s_k^0 = \int_{-\infty}^{\infty} f(x)\phi_k^0(x)dx.$$

The approximation properties of scaling functions are determined by the number of vanishing moments of the associated wavelet: if the mother wavelet has M vanishing moments then the polynomials 1, x, ..., x^{M-1} are linear combinations of the translates of the mother scaling function $\phi(x - k)$ [Strang]. Furthermore, smooth functions can be approximated with error $\bigcirc(h^M)$ [Strang], where h represents the grid size.

Once the function $f(x)$ has been approximated on the finest scale, $j = 0$, then the coefficients s_k^0 can be decomposed into coefficients at scales that are twice as course at each decomposition using the following equations [Mallat]:

$$s_k^j = \sum_{n=1}^{n=2M} h_n s_{n+2k-2}^{j-1}$$

$$d_k^j = \sum_{n=1}^{n=2M} g_n s_{n+2k-2}^{j-1},$$

where M is the number of vanishing moments of the wavelet and $\{h\}$ and $\{g\}$ are the quadrature mirror filters defined in the previous section.

To restate, first the function $f(x)$ is approximated at the finest scale $j = 0$ with an error of order M to the coefficients s_k^0 then the coefficients at more coarse scales are found by the above pyramid-like decomposition.

A second method suggested by Beylkin et. al. [Beylkin] is to approximate the integral of each scaling function coefficient and each wavelet coefficient directly from the integral, i.e., by an appropriate quadrature formula approximating the following integrals:

$$s_k^j = \int_{-\infty}^{\infty} f(x)\phi_k^j(x)dx,$$

$$d_k^j = \int_{-\infty}^{\infty} f(x)\psi_k^j(x)dx.$$

In this paper the first method will be used so that all approximations will be made at the finest scale. Furthermore, all work will be done with the usual Daubechie wavelets. For wavelets supported on $[0, 3M]$ see appendix 1.

0.4 Quadrature Formula for Scaling Function

The scaling function coefficients of a function $f(x)$ on the finest scale are calculated exactly by,

$$s_k^0 = \int_{-\infty}^{\infty} f(x)\phi(x-k)dx.$$

For a numerical calculation, however, one must work with an estimate of the above coefficients, s_k^0, i.e., a suitable quadrature formula is needed.

Recall from the previously stated approximation properties of scaling functions that if the associated wavelet has M vanishing moments then one can represent polynomials up to order $M-1$ exactly by translations of the scaling function $\phi(x)$. Therefore, for $f(x)$ equal to a polynomial up to order $M-1$ the scaling function coefficients, s_k^0, can be found exactly. Consequently, there exist a set of coefficients $\{c_l\}_{l=0}^{M-1}$ such that

$$\int_{-\infty}^{\infty} f(y+k)\phi(y)dy = \sum_{l=0}^{M-1} c_l f(l+k),$$

where $f(x)$ can be a polynomial up to degree $M-1$, and the above integral is a shifted version of $s_k^0 = \int_{-\infty}^{\infty} f(x)\phi(x-k)dx$. More simply, the coefficients $\{c_l\}_{l=0}^{M-1}$ can be found [Beylkin] by solving the following linear system:

$$\sum_{l=0}^{M-1} l^m c_l = \int_{-\infty}^{\infty} x^m \phi(x)dx,$$

for $m = 0, 1, ..., M-1$.

Note that the above quadrature formula will yield an estimate of s_k^0 with error of order M. Also, note that the derivation of the coefficients of the quadrature formula depend only on the moments of the scaling function $\int_{-\infty}^{\infty} x^m \phi(x)dx$.

0.5 Example with D_6

The ideas in this paper are quite simple and are probably best illustrated by an example. The example will be for the Daubechie [Daubechie] wavelet D_6. The objective is to derive the matrix form of the mapping from evenly-spaced samples of a periodic function $f(x)$ to the scaling function coefficients on the finest scale s_k^0. Comparable results for the wavelets D_4 and D_8 are presented in appendix 2.

Recall, first of all, that in the previous subsection the coefficients $\{c_l\}_{l=0}^{M-1}$ were determined from the moments of the scaling function. Therefore, the scaling function moments must first be calculated.

