行政院國家科學委員會補助專題研究計畫成果報告

小波與數值偏微分方程(3/3)

計畫類別：■個別型計畫　□整合型計畫

計畫編號：NSC 90－2115－M－002－020

執行期間：90 年 8 月 1 日至 91 年 7 月 31 日

計畫主持人：陳宜良教授

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份

□赴大陸地區出差或研習心得報告一份

□出席國際學術會議心得報告及發表之論文各一份

□國際合作研究計畫國外研究報告書一份

執行單位：台大數學系

中　華　民　國 91 年 10 月 31 日

Difference Wavelet— Theory and A Comparison Study *

I-Liang Chern † Chien-Chang Yen ‡

May 1, 2001

Abstract

Wavelet methods with polynomial filters are usually favored in applications for their availability of fast wavelet transforms and compactly supported property. However, wavelet methods with rational filters have more degree of freedom to probably achieve smaller condition numbers, more regularity and better efficiency. Such methods can be attractive if they also possess fast transformation algorithms and have fast decay property as if the corresponding wavelets are of compact supports. In the first part of this paper, we propose a new wavelet method with rational filter which does have these properties. We call it the difference wavelet method. It is a generalization of Butterworth wavelets. The analysis part is simply averaging and finite differencing. The corresponding wavelet coefficients measure the finite differences of the averages of an input data sequence. Its synthesis part involves rational filters, which can be performed with linear computational complexity by the cyclic reduction method. Their Riesz basis property, biorthogonality, decay and regularity properties are investigated.

In the second part of this paper, we perform comparison studies of the difference wavelet method (Diff) with three other popular wavelet methods: the Cohen-Daubechies-Feauveau biorthogonal wavelet method (CDF), the Daubechies orthogonal wavelet method (Daub) and the Chui-Wang semi-orthogonal wavelet method (CW). Natural criteria in designing good wavelet methods for representing functions and operators are *fast, stable* and *efficient*. Therefore, the items of our first comparison include (i) operation counts for performing transformations, (ii) condition numbers of the wavelet transformations, (iii) compression ratios, by some numerical experiments, for representing (smooth or non-smooth) data sequences and matrices (smooth or non-smooth kernels).

*This work was supported by the National Science Council of the Republic of China under Contract NSC86-2115-M002-008

†Department of Mathematics, National Taiwan University, Taipei, Taiwan.

‡Institute of Astronomy and Astrophysics, Academia Sinica, Taipei, Taiwan.

The results show that (i) Diff, Daub and CDF have about the same operation counts for performing wavelet transform, and CW has more; (ii) Diff has about the same condition numbers as those of CDF and CW; (iii) Diff has better compression ratio for both (smooth or non-smooth) data sequences and matrices (smooth or non-smooth kernels).

The items of our second comparison include regularity, approximation power (the constant appears in the approximation estimate), approximation errors for non-smooth functions (where Gibbs phenomena appear) and the "essential supports." The results show that Diff has better regularity and better approximation ability with only slightly bigger essential supports. It is evidently that the better efficiency of Diff for smooth functions is due to its better regularity. It is surprising that, even for non-smooth functions, Diff is comparable to, sometimes even superior to, other methods, despite of its infinite-support property.

This paper is organized as follows. Sec. 1 is the preliminary. Sec. 2 provides the theory of the difference wavelet method. Sec. 3 contains the comparison studies. Experts are suggested to read Sec. 3 directly.

1 Preliminary

A wavelet expansion method decomposes data (or functions) into fluctuations at various resolutions. It depends on four sets of filter coefficients: $\{h_k\}_{k\in\mathbf{Z}}$, $\{g_k\}_{k\in\mathbf{Z}}$, $\{\tilde{h}_k\}_{k\in\mathbf{Z}}$ and $\{\tilde{g}_k\}_{k\in\mathbf{Z}}$. The first two are called the analysis filters, the latter two the synthesis filters. A data sequence $c_j = \{c_{j,i}\}_{i\in\mathbf{Z}}$ at resolution level j can be decomposed, through the analysis filters, into the following two sets of data sequences at level $j-1$:

$$\begin{cases} \text{the low-pass data: } c_{j-1,i} = \sqrt{2}\sum_k h_k c_{j,2i-k}, \\ \text{the high-pass data: } d_{j-1,i} = \sqrt{2}\sum_k g_k c_{j,2i+1-k}. \end{cases} \tag{1.1}$$

Here, $\sqrt{2}$ is a normalized scale factor [12]. By applying the above transform: $c_j \mapsto (c_{j-1}, d_{j-1})$ recursively for $j = J, J-1, \cdots, 1$, one can decompose a given data sequence c_J at the finest resolution J into $(c_0, d_0, d_1, \cdots, d_{J-1})$, the averages at coarsest resolution and the fluctuations at various resolutions. The mapping $T_J : c_J \mapsto (c_0, d_0, d_1, \cdots, d_{J-1})$ is called a discrete wavelet transformation.

The data sequence c_j can be reconstructed from (c_{j-1}, d_{j-1}) through the synthesis filters $\{\tilde{h}_k\}_{k\in\mathbf{Z}}$ and $\{\tilde{g}_k\}_{k\in\mathbf{Z}}$ by:

$$c_{j,i} = \sqrt{2}\sum_k \left[\tilde{h}_{i-2k} c_{j-1,k} + \tilde{g}_{i-2k-1} d_{j-1,k}\right]. \tag{1.2}$$

By applying this inverse transform recursively for $j = 1, \cdots, J-1$, c_J can be recovered from $(c_0, d_0, \cdots, d_{J-1})$.

Let us define the $z-$transform of $\{h_k\}_{k\in\mathbb{Z}}$ by $h(z) = \sum_k h_k z^k$, and still call it a filter. It can be a polynomial (i.e. a filter with finite length), or a rational function (infinite length), or in general, a Laurent series.

In order to have c_j be reconstructed from (1.2), the filter bank (i.e. the collection of analysis and synthesis filters) need to satisfy the following perfect reconstruction condition [9, 16]:

$$h(z)\tilde{h}(z) + g(z)\tilde{g}(z) = 1, \tag{1.3}$$
$$h(-z)\tilde{h}(z) - g(-z)\tilde{g}(z) = 0. \tag{1.4}$$

A formal calculation gives

$$h(z)\tilde{h}(z) + h(-z)\tilde{h}(-z) = 1, \tag{1.5}$$

$$g(z) = \tilde{h}(-z)P(z^2), \tag{1.6}$$
$$\tilde{g}(z) = h(-z)/P(z^2), \tag{1.7}$$

for some Laurent series $P(z)$. If we require both analysis and synthesis filters are polynomials, then $P(z) = z^m$ for some integer m. In this case, we can normalize $P(z) = 1$ [9]. In general, for stability consideration, we should choose $P(z)$ to be in the Wiener class, that is, $P(z) = \sum_k p_k z^k$ with $\sum_k |p_k| < \infty$ and $P(z) \neq 0$ for all $|z| = 1$ (see Chui [3]).

So, a general procedure to construct filter bank is to find $h(z)$ and $\tilde{h}(z)$ to satisfy (1.5). We may normalize them by

$$h(1) = \tilde{h}(1) = 1. \tag{1.8}$$

Then we define $g(z)$ and $\tilde{g}(z)$ by (1.6) and (1.7) with a proper function $P(z)$ chosen in the Wiener class.

A general principle to design filters in applications is to have the corresponding wavelet transform to be *fast, stable* and the corresponding wavelet approximation to be *efficient*. The term "fast" means that the wavelet transform T_J and its inverse are of linear computational complexity. Usually, polynomial filters are favored. However, a rational filter which can be performed with linear complexity is also acceptable in many applications. The term "stable" means that the forward and inverse wavelet transforms are unconditionally stable in ℓ^2, independent of the resolution level J, that is, both $\|T_J\|, \|T_J^{-1}\| = O(1)$. The term "efficient" means that only a small amount of wavelet coefficients $d_{j,i}$ plus the averages $c_{0,i}$ are sufficient to approximate the original data accurately.