Let M_l be the $l-th$ moment of the scaling function $\phi(x)$:

$$M_l = \int \phi(x)x^l dx,$$

and let μ_l be the $l-th$ moment of the filter h_k:

$$\mu_l = \sum_k k^l h_k.$$

Recall that it is required that $\phi(x)$ be normalized:

$$M_0 = \int \phi(x)dx = 1.$$

Also, by integrating the definition of $\phi(x)$ the following results:

$$\int \phi(x)dx = \sum_k h_k \int \phi(2x-k)dx.$$

Let $y = 2x - k$ to get,

$$1 = \frac{1}{2}\sum_k h_k \int \phi(y)dy,$$

which implies,

$$\mu_0 = \sum_k h_k = 2.$$

The μ_l for $l > 0$ can be found by straight-forward calculation. The M_l for $l > 0$ can be found from [see appendix 2]

$$M_m = (\frac{1}{2})^{m+1}\sum_l \begin{pmatrix} m \\ l \end{pmatrix} \mu_{m-l} M_l.$$

For the examples presented here moments up through the third moment are needed: $M_1 = \frac{1}{2}\mu_1$, $M_2 = \frac{1}{6}((\mu_1)^2 + \mu_2)$, and $M_3 = \frac{1}{28}((\mu_1)^3 + 4\mu_1\mu_2 + 2\mu_3)$.

After the appropriate moments are found, the coefficients $\{c_l\}_{l=0}^{M-1}$ can be found from

$$\sum_{l=0}^{M-1} l^m c_l = \int x^m \phi(x)dx$$

for $m = 0, 1, ..., M-1$. Specifically, for the D_6 wavelet the linear system in matrix form appears as,

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} M_0 \\ M_1 \\ M_2 \end{pmatrix},$$

which has the solution $c_0 = .1080$, $c_1 = .9667$, and $c_2 = -.0746$. In tabular form, the complete results for D_6 are,

i	M_i	μ_i	c_i
0	1	2	.1080
1	.8174	1.6348	.9667
2	.6681	1.3363	-.0746

Recall that the quadrature formula has the form,

$$s_k^0 = \sum_{l=0}^{M-1} c_l f(l+k) + \bigcirc(\frac{1}{N})^M,$$

where N is the number of points in the grid. If the function $f(x)$ is periodic then in matrix notation the above operation is $\vec{s} = C\vec{f}$ where C for D_6 and

a grid of 6 points appears as,

$$\begin{pmatrix} .108 & .967 & -.075 & 0 & 0 & 0 \\ 0 & .108 & .967 & -.075 & 0 & 0 \\ 0 & 0 & .108 & .967 & -.075 & 0 \\ 0 & 0 & 0 & .108 & .967 & -.075 \\ -.075 & 0 & 0 & 0 & .108 & .967 \\ .967 & -.075 & 0 & 0 & 0 & .108 \end{pmatrix}$$

The important point here is that the above matrix is circulent [Strange's book] in form. This is the most important observation in this paper, because all circulent matrices can be diagonalized by the Fourier matrix, i.e., all circulent matrices of the same dimensions have the same eigenvectors and therefore they commute. The importance of this property will become apparent after the wavelet derivative is discussed in the next section.

0.6 Derivative based on Wavelets

In the previous section the mapping from evenly-spaced samples of a periodic function, $f(x)$, to the scaling function coefficients on the finest scale, s_k^0, was discussed. The mapping is nothing more than a quadrature formula which is exact for $f(x)$ equal to a polynomial up to order $M-1$, where M is the number of vanishing moments of the wavelet. The question now is what is the mapping from s_k^0 to the coefficients of the derivative of $f(x)$, i.e., the scaling function coeffiecients, $\acute{s}_k^0$, of $f'(x)$. The answer is provided by Beylkin [Beylkin], and is presented in the following subsection. This section of the paper is organized as follows: i) Beylkin's results on derivative projections will be presented. ii) It will be argued that one need only consider the derivative mapping acting on the scaling function coefficients at the finest scale. iii) The similarity between the coefficients derived by Beylkin to finite difference approximations to the derivative will be presented.

0.7 Wavelet Coefficients of the Derivative

An arbitrary wavelet expansion of a function might contain wavelet coefficients and scaling coefficients at many scales. Beylkin derives the projection coefficients that map from scaling function coefficients and wavelet function

coefficients at a given scale to the derivative scaling function coefficients and wavelet function coefficients at the same scale. The matrix elements of these projections are computed from,

$$a_{il}^{j} = 2^{-2j} \int_{-\infty}^{\infty} \psi(2^{-j}x - i)\dot{\psi}(2^{-j}x - l)dx,$$

$$b_{il}^{j} = 2^{-2j} \int_{-\infty}^{\infty} \psi(2^{-j}x - i)\dot{\phi}(2^{-j}x - l)dx,$$

$$c_{il}^{j} = 2^{-2j} \int_{-\infty}^{\infty} \phi(2^{-j}x - i)\dot{\psi}(2^{-j}x - l)dx,$$

$$r_{il}^{j} = 2^{-2j} \int_{-\infty}^{\infty} \phi(2^{-j}x - i)\dot{\phi}(2^{-j}x - l)dx.$$

It is important to note that these projections are all at the same scale j, and that projections across different scales appear to be too complicated to yield closed-form solutions. The above projections, however, yield equations that are simple to work with [Beylkin].

It will be argued in the next section that in order to understand the numerical properties of the above projections it is only necessary to consider

$$r_l = \int_{-\infty}^{\infty} \phi(x - l)\frac{d}{dx}\phi(x)dx,$$

for $l \in Z$.

0.8 Derivative of Scaling Function Only

Before beginning the main argument of this subsection some new notation will be introduced. The vectors $\vec{h}$ and $\vec{g}$ contain the coefficients of the quadrature mirror filters which define the mother scaling function and mother wavelet, respectively. Define the unitary projection matrix P as,

$$P_{N\times N} \equiv \begin{bmatrix} \vec{h} & 0 & 0 \\ 0 & \vec{h} & 0 \\ 0 & 0 & \ddots \\ \vec{g} & 0 & 0 \\ 0 & \vec{g} & 0 \\ 0 & 0 & \ddots \end{bmatrix},$$

where the matrix subscripts denote the size of the matrix. Of course, P is nothing more than a matrix with the vectors $\vec{h}$ and $\vec{g}$ placed in its rows with the vector shifted two places to the right with each subsequent row. Also, let $\vec{s}^j$ contain the scaling function coefficients at scale j. P is, therefore, the matrix that maps $\vec{s}^j$ onto $\vec{s}^{j+1}$ and $\vec{d}^{j+1}$. Note that the vectors at scale $j+1$ are half as long as the vectors as scale j:

$$P : \left[\begin{array}{c} \vec{s}^j \end{array} \right] \rightarrow \left[\begin{array}{c} \vec{s}^{j+1} \\ \vec{d}^{j+1} \end{array} \right].$$

Let the four matrices R, A, B, C be the derivative projections defined in the previous subsection, i.e., Beylkin's coefficients, are $R \leftrightarrow r_{ij}$, $A \leftrightarrow \alpha_{ij}$, $B \leftrightarrow \beta_{ij}$, and $C \leftrightarrow \gamma_{ij}$. Explicitly, the mappings are

$$R : \vec{s}^j \rightarrow \vec{\vec{s}}^j,$$

$$A : \vec{d}^j \rightarrow \vec{\vec{d}}^j,$$

$$B : \vec{s}^j \rightarrow \vec{\vec{d}}^j,$$

$$C : \vec{d}^j \rightarrow \vec{\vec{s}}^j.$$

That is, if $\vec{s}^j$ and $\vec{d}^j$ denote the scaling and wavelet coefficients of a function at scale j then $\vec{\vec{s}}^j$ and $\vec{\vec{d}}^j$ denote the scaling and wavelet coefficients of the derivative of the function at the same scale.

For further illustration, suppose that a periodic function has been approximated on a grid with 16 scaling function coefficients. One application of the above defined matrix $P_{16\times16}$ on the vector $\vec{s}^0$ followed by the application of

the matrix $P_{8\times 8}$ on the vector $\vec{s}^1$ would appear as,

$$\begin{bmatrix} s_1^0 \\ s_2^0 \\ s_3^0 \\ s_4^0 \\ s_5^0 \\ s_6^0 \\ s_7^0 \\ s_8^0 \\ s_9^0 \\ s_{10}^0 \\ s_{11}^0 \\ s_{12}^0 \\ s_{13}^0 \\ s_{14}^0 \\ s_{15}^0 \\ s_{16}^0 \end{bmatrix} \rightarrow \begin{bmatrix} s_1^1 \\ s_2^1 \\ s_3^1 \\ s_4^1 \\ s_5^1 \\ s_6^1 \\ s_7^1 \\ s_8^1 \\ d_1^1 \\ d_2^1 \\ d_3^1 \\ d_4^1 \\ d_5^1 \\ d_6^1 \\ d_7^1 \\ d_8^1 \end{bmatrix} \rightarrow \begin{bmatrix} s_1^2 \\ s_2^2 \\ s_3^2 \\ s_4^2 \\ d_1^2 \\ d_2^2 \\ d_3^2 \\ d_4^2 \end{bmatrix}.$$