It is well-known that the stability condition can be characterized by the Riesz basis property of a corresponding wavelet function ψ, see [9, 10]. More precisely,

associated with the analysis filter bank, one can define a refinable function (or called scaling function) ϕ and a wavelet function ψ as follows:

$$\phi(x) = 2\sum_k h_{-k}\phi(2x-k), \tag{1.9}$$

$$\psi(x) = 2\sum_k g_{1-k}\phi(2x-k). \tag{1.10}$$

The function $h(z)$ is called the mask of the refinable function ϕ. Let us define $\psi_{j,i}(\cdot) = 2^{j/2}\psi(2^j \cdot -i)$. Then the forementioned stability condition is equivalent to that $\{\psi_{j,i}\}_{i,j\in\mathbb{Z}}$ forms a Riesz basis in $\mathrm{L}^2(\mathbb{R})$. That is, there exist constants $0 < \gamma$, $\Gamma < \infty$ such that for any sequence $\{d_{j,i}\}_{i,j\in\mathbb{Z}}$, we have

$$\gamma \sum_{i,j\in\mathbb{Z}} |d_{j,i}|^2 \le \| \sum_{i,j\in\mathbb{Z}} d_{j,i}\psi_{j,i}\|_{L^2}^2 \le \Gamma \sum_{i,j\in\mathbb{Z}} |d_{j,i}|^2.$$

For the synthesis filter bank, one also defines

$$\tilde{\phi}(x) = 2\sum_k \tilde{h}_k\tilde{\phi}(2x-k), \tag{1.11}$$

$$\tilde{\psi}(x) = 2\sum_k \tilde{g}_{k-1}\tilde{\phi}(2x-k). \tag{1.12}$$

Then $\{\tilde{\psi}_{j,i}\}_{i,j\in\mathbb{Z}}$ forms the dual Riesz basis of $\{\psi_{j,i}\}_{i,j\in\mathbb{Z}}$ in $\mathrm{L}^2(\mathbb{R})$, i.e. $(\psi_{j,i}, \tilde{\psi}_{\ell,k}) = \delta_{j,\ell}\delta_{i,k}$. These two wavelets are called biorthogonal wavelets [9]. It was commented by Cohen-Daubechies-Feauveau [9] and Dahmen [10] that biorthogonal wavelets are more flexible to use than orthogonal wavelets in practice.

Three popular wavelets are the Daubechies's orthogonal wavelet (Daub) [11], Cohen-Daubechies-Feauveau's biorthogonal wavelet (CDF) [9] and Chui-Wang's semi-orthogonal wavelet (CW) [4]. The filter bank of the CDF wavelet is defined as follows:

$$h(z) = \tilde{g}(-z) = z^{-\lfloor r/2\rfloor}\left(\frac{1+z}{2}\right)^r, \tag{1.13}$$

$$\tilde{h}(z) = g(-z) = z^{-\lfloor(\tilde{r}+1)/2\rfloor}\left(\frac{1+z}{2}\right)^{\tilde{r}} Q_K(z). \tag{1.14}$$

Here, $r + \tilde{r} = 2K$, $K > 0$ is an integer parameter, and

$$Q_K(z) = \sum_{n=0}^{K-1}\binom{K-1+n}{n}\left(\frac{2-z-z^{-1}}{4}\right)^n. \tag{1.15}$$

We call the parameter r the averaging order and $\tilde{r}$ the differencing order, or the number of vanishing moments in other literatures.

The filter bank of Daubechies' orthogonal wavelet is defined by

$$h(z) = \tilde{h}(z^{-1}) = \tilde{g}(-z) = g(-z^{-1}) = \left(\frac{1+z}{2}\right)^K Q(z), \tag{1.16}$$

where $Q(z)Q(z^{-1}) = Q_K(z)$. The filter bank of Chui-Wang's semi-orthogonal wavelet is defined by

$$\begin{aligned} h(z) &= \left(\frac{1+z}{2}\right)^K G_K(z)/G_K(z^2), \\ g(z) &= \left(\frac{1-z}{2}\right)^K /G_K(z^2), \\ \tilde{h}(z) &= \left(\frac{1+z}{2}\right)^K, \\ \tilde{g}(z) &= \left(\frac{1-z}{2}\right)^K G_K(-z), \end{aligned}$$

where $G_K(z) := \sum_k N_{2K}(k) z^k$ and $N_{2K} := 1_{[0,1)}^{*2K}$ is the B-spline of order $2K-1$.

It is well-known that the regularity of a wavelet is related to its orders r, $\tilde{r}$ and the magnitude of its amplification factor (i.e. $Q_K(e^{i\xi})$ in CDF wavelet and $G_K(e^{i\xi})/G_K(e^{i2\xi})$ in CW wavelet) [12]. The more regularity a wavelet has, the better approximation ability it can have [12]. We can look for rational filters that have smaller amplification factors. The gain is that the corresponding wavelet function is smoother. The prices to pay are that (i) it has infinite support, and (ii) a linear system needs to be solved to perform wavelet transform. Problem (i) is not severe if the essential support (e.g. the region where $|\tilde{\phi}| > \epsilon$) is still small. Problem (ii) is also solvable if a fast algorithm for solving this linear system is available. Below, we propose the difference wavelet method which has a relatively small amplification factor.

2 Difference Wavelets

2.1 The filter bank of the difference wavelet method

Given a positive integer K, let us define the filter bank of the difference wavelet method to be

$$h(z) = z^{-[\frac{r}{2}]}\left(\frac{1+z}{2}\right)^r, \tag{2.1}$$

$$g(z) = z^{-[\frac{\tilde{r}+1}{2}]}\left(\frac{1-z}{2}\right)^{\tilde{r}}, \tag{2.2}$$

$$\tilde{h}(z) = z^{-[\frac{\tilde{r}+1}{2}]}\left(\frac{1+z}{2}\right)^{\tilde{r}}/P_K(z^2), \tag{2.3}$$

$$\tilde{g}(z) = z^{-[\frac{r}{2}]}\left(\frac{1-z}{2}\right)^{r}/P_K(z^2), \tag{2.4}$$

$$P_K(z^2) = z^{-K}\left(\frac{1+z}{2}\right)^{2K} + (-z)^{-K}\left(\frac{1-z}{2}\right)^{2K}, \tag{2.5}$$

where $r + \tilde{r} = 2K$. Roughly speaking, the difference of this method from the CDF method is to replace the filter $Q_K(z)$ by $1/P_K(z^2)$. We call this method the difference wavelet method because its high-pass filter is simply a finite difference. In the case of $r = 0$ and $\tilde{r} = 2K$, $\tilde{h}(z)$ is the Butterworth filter, which was well-known in the field of signal processing [15]. To justify this method to be valuable, we shall show that

1. the operation count to perform $1/P_K(z^2)$ is almost the same as that of $Q_K(z)$;
2. it is stable in $\mathrm{L}^2(\mathbb{R})$; in fact the condition numbers of the difference wavelet transforms are comparable to those of the CDF transform;
3. the corresponding $\tilde{\phi}$ decays exponentially at far field, in fact, the lengths of the "essential supports" of the difference wavelets are about twice of those of the CDF wavelets;
4. it is more efficient in the sense that it has better approximation ability and better compression ratio.

We shall devote to these issues below.

2.2 Fast algorithm for difference wavelet transform

We adopt the cyclic reduction method [14] to perform the filter $1/P_K(z^2)$. Firstly, we factor $P_K(z^2)$ into

$$\begin{aligned} P_K(z^2) &= \prod_{k=1}^{[K/2]} \frac{1}{1+2\alpha_k}\left(\alpha_k z^{-2} + 1 + \alpha_k z^2\right) \\ &\equiv \prod_{k=1}^{[K/2]} P^k(z^2). \end{aligned} \tag{2.6}$$

Here,

$$0 < \alpha_k = 1/\left(\tan^2\theta_k + 1/\tan^2\theta_k\right) < 1/2, \tag{2.7}$$

$$\theta_k = \begin{cases} \frac{(2k-1)\pi}{4K} & \text{if } K \text{ is even,} \\ \frac{k\pi}{2K} & \text{if } K \text{ is odd,} \end{cases} \tag{2.8}$$

where $k = 1, \cdots, 2K$ for even K, and $k = 1, \cdots, K-1, K+1, \cdots, 2K-1$ for odd K. This factorization can be derived from the fact that $z = i \tan\theta_k$ are the roots of $P_K(z^2) = 0$.

Secondly, for each $k = 1, \cdots, [K/2]$, we perform the filter $1/P^k(z^2)$ by solving a tridiagonal system $Av = f$, where $A = \mathrm{diag}(\alpha_k, 1, \alpha_k)$ with $|\alpha_k| < 1/2$. This system can be solved by the cyclic reduction method [14]. We briefly describe it below for reader's convenience.