In the above decomposition there are three ways to represent exactly the same information: i) All information is at scale 0, i.e., use only the coefficients s_i^0 for $i = 1, ..., 16$. In this case the derivative coefficients would be found by applying the above defined matrix [Beylkin] $R_{16\times 16}$. ii) All information is at scale 1, i.e., use the coefficients s_i^1 and d_i^1 for $i = 1, ..., 8$. In this case all four of the above defined matrices $R_{8\times 8}$, $A_{8\times 8}$, $B_{8\times 8}$, and $C_{8\times 8}$, but the application of these four matrices is exactly the same as applying $R_{16\times 16}$ at scale 0 as is scenario (i). iii) This third scneario is the most unwieldy. The information is contained in two scales: the eight coefficients at scale 1, s_i^1 for $i = 1, ..., 8$, and eight coefficients at scale 2, s_i^2 and d_i^2 for $i = 1, ..., 4$. The difficulty here is the projection across scales. That is, how does one project the derivative of the wavelet at scale 2 onto the scaling function coefficients at scale 1. An attempt to calculate this projection has been made by this author but without success. One can, of course, approximate this projection but this is not very accurate and not elegant.

Recall that the main argument of this subsection is to illustrate that it is only necessary to take the derivative of a wavelet expansion on the finest scale, $j = 0$. First note that regardless of how the the information is represented in each of the above three scenarios there are always 16 degrees-of-freedom, i.e.,

it does not matter which 16 parameters are used to represent the function and its derivative. More explicitly, in the first scenario the derivative coefficients $\vec{s}^0$ are calculated by applying $R_{16\times16}$ to $\vec{s}^0$:

$$\vec{s}^0 = R_{16\times16} \cdot \vec{s}^0.$$

In the second scenario, however, one must first apply $P_{16\times16}$ to $\vec{s}^0$ to get the scaling and wavelet coefficients at scale 1. The derivative coefficients at scale 1 are then calculated by applying $D_{16\times16}$, where

$$D_{N\times N} \equiv \begin{bmatrix} R_{N/2\times N/2} & C_{N/2\times N/2} \\ B_{N/2\times N/2} & A_{N/2\times N/2} \end{bmatrix}.$$

To clarify, in scenario 2 the following operations are performed:

$$\begin{bmatrix} \vec{s}^1 \\ \vec{d}^1 \end{bmatrix} = D_{16\times16} \cdot P_{16\times16} \cdot \begin{bmatrix} \vec{s}^0 \end{bmatrix}.$$

However, scenarios 1 and 2 are exactly the same since

$$R_{N\times N} = 1/2(P^t_{N\times N} \cdot D_{N\times N} \cdot P_{N\times N}).$$

In summary, an attempt has been made to illustrate that the derivative coefficients of a scaling and wavelet expansion can be calculated at any scale. The goal for this author is to understand exactly what wavelets are and what they are doing, therefore, scale 0 provides the clearest scenario in which to work without sacrificing essential properties of wavelets.

Given, now, that it is sufficient to work on scale 0 to understand exactly what the wavelet derivative does, one must understand the ramifications of applying the matrix $R_{N\times N}$ to the vector $\vec{s}^0$. In the next subsection the similarity between the above defined matrix R and finite difference formulas for taking the derivative will be explored.