A one-step cyclic reduction reduces this system into a system of half size with the same structure. We apply this reduction recursively until a small system is met or until A becomes almost diagonal. Then this reduced system can be solved directly. To describe this reduction procedure, we assume the current linear system is of the form:

$$a_\ell v_{i-1}^\ell + v_i^\ell + a_\ell v_{i+1}^\ell = f_i^\ell, \tag{2.9}$$

Here, ℓ is the index of the recursion procedure. We eliminate $v_{2i\pm1}^\ell$ terms to obtain

$$-{a_\ell}^2 v_{2i-2}^\ell + (1 - 2{a_\ell}^2) v_{2i}^\ell - {a_\ell}^2 v_{2i+2}^\ell = f_{2i}^\ell - a_\ell(f_{2i-1}^\ell + f_{2i+1}^\ell).$$

We rename the variables:

$$\begin{aligned} v_i^{\ell-1} &= v_{2i}^\ell, \\ f_i^{\ell-1} &= \frac{1}{1-2{a_\ell}^2}\left(f_{2i}^\ell - a_\ell(f_{2i-1}^\ell + f_{2i+1}^\ell)\right), \\ a_{\ell-1} &= \frac{-{a_\ell}^2}{1-{a_\ell}^2}. \end{aligned} \tag{2.10}$$

Then $v_i^{\ell-1}$ satisfies

$$a_{\ell-1} v_{i-1}^{\ell-1} + v_i^{\ell-1} + a_{\ell-1} v_{i+1}^{\ell-1} = f_i^{\ell-1}.$$

Thus, we arrive a system of half size with the same structure. Notice that, from (2.10), $a_{\ell-m}$ converges to zero quadratically as $m \to \infty$. This is because $|a_\ell| < 1/2$ initially. Usually, $m = 5$ (i.e. five-level reductions) will make $a_{\ell-m}$ go down to 10^{-14}. Once $\{v_i^{\ell-m}\}$ are found, we can reconstruct $\{v_i^{\ell-m+1}\}$ from

$$\begin{aligned} v_{2i}^{\ell-m+1} &= v_i^{\ell-m}, \\ v_{2i+1}^{\ell-m+1} &= f_{2i+1}^{\ell-m+1} - a_{\ell-m+1}\left(v_i^{\ell-m} + v_{i+1}^{\ell-m}\right). \end{aligned}$$

We can continue this reconstruction procedure recursively from $\ell - m + 1$ to ℓ to obtain v_i^ℓ.

By a direct calculation, the amount of work for performing $1/P_K(z^2)$ is less than $(4A + 3M)[K/2]$ per each datum. Here, A is the addition operation and M is the multiplication operator. Thus, the procedure to perform the rational filters $\tilde{h}(z)$ and $\tilde{g}(z)$ is of linear complexity. In the next section, a comparison study shows that the operation counts for the difference wavelet method is about the same as those of the Daubechies' orthogonal wavelet method and of the CDF method.

2.3 Stability of the difference wavelet method

For the difference wavelet method, from (2.1), the corresponding scaling function ϕ is the B-spline:

$$\phi(x) = 1^{*r}_{[-1,0)}(x - [r/2]).$$

The corresponding dual scaling function $\tilde{\phi}$ has the following properties.

Proposition 1 *1. $\tilde{\phi}$ is symmetric.*

2. The Fourier transform of $\tilde{\phi}$ satisfies

$$|\widehat{\tilde{\phi}}(\xi)| = O\left(|\xi|^{-(\tilde{r}-r)/2-1}\right). \tag{2.11}$$

3. $\tilde{\phi}$ decays at $\pm\infty$ exponentially for $\tilde{r} \geq r$:

$$\int \left|e^{\sigma'|x|}\tilde{\phi}(x)\right|^2 dx < \infty, \ \forall \sigma' < \sigma, \tag{2.12}$$

where

$$\sigma = 2\left|\ln\tan\left(\frac{K+1}{4K}\pi\right)\right|.$$

Proof. 1. The symmetric property of $\tilde{\phi}$ follows from $\tilde{h}(z) = \tilde{h}(-z)$.

2. Using Fourier transform and (1.11), we have

$$\begin{aligned}\widehat{\tilde{\phi}}(\xi) &= \prod_{\ell=1}^{\infty}\tilde{h}(e^{-i\xi/2^\ell}) \\ &= e^{-i\xi/2}\left(\frac{\sin\xi/2}{\xi/2}\right)^{\tilde{r}}\prod_{\ell=1}^{\infty}\frac{1}{P_K(e^{-i2\xi/2^\ell})}.\end{aligned} \tag{2.13}$$

So, to prove (2.11) we should show that the infinite product $\prod_{\ell=1}^{\infty} 1/P_K(e^{-2i\xi/2^\ell})$ converges unformly and absolutely on every compact subsets in the complex plane and has the following asymptotic estimate:

$$\prod_{\ell=1}^{\infty}\frac{1}{P_K(e^{-i2\xi/2^\ell})} = O(|\xi|^{K-1}) \ \text{ for } |\xi| \gg 1. \tag{2.14}$$

To see this, first, from the continuity of P_K and $P_K(1) = 1$, there exist constants $\xi_0, C > 0$ such that

$$\left|\frac{1}{P_K(e^{-2i\xi})} - 1\right| < C|\xi| \tag{2.15}$$

for all $|\xi| \leq \xi_0$.

Given any compact subset $B \subset \mathbb{C}$, there exists an integer L_0 such that $B \subset \{\xi \in \mathbb{C} \mid |\xi| \le 2^{L_0}\xi_0\}$. For any $\xi \in B$, there exists a positive integer L such that $2^{L-1}\xi_0 < |\xi| < 2^L\xi_0$. We split the above infinite product $\prod_{\ell=1}^{\infty}$ into $\prod_{\ell=1}^{L} \cdot \prod_{\ell=L+1}^{\infty}$. From (2.15), the second term has the estimate:

$$\begin{aligned}\prod_{\ell=L+1}^{\infty}\left|\frac{1}{P_K(e^{-i2\xi/2^\ell})}\right| &\le \prod_{\ell=L+1}^{\infty}\left(1+C\frac{|\xi|}{2^\ell}\right)\\ &\le \exp(C\xi_0) \equiv C_1.\end{aligned}$$

By the dominant convergence theorem, this infinite product converges absolutely and uniformly for $|\xi| \le 2^{L_0}\xi_0$.

Next, from (2.5),

$$P_K(e^{-i2\xi}) = \cos^{2K}\xi + \sin^{2K}\xi. \tag{2.16}$$

It is easy to see that its minimum is at $\xi = \pi/4$ with minimal value $2^{-(K-1)}$. This yields $\max_{\xi\in R}|1/P_K(e^{-i2\xi})| = 2^{K-1}$. Hence,

$$\prod_{\ell=1}^{L}\left|\frac{1}{P_K(e^{-i2\xi/2^\ell})}\right| \le 2^{(K-1)L} \le \left(\frac{2|\xi|}{\xi_0}\right)^{K-1}.$$

Therefore, we obtain $\prod_{\ell=1}^{\infty} 1/P_K(e^{-i2\xi/2^\ell}) = O(|\xi|^{K-1})$.

3. First, we show that $\tilde{\phi}(\xi_1+i\xi_2)$ is analytic for $|\xi_2| < \sigma$. We claim that $\tilde{\phi}(\xi_1+i\xi_2)$ is a meromorphic function with poles at:

$$2^\ell\left(\pm\pi/2 + i\ln|\tan\theta_k|\right),$$

for $\ell = 1, 2, \cdots$, $k = 1, \cdots, K$ for even K and for $k = 1, \cdots, K-1$ for odd K. To see this, from (2.13), the poles of $\tilde{\phi}$ are the roots of $\prod_{\ell=1}^{\infty} P_K(e^{-i2\xi/2^\ell})$. From (2.6), (2.7), the roots of $P_K(z^2) = 0$ are $z = e^{-i\xi} = \pm i\tan\theta_k$ ($k = 1, \cdots, K$ for even K and $k = 1, \cdots, K-1$ for odd K). That is, $\xi = \pm\pi/2 + i\ln|\tan\theta_k|$. The claim follows immediately. With this, the poles of $\widehat{\tilde{\phi}}(\xi)$ with the smallest imaginary part are $2\left(\pm\pi/2 \pm i\ln\tan\left(\frac{K+1}{4K}\right)\right)$. Therefore, $\widehat{\tilde{\phi}}(\xi_1 + i\xi_2)$ is analytic for $|\xi_2| < \sigma$.

Next, we show $e^{\sigma'|x|}\tilde{\phi} \in L^2(\mathbb{R})$ for any $\sigma' < \sigma$. When $x \ge 0$, we move the integration line in the Fourier inversion formula from the real line to $\{\xi_1 + i\sigma' \mid \xi_1 \in \mathbb{R}\}$:

$$\tilde{\phi}(x) = \frac{1}{\sqrt{2\pi}}\int e^{ix(\xi_1+i\sigma')}\tilde{\phi}(\xi_1 + i\sigma')\,d\xi_1.$$

Here, we have used $\tilde{r} \ge r$, (2.11) and the fact that $\tilde{\phi}$ is analytic for $|\xi| < \sigma$. From $\tilde{\phi}(\cdot \pm i\sigma') \in L^2(\mathbb{R})$ for $\tilde{r} \ge r$ and the Planchel equality, we obtain $e^{\sigma'|x|}\tilde{\phi} \in L^2(\mathbb{R})$. When $x < 0$, we can move the integration line to $\{\xi_1 - i\sigma' \mid \xi_1 \in \mathbb{R}\}$ and prove similarly. ∎

Definition 2.1 *1. A function ϕ is said to satisfy the Riesz basis property if $\phi \in L^2(\mathbb{R})$ and there exist constants $0 < A, B < \infty$ such that for any finite sequence $\{c_k\}$ we have*

$$A \sum_k |c_k|^2 \leq \| \sum_k c_k \phi(\cdot - k) \|_{L^2}^2 \leq B \sum_k |c_k|^2.$$

2. Two refinable functions ϕ and $\tilde{\phi}$ are said to be biorthogonal if both of them satisfy the Riesz basis property and they are dual to each other, namely, $\int \phi(x - i)\, \tilde{\phi}(x - k)\, dx = \delta_{i,k}$.

It is known [20, 9, 6] that a refinable function ϕ with mask $h(z)$ satisfies the Riesz basis property if and only if $\phi \in L^2$ and $h(z)$ satisfies the Cohen criterion [6]. That is, there exist a compact set K and finite many disjoint closed intervals K_i, associated with an integer n_i such that

(1) K contains 0 as an interior point,

(2) $K = \cup_i K_i$ and $[-\pi, \pi] = \cup(2n_i\pi + K_i)$, and $2n_i\pi + K_i$ can intersect to each other at most at their boundaries,

(3) $h(e^{i\xi/2^j}) \neq 0$ for all $j > 0$ and for all $\xi \in K$.

Proposition 2 *The refinable functions ϕ and $\tilde{\phi}$ constructed from the difference wavelet method are biorthogonal for $\tilde{r} \geq r \geq 1$.*

Proof. Since ϕ is a spline for $r \geq 1$, it is in $L^2(\mathbb{R})$. It also satisfies Cohen's criterion trivially with $K = [-\pi, \pi]$. For $\tilde{\phi}$, we notice from (2.16) that $1/P(e^{i2\xi}) > 0$ for $\xi \in [-\pi, \pi]$. Hence, $\tilde{\phi}$ also satisfies Cohen's criterion trivially with $K = [-\pi, \pi]$. From (2.11), we see that $\tilde{\phi} \in L^2(\mathbb{R})$ as $\tilde{r} \geq r$. The duality of ϕ and $\tilde{\phi}$ follows from (1.5) [9]. ∎

Theorem 2.1 *The difference wavelets ψ and $\tilde{\psi}$ with $\tilde{r} \geq r \geq 1$ are biorthogonal in $L^2(\mathbb{R})$.*

Proof. Our theorem follows from a theorem of Chui [5] which says that ψ and $\tilde{\psi}$ are biorthogonal if and only if ϕ and $\tilde{\phi}$ are biorthogonal and $g(z) = \tilde{h}(-z)P(z^2)$ and $\tilde{g}(z) = h(-z)/P(z^2)$ with P being in the Wiener class. We choose $P = P_K$ here. $P_K(z)$ is a Laurent polynomial and, from (2.16), $P_K(z) \neq 0$ for all $|z| = 1$. Hence P_K is in the Wiener class. ∎

3 A Comparison Study

In this section, we compare four wavelet methods: Cohen-Daubechies-Feauveau wavelet (CDF), Daubechies orthogonal wavelet (Daub), Chui-Wang's semi-orthogonal wavelet (CW), and the difference wavelet (Diff). Natural criteria of a good wavelet method are fast, stable and efficient. Therefore our comparisons include

(1) operation counts,

(2) condition numbers,

(3) compression ratio for data sequence and matrices.

Roughly speaking, the results below show that the difference wavelet method is more efficient than other method. In addition, we also compare

(4) regularity,

(5) approximation power,

(6) approximation error for functions with jumps,

(7) length of "essential support,"

to demonstrate that this better efficiency is probably due to its better regularity with slightly bigger "essential support." It is surprising that even for non-smooth functions, Doff has better representation in the sense that it is smoother and less overshoots and undershoots.

3.1 Operation Counts

In this comparison, we shall show that the operation cost of Diff is about the same as those of CDF or Daub, even though it is a rational filter. We compute the number of operations per each datum in a one-level wavelet transform. The operation counts here include both forward and inverse transforms (i.e. (1.1) and (1.2)). The common part of the four methods is $\left(\frac{1+z}{2}\right)^{2K}$. A factorization of Laurant polynomials does not change its operation counts. Thus, the differences among these four methods are the operations for $Q_K(z)$, $G_K(z)/G_K(z^2)$ and $1/P_K(z)$. The inversion of $1/P_K(z^2)$ and $1/G_K(z^2)$ are performed by the cyclic reduction method. Table 1 shows that Diff has about the same number of operations as those of CDF or Daub, while CW has more.

3.2 Condition numbers

We compute the condition number of $T_J : c_J \mapsto (c_0, d_0, \cdots, d_{J-1})$ by matlab to study the sensitivity of various wavelet transforms. The matrix size is 1024×1024. Table 2 shows the following things.

Diff	CDF, Daub	CW
$1/P_K(z)$	$Q_K(z)$	$G_K(z)/G_K(z^2)$
$< (4A + 3M)[K/2]$	$(2A + 1M)K$	$(6A + 5M)K$

Table 1: Comparison of operation counts for performing filters $1/P_K(z)$ (Diff), $Q_K(z)$ (CDF and Daub) and $G_K(z)/G_K(z^2)$ (CW) per each datum. Here, A is the addition operation and M is the multiplication.

1. It is clear that orthogonal wavelet transform has the smallest condition number, which is 1.

2. Among the biorthogonal wavelet methods considered here, the difference wavelet method has the smallest condition numbers for the cases $r = \tilde{r}$. In general, the condition numbers of difference wavelet transforms are reasonable small for application.

3. Notice that the condition numbers of CDF are big for the cases $r = \tilde{r} \geq 4$. This is because the corresponding $\tilde{\phi}$'s are not in $L^2(\mathbb{R})$, and the corresponding wavelets does not form a Riesz basis in $L^2(\mathbb{R})$.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	5.2	2.1	–	–
2	4	3.5	2.5	–	–
3	3	3.1	9.1	5.1	1.0
1	7	10.1	2.4	–	–
2	6	7.0	2.5	–	–
3	5	5.5	5.9	–	–
4	4	4.5	35.4	10.0	1.0
1	9	19.5	2.5	–	–
2	8	14.0	2.5	–	–
3	7	11.0	5.5	–	–
4	6	8.6	14.7	–	–
5	5	7.0	154.9	19.3	1.0

Table 2: Comparison of the condition numbers of various wavelet transforms T_J : $c_J \to (c_4, d_4, \ldots, d_{J-1})$. Here, r is the average order, $\tilde{r}$ is the difference order and $J = 10$. The matrix size is 1024×1024. The results are computed by Matlab.

3.3 Compression ratio for data sequences

We demonstrate by numerical tests to show that Diff method does have better compression ratio for data sequences (both smooth and nonsmooth) and matrices (smooth kernel and singular kernel).