0.9 Wavelet Derivatives and Finite Difference

As the previous subsection attempted to illustrate, the essential properties of the wavelet derivative are contained in the matrix R. It was suprising, at least to this author, that the elements of the matrix R could differentiate

not only the vector $\vec{s}$ but also the equally spaced samples of a function $f(x)$, i.e., the matrix R displays finite difference properties. First of all, it is useful to simply note the similarity between finite difference coefficients and the elements of the matrix R. The following is a table of centered finite difference coefficients and the order of accuracy of the approximation to the derivative:

Order of Accuracy	Coefficients
2	$-\frac{1}{2}\ 0\ \frac{1}{2}$
4	$\frac{1}{12}\ -\frac{2}{3}\ 0\ \frac{2}{3}\ -\frac{1}{12}$
6	$-\frac{1}{60}\ \frac{3}{20}\ -\frac{3}{4}\ 0\ \frac{3}{4}\ -\frac{3}{20}\ \frac{1}{60}$
8	$\frac{1}{280}\ -\frac{4}{105}\ \frac{1}{5}\ -\frac{4}{5}\ 0\ \frac{4}{5}\ -\frac{1}{5}\ \frac{4}{105}\ -\frac{1}{280}$

Recall that the elements of the matrix R calculated by Beylkin provide the transformation from scaling function coefficients of a function to the scaling function coefficients of the derivative of the same function. The coefficients for the D_2 and D_4 wavelet expansions are exactly the same as the coefficients for the 2-nd and 4-th order centered finite difference formulas. The coefficients for the D_6 and D_8 wavelets are not exactly the same but are essentially the same (in a finite difference sense). The similarity can be seen clearly by plotting the finite difference coefficients and the wavelet coefficients on the same plot. The wavelet coefficients are,

Wavelet	Convolution Coefficients
D_2	$-\frac{1}{2}\ 0\ \frac{1}{2}$
D_4	$\frac{1}{12}\ -\frac{2}{3}\ 0\ \frac{2}{3}\ -\frac{1}{12}$
D_6	$-\frac{1}{2920}\ \frac{16}{1095}\ -\frac{53}{365}\ \frac{272}{365}\ 0\ -\frac{272}{365}\ \frac{53}{365}\ -\frac{16}{1095}\ \frac{1}{2920}$
D_8	$-\frac{39296}{49553}\ \frac{76113}{396424}\ -\frac{1664}{49553}\ \frac{2645}{1189272}\ \frac{128}{743295}\ -\frac{1}{1189272}$

If the above coefficients are treated as finite-difference coefficients then it would be nice to know the accuracy. To establish the finite-difference accuracy of the coefficients calculated by Beylkin note that a centered-finite-difference derivative approximation with $2K$ coefficients, $(\alpha_k)_{k=-K}^{K}$, can be written

$$\acute{f}(x_j) \sim \sum_{k=1}^{K} \alpha_k (f_{j+k} - f_{j-k}).$$

If the above equation is exact for $f(x) = x^r$ for $r = 0, ..., N$ but not for $r = N + 1$ then the equation is said to be N-th order accurate. Therefore,

one must check to see if

$$rx_j^{r-1} = \sum_{k=1}^{K} \alpha_k (x_{j+k}^r - x_{j-k}^r),$$

when $f(x) = x^r$. To simplify, one can let $x_j = j$ and check the following:

$$rj^{r-1} = \sum_{k=1}^{K} \alpha_k ((j+k)^r - (j-k)^r).$$

Now, treating the coefficients derived by Beylkin as nothing more than finite-difference coefficients one can check the accuracy. The following table contains the results of applying the Beylkin coefficients to various polynomials:

Wavelet	Exact for	But not for
D_2	x^2	x^3
D_4	x^4	x^5
D_6	x^6	x^7
D_8	x^8	x^9
D_{10}	x^{10}	x^{11}

The pattern in the above table is obvious and leads to the following conjecture: the cofficients derived by Beylkin which map scaling function coefficients of a function to the scaling function coefficients of the derivative of the function for the Daubechie wavelet D_{2M} can differentiate, exactly, a polynomial of degree $2M$ when applied to the samples of the polynomial in a finite-difference sense.

This concludes the second important observation of this paper (the first concerned approximating a function by scaling-function coefficients). The previously defined matrix R has as its elements the coefficients which display this finite-difference quality, and if the original function $f(x)$ is periodic then the matrix R is circulent in form. The following concluding section of this paper should unify the presentation.