We measure the efficiency of a wavelet representation for a data sequence c_J by a quantity C_2 defined by

$$C_2(\epsilon, J, r, \tilde{r}) = N_2(\epsilon, J, r, \tilde{r})/N, \tag{3.1}$$

where $N = 2^J$ is the total number of data. The number N_2 is defined as follows. Firstly, we transform the discrete data c_J to $(c_0, d_0, \cdots, d_{J-1})$. Next, we truncate $(c_0, d_0, \cdots, d_{J-1})$ by a threshold δ to yield $(\bar{c}_0, \bar{d}_0, \cdots, \bar{d}_{J-1})$. The threshold δ is chosen so that the inverse transform of $(\bar{c}_0, \bar{d}_0, \cdots, \bar{d}_{J-1})$ (denoted by $\bar{c}_J$) is within ϵ neighborhood of c_J in ℓ^2, i.e.

$$\left(\sum_i (c_{J,i} - \bar{c}_{J,i})^2 2^{-J}\right)^{1/2} \leq \epsilon.$$

Then we define $N_2(\epsilon, J, r, \tilde{r})$ to be the number of nonzero elements in $(\bar{c}_0, \bar{d}_0, \cdots, \bar{d}_{J-1})$.

We perform two tests: one is a smooth data, the other is a nonsmooth data. Namely, we choose $c_{J,i} = u(2^{-J}i)$, $i = 1, \cdots, 2^J$, where $u(x) = \sin 4\pi x + \sin 6\pi x$ for the first test, and $u(x) = \chi_{[0,1/2]}(x)$ for the second test.

Table 3 and 4 are the value of C_2 corresponding to the above two tests, where $N = 2^{10}$ and $\delta = 10^{-6}$. We should compare various methods with fixed K, because they have about the same amount of operation cost). We observe that the Diff with $r = 1$ is the best in both tests.

3.4 Compression ratio for matrices

Fast matrix-vector multiplication is important in many applications. Following Beylkin-Coifman-Rokhlin [2], we use "standard method" for matrix compression. That is, given a $2^J \times 2^J$ matrix G, we transform it to $T_J G T_J^t$. A truncation is applied to every entries of $T_J G T_J^t$ with threshold δ. The number N_2 is the total number of nonzero entries. The error of $\bar{G}$ (i.e. the inverse transform of the truncated $T_J G T_J^t$) is measured by a matrix ℓ^2 norm. The quantity C_2 is defined by N_2/N. Roughly speaking, C_2 is the number of operation needed per each datum for a matrix-vector multiplication. Notice that the compression ratio mentioned in other literatures is N^2/N_2 which is N/C_2 is our language.

We perform three tests. The first one is the heat kernel on periodic domain. The second one is a singular kernel which is basically the Green's function of the Laplacian in 2-d. The third one is the matrix which converts the coefficients of finite Chebyshev expansion into the coefficients of a finite Legendre expansion.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	0.105	0.119	–	–
2	4	0.203	0.330	–	–
3	3	0.423	0.916	0.423	0.746
1	7	0.057	0.063	–	–
2	6	0.061	0.111	–	–
3	5	0.113	0.213	–	–
4	4	0.223	0.455	0.111	0.234
1	9	0.031	0.053	–	–
2	8	0.031	0.061	–	–
3	7	0.057	0.100	–	–
4	6	0.061	0.143	–	–
5	5	0.117	0.262	0.057	0.119

Table 3: Comparison of C_2, where $C_2 = N_2/N$, N_2 is the number of nonzero truncated wavelet coefficients, N is the total number of data. The test data is $c_{J,k} = u(2^{-J}k)$, where $u(x) = \sin 4\pi x + \sin 6\pi x$. Here, $N = 2^{10}$, the threshold $\delta = 10^{-6}$, r, the averaging order and $\tilde{r}$, the differencing order.

3.4.1 Heat kernel

We evaluate the integral:

$$u(x,t) = \int_{-1}^{1} G(x,y,t) f(y) dy$$

where

$$G(x,y,t) = \sum_{\ell=-\infty}^{\infty} \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{[2\pi(x-y-\ell)]^2}{4t}\right),$$

that is, the fundamental solution of the heat equation on the periodic domain $[0,1]$. This integral appears commonly for solving convection-diffusion equations [8]. In our test, we choose $t = 0.1$, and the kernel is obtained by summing the heat kernel over 20 periods. We discretize the above integral by trapezoidal rule on a uniform grid. Table 5 is the quantity C_2 and Table 6 is the corresponding L^2-errors and the thresholds. In the comparison, we fix the threshold δ used for the Diff. The thresholds used for other methods are chosen so that the corresponding errors are no less than the errors of the Diff produced by using the fixed threshold δ. This is to guarantee that the Diff produces the least error in our test.

The result shows that the difference wavelet method with $r = 1$ is most efficient method.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	0.031	0.059		
2	4	0.031	0.061		
3	3	0.033	0.061	0.266	0.059
1	7	0.059	0.082		
2	6	0.059	0.084		
3	5	0.057	0.084		
4	4	0.057	0.084	0.334	0.082
1	9	0.059	0.102		
2	8	0.059	0.104		
3	7	0.057	0.107		
4	6	0.061	0.107		
5	5	0.059	0.105	0.387	0.094

Table 4: Comparison of the C_2 for non-smooth data: $c_{J,k} = u(2^{-J}k)$ and $u(x) = \chi_{[0,1/2)}$, $N = 2^{10}$, $\delta = 10^{-6}$.

3.4.2 Singular kernel

Singular kernel is often appeared in boundary integral methods. Here, we consider the kernel corresponding to the Laplacian in 2-dimension [2]. Namely,

$$\Lambda_{ij} = \begin{cases} \ln(i-j)^2 & \text{if } i \neq j, \\ 0 & \text{if } i = j. \end{cases} \tag{3.2}$$

Table 7 is the value of C_2 and Table 8 is the corresponding errors ϵ and thresholds δ. The result shows that the difference wavelet method is the most efficient method.

3.4.3 Fast Legendre transform

It is well-known that a fast Legendre transform can be achieved through a fast Chebyshev transform followed a compressed matrix which converts the coefficients of a finite Chebyshev expansion into the coefficients of a finite Legendre expansion of the same polynomial [1, 2]. The matrix is given by

$$\Lambda_{ij} = \begin{cases} \frac{1}{\pi}\Lambda^2(j) & \text{if } 0 = i \leq j < N, \\ \frac{2}{\pi}\Lambda(j-i)\Lambda(j+i) & \text{if } 0 < i \leq j < N, \\ 0 & \text{otherwise.} \end{cases}$$

Table 9 is the value of C_2 and Table 10 is the corresponding errors and thresholds. The result shows that the difference wavelet method is again the most efficient method.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	1.86	4.59		
2	4	4.41	12.22		
3	3	14.23	50.88	15.19	30.31
1	7	0.72	1.95		
2	6	1.50	2.00		
3	5	1.86	6.38		
4	4	4.31	19.22	2.48	4.98
1	9	0.70	1.09		
2	8	0.75	2.13		
3	7	0.72	2.16		
4	6	1.59	5.38		
5	5	1.86	14.59	2.5	2.09

Table 5: This table is the values of C_2, the number of operations per each datum in matrix-vector multiplication. The matrix G is a finite approximation of the fundamental solution of heat equation over period domain $(0,1)$. The matrix size is 1024×1024. The corresponding errors and thresholds are tabulated in Table 6 The result demonstrates that the Diff method with $r = 1$ is the most efficient method.

3.5 Regularity

In this subsection, we compare the smoothness of $\tilde{\phi}$ for various methods. We shall find s_0 such that $\tilde{\phi} \in H^s(\mathbb{R})$ for all $s < \tilde{s}$. Here, $H^s(\mathbb{R})$ denotes for the Sobolev space of regularity order s.

Firstly, for semi-orthogonal wavelet, $\tilde{r} = K$ and $\tilde{\phi}$ is the spline $1^{*K}_{[0,1)}$. Hence, $\tilde{s} = \tilde{r} - 1/2$. For other methods, basically we compute the spectral radius of the transition matrix to determine its regularity. First, we use the following lemma [9, 20] to normalize our comparison.

Lemma 3.1 *Suppose* $\tilde{h}(z) = \left(\frac{1+z}{2}\right)^{\tilde{r}} \tilde{H}(z)$ *with* $\tilde{H}(-1) \neq 0$ *and* $\tilde{H}(1) = 1$. *Let* $\tilde{\phi}$ *and* $\tilde{\Phi}$ *be the scaling functions associated with* $\tilde{h}(z)$ *and* $\tilde{H}(z)$. *Then* $\tilde{\phi} \in H^{s+\tilde{r}}(\mathbb{R})$ *if and only if* $\tilde{\Phi} \in H^s(\mathbb{R})$.

The proof of this lemma follows easily from

$$\widehat{\tilde{\phi}}(\xi) = e^{-i\tilde{r}\xi/2} \left(\frac{\sin \frac{\xi}{2}}{\frac{\xi}{2}} \right)^{\tilde{r}} \widehat{\tilde{\Phi}}(\xi),$$

see [20]. Now, we apply this lemma to our cases:

$$\tilde{h}_{\mathrm{CDF}}(z) = \left(\frac{1+z}{2} \right)^{\tilde{r}} \tilde{H}_{\mathrm{CDF}}(z),$$

$$\tilde{h}_{\text{Diff}}(z) = \left(\frac{1+z}{2}\right)^{\tilde{r}} \tilde{H}_{\text{Diff}}(z).$$

Thus, we only need to compare the regularity of the scaling functions $\tilde{\Phi}_{\text{CDF}}$ and $\tilde{\Phi}_{\text{Diff}}$ associated with $\tilde{H}_{\text{CDF}}$ and $\tilde{H}_{\text{Diff}}$, respectively. Table 11 gives the regularity for these functions. It is basically quoted from Cohen-Daubechies [7]. They gave a sharp estimates on s_0 by using transition matrix method. The case of Diff in the first row is the Butterworth case in their paper. See also Fan and Sun [13].

We can transfer this table to a regularity table for $\tilde{\phi}$, see Table 12. We observe that

1. Diff is more regular than CDF and Daub.
2. With $\tilde{r}$ fixed, $\tilde{\phi}_{CW}$ is the B-spline, which is the most regular function with given $\tilde{r}$. However, with K fixed, which means it takes about same amount of time to take wavelet transformation, Diff with $r = 1$ is even more regular than CW.

3.6 Approximation power

Given any function $u \in \mathrm{L}^2(\mathbb{R})$, it can be approximated by the projection

$$\mathcal{P}^J u := \sum_k (u, \phi_{J,k}) \tilde{\phi}_{J,k}$$

as $J \to \infty$ [9]. When $u \in \mathrm{L}^2(\mathbb{R}) \cap \mathrm{C}^{\tilde{r}+1}(\mathbb{R})$, this approximation has the following sharp estimate [17, 18, 19]:

$$\|\mathcal{P}^J u - u\|_{\mathrm{L}^2} = C_{\tilde{\phi}} 2^{-\tilde{r}J} \|u^{(\tilde{r})}\|_{\mathrm{L}^2} + O(2^{-(\tilde{r}+1)J}) \text{ as } J \to \infty,$$

where

$$C_{\tilde{\phi}} = \frac{1}{\tilde{r}!} \left(\sum_{k \neq 0} |\widehat{\tilde{\phi}}^{(\tilde{r})}(2k\pi)|^2 \right)^{1/2}.$$

The smaller $C_{\tilde{\phi}}$ is, the better the approximation power is. Following Unser [19], the constant $C_{\tilde{\phi}}$ has the following expression:

$$C_{\tilde{\phi}} = \begin{cases} \frac{Q_K(-1)}{\sqrt{(4^{\tilde{r}}-1)2^{\tilde{r}}}} D & \text{for CDF} \\ \frac{1/P_K(1)}{\sqrt{(4^{\tilde{r}}-1)2^{\tilde{r}}}} D & \text{for difference} \\ \frac{1}{\sqrt{(4^K-1)2^K}} D & \text{for Chui-Wang} \\ \frac{Q(-1)}{\sqrt{(4^K-1)2^K}} D & \text{for Daub} \end{cases}$$

$$D = \left(\sum_{k \in \mathbb{Z}} |\widehat{\tilde{\phi}}((2k+1)\pi)|^2 \right)^{1/2}.$$

In Table 13, we compute $C_{\tilde{\phi}}$ for various methods. We observe the following things.

1. When $r = \tilde{r} = K$ fixed, CW has the best approximation power and Diff is the second. This is due to that the approximate space $\tilde{V}^J$ of CW is spanned by splines which is the most regular scaling function with given K.

2. However, if we fix K (this means the amount of works for wavelet transform is of the same order), then we see Diff has the best approximation power. This table is consistent to the previous regularity table.

3.7 Approximation error for non-smooth functions

When u is not smooth, the representation $\mathcal{P}^J u = \sum (u, \phi_{J,k}) \tilde{\phi}_{J,k}$ exhibits the Gibbs phenomena. In the test below, we choose the test function to be $\chi_{[1/4,3/4]}$ on the periodic domain $[0, 1]$. We compare the L^1 and L^2 errors and the heights of overshoots and undershoots. In this comparison, the coefficients $(u, \phi_{J,k})$ are computed as follows. For Diff and CDF, ϕ is the splines, we compute (u, ϕ) exactly by using Maple. For CW, $\mathcal{P}^J u$ is the orthogonal projection of u onto the space spanned by splines, therefore we find the coefficients $c_{J,k} = (u, \phi_{J,k})$ by solving the linear system

$$\sum_k (\tilde{\phi}_{J,i} \tilde{\phi}_{J,k}) c_{J,k} = (u, \tilde{\phi}_{J,i}),$$

where $\tilde{\phi}_{J,k}$ is the splines. For Daub, we do not have an analytic formula for $\phi_{J,k}$. Therefore we compute $(u, \phi_{J,k})$ by direct numerical integration. We first transform the integral $(u, \phi_{J,k})$ to $2^{-J/2} \int u(2^{-J}(k + y)) \phi(y)\, dy$. Since u is a characteristic function, the integral only involves integration of ϕ. A trapezoidal rule is then applied for this integration using $N = 2^{13}$ grid points. Thus, the numerical error for this integration is of order $O(2^{-J/2-13})$, which is relatively small in comparing with the approximation errors.

Table 14–15 are the L^1 and L^2 norms of the error $\mathcal{P}^J u - u$. Here, $u = \chi_{[1/4,3/4]}$, $J = 10$ and the numerical errors are computed using 2^{13} grid points. Table 16 shows the heights of the corresponding overshoots or undershoots (i.e. $\max_{x \in (1/4,3/4)} \mathcal{P}^J u(x) - 1$, or $-\min_{x \notin [1/4,3/4]} \mathcal{P}^J u$). We do not show the results of CDF for the case $(3, 3)$, $(4, 4)$ and $(5, 5)$, because the corresponding errors are too large. In the comparison, for each $K = 3, 4, 5$, we choose the best case for each methods. For instance, for Diff, we choose $(r, \tilde{r}) = (3, 3), (2, 6), (2, 8)$. For CDF, we select $(r, \tilde{r}) = (1, 5), (2, 6), (2, 8)$. For CW and Daub, we have no other choices but $(r, \tilde{r}) = (3, 3), (4, 4), (5, 5)$. It is surprising that the two non-compact supported wavelet methods, Diff and CW, have the least errors of overshoots and undershoots. Indeed, Diff is slightly better than CW. It only has 9% overshoots or undershoots, which is about the same as those using finite Fourier method. Figure 1 are the corresponding graphes of $\mathcal{P}^J u$ in the interval $[0.24, 0.26]$. We observe that Diff does have better representation in the sense of smoothness and smaller overshoots/undershoots.

3.8 Support

We have shown that $\tilde{\phi}_{Diff}$ decays exponentially in previous section. Below, the length of the region where $|\tilde{\phi}(x)| \geq 10^{-3}$. Table 17 is a comparison of this "essential support." We observe that the essential support of $\tilde{\phi}_{Diff}$ is about twice of the support of $\tilde{\phi}_{CDF}$. Figures 2, 3 are the graphs of $\tilde{\phi}$, ψ and $\tilde{\psi}$ of the difference wavelet method for $K = 3$ and 4, respectively, with various r. One observes that their "essential supports" are indeed quite compact.

References

[1] B.K. Alpert and V. Rokhlin, A fast algorithm for the evaluation of Legendre expansions, *SIAM J. Sci. Stat. Comput.*, **12** No. 1 (1991) 158-179.

[2] G. Beylkin, R. Coifman, and V. Rokhlin, Fast wavelet transforms and numerical algorithms I, *Communications on Pure and Applied Mathematics*, Vol. XLIV, (1991) 141–183

[3] C. K. Chui, An Introduction of Wavelets, Academic Press, San Diego, CA (1992).

[4] C.K Chui and J. Z. Wang, A cardinal spline approach to wavelets, *Proc. Amer. Math. Soc.* **113** (1991) 785-793.