0.10 Conclusion

The two important sections of this paper are sections 3.4.3 and 3.4.44. In section 3.4.3 it was established that if given a periodic function $f(x)$ then the scaling function coefficients $\vec{s}$ of the function at the finest scale can be

approximated by a quadrature formula which in matrix form,

$$\vec{s} = C\vec{f},$$

yields a circulent matrix C. In section 4 it was noted that the coefficients derived by Beylkin which map the scaling function coefficients of a periodic fuction to the scaling function coefficients of the derivative of the function is also circulent in form when written in matrix notation,

$$\vec{s'} = D\vec{s}.$$

Furthermore, the matrix D can differentiate (in a finite-difference sense) polynomials exactly up to the order of the wavelet. Now, combine the results of sections 3 and 4 to get the following relation:

$$\vec{f'} = C^{-1}DC\vec{f}.$$

Throughout the paper it has been noted that C and D are circulent in form when $f(x)$ is periodic. Circulent matrices of the same dimensions can, however, be diagonalized by the same matrix, the Fourier matrix of appropriate dimensions, and this implies that all circulent matrices of the same dimensions commute. Therefore, the previous relation simply becomes,

$$\vec{f'} = D\vec{f},$$

but when D is applied to the samples of a function it acts as a finite-difference operator. In conclusion, when wavelets are used to solve partial differential equations numerically they appear to provide nothing more than finite-difference methods provide.

0.11 References

[1] G. Beylkin, R. Coifman and V. Rokhin, "Fast Wavelet Transforms and Numerical Algorithms I", Comm. Pure Appl. Math., **64** (1991), pp. 141-184.

[2] I. Daubechies, "Orthonormal Basis of Compactly Supported Wavelets", Comm. Pure Appl. Math., **41** (1988), pp. 909-996.

[3] G. Beylkin, "On the Representation of Operators in Bases of Compactly Supported Wavelets", Preprint

[4] G. Strang, "Wavelets and Dilation Equations: A Brief Introduction", SIAM Review, Vol. 31, No.4, pp. 614-627, Dec. 1989.

[5] S. Mallat, "Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(R)$", Trans. Amer. Math. Soc., Vol. 315, No. 1, pp. 69-87, Sept. 1989.

[6] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. Pattern Anal. and Machine Intel., 11 (1989), pp. 674-693.

[7] G. Strang, "Introduction to Applied Mathematics", Wellesley-Cambridge Press, 1986, pp. 297-298.

0.12 Appendix 1

0.13 Wavelets Supported on (0,3M)

In this appendix our wavelets are supported on $[0, 3M]$ where M is the number of vanishing moments of the wavelet. These are not the usual Daubechie wavelets, but for these wavelets the scaling function coefficients of a periodic function $f(x)$ can be approximated with error of order M simply by sampling $f(x)$ at the correct location.

To begin, assume that there exist a unique τ_M, fixed for a fixed number of vanishing moments, M, of the wavelet, such that

$$\int \phi(x + \tau_M) x^m dx = 0$$

for $m = 1, 2, ..., M-1$. Furthermore, recall the definition of the scaling function coefficient and expand the integrand $f(x)$ in a Taylor series about x_0 ($f_0' = f'(x_0)$):

$$s_k^0 = \int f(x)\phi(x-k)dx =$$

$$f_0 \int \phi(x-k)dx + f_0' \int (x-x_0)\phi(x-k)dx + f_0'' \int (x-x_0)^2 \phi(x-k)dx + ...$$

Now, shift the variable of integration by $y = x - \tau - k$, and choose the point of expansion, x_0, to be $\tau + k$ to get,

$$s_k^0 =$$

$$f(\tau+k) \int \phi(y+\tau)dy + f'(y+k) \int y\phi(y+\tau)dy + f''(y+k) \int y^2 \phi(y+\tau)dy + ...$$

Now, rename τ as τ_M and the above integrals are of the form,

$$\int \phi(x + \tau_M) x^m dx = 0,$$

and therefore vanish for $m = 1, ..., M - 1$ leading to,

$$s_k^0 = f(\tau_M + k) + f^{(M)}(\tau_M + k) \int y^M \phi(y + \tau_M) dy + ...,$$

i.e., the approximation of the scaling function coefficient s_k^0 up to order M is made by sampling $f(x)$ at the position $\tau_M + k$.