[5] Charles K. Chui, Wavelets: A Mathematical Tool for Signal Analysis, SIAM Monographs on Mathematical Modeling and Computation, 1997. (1992).

[6] A. Cohen, *Ondelettes, analyses multirésolutions et traitement numérique du signals,* thesis, Université Paris IX Dauphine, Dauphine, France, (1990).

[7] A. Cohen and I. Daubechies, A new technique to estimate the regularity of refinable functions, Revista Matemática Iberoamericana, Vol. 12, No. 2, (1996), 527-593.

[8] B. Enquist, S. Osher and S. Zhong, *Fast wavelet based algorithms for linear evolution equations,* SIAM J. Sci. Comput. Vol. **15**, No. 4, (1994), 755-775.

[9] A. Cohen, I. Daubechies and J. Feauveau, Biorthogonal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **45** (1992) p.485-560.

[10] W. Dahmen, Stability of multiscale transformations, *J. Fourier Analysis and Applications.*

[11] Daubechies, Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **41** (1988) p.909-996.

[12] Daubechies, Ten Lectures on Wavelets, CBMS-NSF Regional Conf. Series in Appl. Math. Vol. 61, SIAM, Philadelphia, PA (1992).

[13] A. H. Fan and Qiyu Sun, Regularity of Butterworth refinable functions, preprint, Laboratoire d'Analyse Numérique, Université de Pierre et Marie Curie, 1997.

[14] G. Golub and C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, (1984)

[15] A. Oppenheim and R. Schaefer, Digital Signal Processing, Prentice Hall, New York, 1975.

[16] G. Strang and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press (1996).

[17] G. Strang and G. Fix, A Fourier analysis of the finite element variational method, in *Constructive Aspects of Fundamental Analysis* (1971) 796-830.

[18] W. Sweldens and R. Piessens, Quadrature formulae and asymptotic error expansions for wavelet approximations of smooth functions, I, II, *SIAM J. Numer. Anal.* **31** (1994) p.1240-1264, *Numer. Math.* **68** (1994) p.377-401.

[19] M. Unser, Approximation power of biorthogonal wavelet expansions, *IEEE Trans. SP.* **44** No. 3 (1996) p.519-527.

[20] L. Villemoes, Energy moments in time and frequency of two-scale difference equation solutions and wavelets, *SIAM J. Math. Anal.* **23** (1992) p.1519-1543.

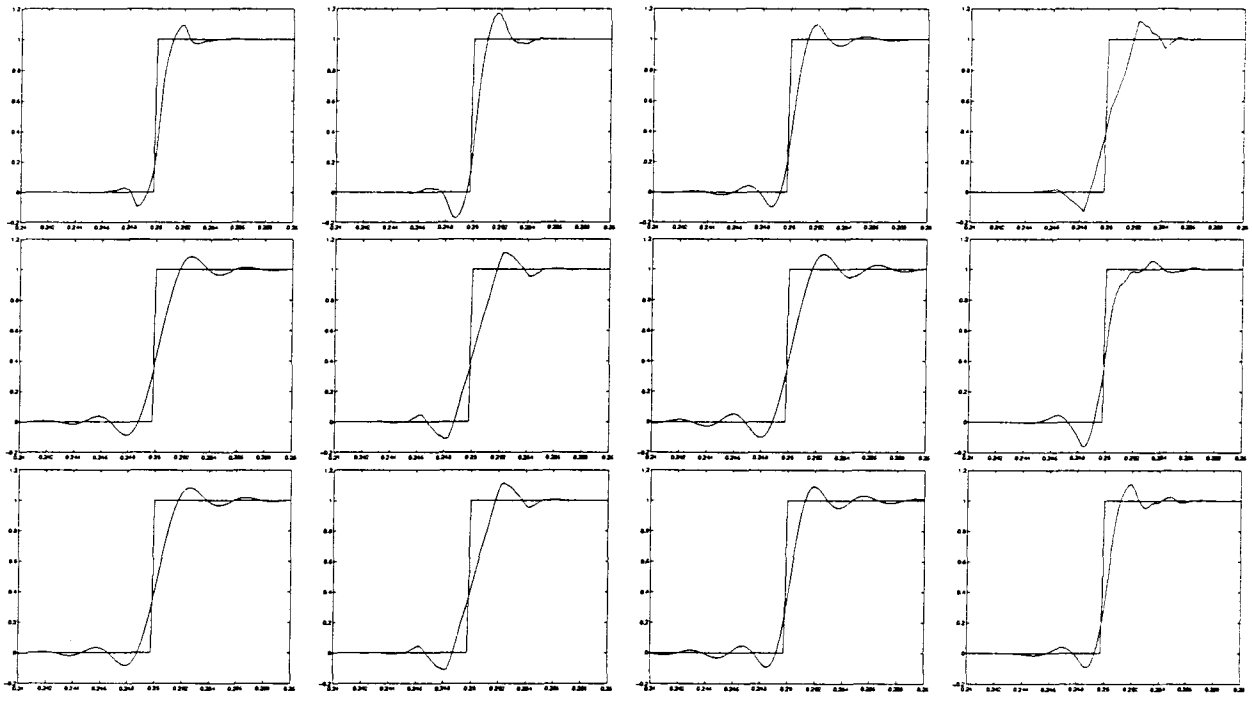


Figure 1: The graphs of $\mathcal{P}^J u$ on the interval $[0.24, 0.25]$, where $u = \chi_{[1/4,3/4]}$ and $J = 10$. The graphs from left to right are the representations by Diff, CDF, CW and Daub, respectively. For each method, the graph from top to bottom corresponding to the best case for each $K = 3, 4, 5$. More precisely, from top to bottom, $(r, \tilde{r}) = (3, 3)$, $(2, 6)$, $(2, 8)$ for the Diff, $(r, \tilde{r}) = (1, 5)$, $(2, 6)$, $(2, 8)$ for CDF, and $r = \tilde{r} = K$ for both CW and Daub.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	1.4e-07	4.2e-07		
2	4	5.2e-07	2.0e-06		
3	3	2.3e-06	4.0e-06	2.9e-06	2.6e-06
1	7	1.1e-08	1.2e-07		
2	6	6.4e-07	2.9e-06		
3	5	3.3e-07	5.3e-07		
4	4	1.2e-06	2.0e-06	1.5e-06	2.0e-06
1	9	1.1e-08	1.9e-07		
2	8	9.1e-10	6.7e-08		
3	7	2.6e-08	8.9e-07		
4	6	1.5e-06	2.0e-06		
5	5	1.1e-06	5.3e-06	1.5e-06	1.4e-06

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	1e-06	2e-06		
2	4	1e-06	1e-06		
3	3	1e-06	1e-06	5e-06	1e-06
1	7	1e-06	1e-06		
2	6	1e-06	1.7e-05		
3	5	1e-06	1e-06		
4	4	1e-06	1e-06	3.9e-05	1e-05
1	9	1e-06	1e-06		
2	8	1e-06	1e-06		
3	7	1e-06	2e-06		
4	6	1e-06	1e-06		
5	5	1e-06	1e-06	3e-05	1e-05

Table 6: The top subtable is the L^2-error ϵ of the truncated matrix $\bar{G}$ in Table 5. The bottom one is the threshold δ used for truncation.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	48.7	64.6		
2	4	55.0	85.1		
3	3	75.9	149.4	141.7	86.5
1	7	50.3	68.2		
2	6	49.0	71.1		
3	5	48.5	81.4		
4	4	55.8	128.1	136.8	75.5
1	9	57.0	63.5		
2	8	54.5	70.6		
3	7	47.6	75.4		
4	6	46.9	89.4		
5	5	49.0	121.4	134.2	68.9

Table 7: Compression ratio C_2 for the matrix (defined in equation 3.2) , the matrix size is $N = 1024$. The result shows that the Diff is the most efficient one.