Note that all of the above calculations could have been carried out for the first derivative of $f(x)$ giving an approximation to the scaling function coefficients, $\acute{s}_k^0$, of $f'(x)$:

$$\acute{s}_k^0 = \acute{f}(\tau + k) + f^{(M+1)}(\tau + k) \int y^M \phi(y + \tau) dy + ...$$

It was assumed above that there exist one τ_M such that

$$\int \phi(x + \tau_M) x^m dx = 0,$$

for $m = 1, ..., M - 1$. For $m = 1$ this τ_M is easy to find:

$$\int \phi(x + \tau_M) x dx = \int \phi(x)(x - \tau_M) dx$$

$$= \int x\phi(x) dx - \tau_M \int \phi(x) dx.$$

But $\int phi(x) dx = 1$, therefore,

$$\tau_M = \int x\phi(x) dx.$$

That is, τ_M is simply the first moment of $\phi(x)$. To find τ_M for $m > 1$ the calculations are simple but a bit longer and require the result from the following theorem to show that there is one τ_M which is the same for all $m = 1, ..., M - 1$.

If $\int \phi(x) dx = 1$ and there exists τ such that $\int \phi(x + \tau) x^m dx = 0$ for $m = 1, ..., M - 1$ **then** $\int \phi(x) x^m dx = (\int \phi(x) x dx)^m$ for $m = 1, ..., M - 1$.

Proof: Start with

$$\int \phi(x+\tau)x^m dx = 0,$$

and let $y = x + \tau$ to get,

$$\int \phi(y)(y-\tau)^m = 0.$$

Using the binomial theorem this becomes,

$$\int \phi(y) \sum_{r=0}^{m} \left(\begin{array}{c} m \\ r \end{array} \right) y^r (-\tau)^{m-r} dy = 0.$$

Let the moments of $\phi(x)$ be denoted by $M_l = \int \phi(x) x^l dx$ to get

$$\sum_{r=0}^{m} \left(\begin{array}{c} m \\ r \end{array} \right) (-\tau)^{m-r} M_r = 0.$$

A simple calculation yields $\tau = M_1$. Using this value of τ and summing only up to $m-1$ the previous expression becomes,

$$\sum_{r=0}^{m-1} \left(\begin{array}{c} m \\ r \end{array} \right) (-M_1)^{m-r} M_r + M_m = 0.$$

Or,

$$M_m = -\sum_{r=0}^{m-1} \left(\begin{array}{c} m \\ r \end{array} \right) (-1)^{m-r} (M_1)^{m-r} M_r.$$

From the hypotheses it is known that $M_0 = \int \phi(x)dx = 1$. Therefore, $M_\rho = M_1^\rho$ for $\rho = 0, 1$, and with this knowledge it is easy to show that $M_\rho = M_1^\rho$ for $\rho = 2$:

$$M_m = -\sum_{r=0}^{m-1} \left(\begin{array}{c} m \\ r \end{array} \right) (-1)^{m-r} (M_1)^{m-r} M_1^r,$$

which holds for $m = 1, 2$. Combine the powers of M_1 to get,

$$M_m = -M_1^m \sum_{r=0}^{m-1} \left(\begin{array}{c} m \\ r \end{array} \right) (-1)^{m-r}.$$

But, this is nothing more than,

$$M_m = -M_1^m[(1-1)^m - 1],$$

or simply,

$$M_m = M_1^m,$$

where $m = 1, 2$. The proof is complete, since higher powers of m can be found by induction.

0.14 Appendix 2

In this appendix the moments of $\phi(x)$ will be calculated in closed form. Begin with the definition of the scaling function,

$$\phi(x) = \sum_k \phi(2x - k).$$

Next, calculating the m-th moment of $\phi(x)$ yields,

$$\int \phi(x)x^m = \sum_k h_k \int \phi(2x - k)x^m dx.$$

Change the variable of integration such that $y = 2x - k$ to get,

$$\int \phi(x)x^m = \sum_k h_k \int \phi(y)(1/2)^m (y + k)^m 1/2 dy,$$

$$= (1/2)^{m+1} \sum_k h_k \int \phi(y)(y + k)^m dy.$$

Now, recall the binomial theorem to get,

$$\int \phi(x)x^m = (1/2)^{m+1} \sum_k h_k \int \phi(y) \sum_{l=0}^{m} \binom{m}{l} y^l k^{m-l} dy$$

Rewrite the moments of $\phi(x)$ as $M_l = \int x^l \phi(x) dx$ to get,

$$M_m = (1/2)^{m+1} \sum_{l=0}^{m} \binom{m}{l} \sum_k h_k k^{m-l} M_l.$$

Now let $\mu_l = \sum_k h_k k^l$ to get

$$M_m = (1/2)^{m+1} \sum_{l=0}^{m} \binom{m}{l} \mu_{m-l} M_l.$$