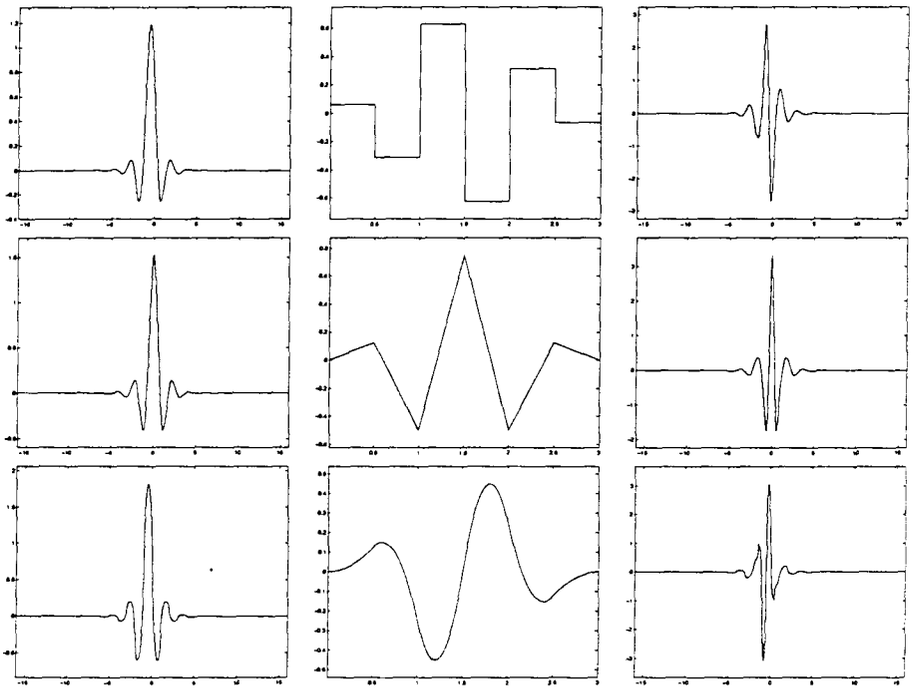


Figure 2: The graphs of $\tilde{\phi}$, ψ and $\tilde{\psi}$ (left to right) of the difference wavelet for $K = 3$ and $r = 1, 2, 3$ (top to bottom).

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	1.1e-06	1.3e-06		
2	4	1.0e-06	1.2e-06		
3	3	2.1e-06	2.5e-06	2.2e-06	2.3e-06
1	7	3.4e-06	3.6e-06		
2	6	2.9e-06	2.9e-06		
3	5	3.7e-06	3.9e-06		
4	4	3.6e-06	5.8e-06	4.2e-06	3.8e-06
1	9	1.5e-05	1.9e-05		
2	8	5.1e-06	5.2e-06		
3	7	5.8e-06	6.0e-06		
4	6	5.1e-06	5.7e-06		
5	5	9.3e-06	2.6e-05	1.6e-05	9.7e-06
1	5	1e-06	5e-06		
2	4	1e-06	2e-06		
3	3	1e-06	1e-06	6.4e-05	7e-06
1	7	1e-06	1.9e-05		
2	6	1e-06	7e-06		
3	5	1e-06	4e-06		
4	4	1e-06	1e-06	0.000256	1.9e-05
1	9	1e-06	7.5e-05		
2	8	1e-06	2.1e-05		
3	7	1e-06	6e-06		
4	6	1e-06	3e-06		
5	5	1e-06	1e-06	0.001024	4.5e-05

Table 8: The matrix error ϵ and the thresholds for the matrix (defined in equation 3.2) in Table 7

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	29.3	39.6		
2	4	28.1	42.3		
3	3	29.1	55.3	69.4	40.1
1	7	39.1	37.3		
2	6	33.3	32.1		
3	5	29.6	46.6		
4	4	28.2	62.1	73.3	41.4
1	9	44.0	31.4		
2	8	35.7	28.3		
3	7	31.3	43.8		
4	6	29.4	49.2		
5	5	28.5	67.4	82.3	39.5

Table 9: Compression ratio C_2 for matrix which converts the finite Chebyshev expansion to finite Legendre expansion. The matrix size is 1024×1024. The result shows that the different wavelet method has the best compression ratio (see the cases $(r, \tilde{r}) = (2, 4)$, $(4, 4)$ and $(5, 5)$.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	8.8e-07	1.0e-06		
2	4	1.0e-06	1.6e-06		
3	3	1.6e-06	2.2e-06	1.8e-06	1.9e-06
1	7	5.4e-06	5.7e-06		
2	6	5.5e-06	5.9e-06		
3	5	3.0e-06	4.0e-06		
4	4	3.5e-06	7.8e-06	4.9e-06	3.7e-06
1	9	1.8e-05	1.8e-05		
2	8	1.5e-05	1.5e-05		
3	7	7.6e-06	8.0e-06		
4	6	6.5e-06	6.6e-06		
5	5	7.7e-06	6.0e-05	1.41e-05	7.8e-06
1	5	1e-06	4e-06		
2	4	1e-06	3e-06		
3	3	1e-06	1e-06	3.2e-05	6e-06
1	7	1e-06	1.6e-05		
2	6	1e-06	1.3e-05		
3	5	1e-06	3e-06		
4	4	1e-06	1e-06	0.000128	1.2e-05
1	9	1e-06	6.1e-05		
2	8	1e-06	3.9e-05		
3	7	1e-06	8e-06		
4	6	1e-06	3e-06		
5	5	1e-06	1e-06	0.000512	2.5e-05

Table 10: The top subtable is the errors ϵ and the bottom is the thresholds δ corresponding to Table 9.

K	2	3	4	5	6	7
s_0(Diff)	-1.5	-1.9	-2.3	-2.7	-3.1	-3.5
s_0(CDF)	-2.0	-3.2	-4.4	-5.8	-7.2	-8.6
s_0(Daub)	-1.0	-1.6	-2.2	-2.9	-3.6	-4.3

Table 11: $K = (r+\tilde{r})/2$, $\prod Q_K(e^{-i\xi/2^\ell}) \in H^s(\mathbb{R})$ for all $s < s_0$(CDF), and $\prod 1/P_K(e^{-2i\xi/2^\ell}) \in H^s(\mathbb{R})$ for all $s < s_0$(Diff), quoted from [7]. Those with less negative values are of better regularity.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	3.1	1.8	–	–
2	4	2.1	0.8	–	–
3	3	1.1	-0.2	2.5	1.4
1	7	4.7	2.6	–	–
2	6	3.7	1.6	–	–
3	5	2.7	0.6	–	–
4	4	1.7	-0.4	3.5	1.8
1	9	6.3	3.2	–	–
2	8	5.3	2.2	–	–
3	7	4.3	1.2	–	–
4	6	3.3	0.2	–	–
5	5	2.3	-0.8	4.5	2.1

Table 12: Comparison of regularity of $\tilde{\phi}$. The table is the value of $\tilde{s}$, where $\tilde{\phi} \in H^s(\mathbb{R})$ for all $s < \tilde{s}$.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	1.1e-03	1.1e-02	–	–
2	4	6.8e-03	6.8e-02	–	–
3	3	4.3e-02	6.2e-01	5.8e-03	3.0e-01
1	7	6.8e-05	2.4e-03	–	–
2	6	4.3e-04	1.5e-02	–	–
3	5	2.7e-03	9.8e-02	–	–
4	4	1.7e-02	1.8e+00	9.1e-04	5.6e-01
1	9	4.2e-06	5.3e-04	–	–
2	8	2.7e-05	3.3e-03	–	–
3	7	1.7e-04	2.1e-02	–	–
4	6	1.1e-03	1.6e-01	–	–
5	5	6.6e-03	1.1e+01	1.4e-04	1.3e+00

Table 13: Comparison of the constant $C_{\tilde{\phi}}$ in the estimate of the approximation power. The result shows that with K fixed, the difference wavelet method has smallest constant $C_{\tilde{\phi}}$, i.e. the best approximation power.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	0.0045	0.0034	–	–
2	4	0.0046	0.0050	–	–
3	3	0.0027	–	0.0034	0.0046
1	7	0.0053	0.0038	–	–
2	6	0.0045	0.0047	–	–
3	5	0.0036	0.0031	–	–
4	4	0.0052	–	0.0050	0.0034
1	9	0.0057	0.0041	–	–
2	8	0.0045	0.0046	–	–
3	7	0.0042	0.0031	–	–
4	6	0.0052	0.0153	–	–
5	5	0.0044	–	0.0041	0.0033

Table 14: The L^1 error of $u - \mathcal{P}^J u$, where $u = \chi_{[1/4,3/4]}$ on the periodic domain $[0,1]$ and $J = 10$.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	0.00074	0.00072	–	–
2	4	0.00077	0.00083	–	–
3	3	0.00067	–	0.00068	0.00077
1	7	0.00077	0.00073	–	–
2	6	0.00074	0.00080	–	–
3	5	0.00068	0.00072	–	–
4	4	0.00078	–	0.00075	0.00060
1	9	0.00078	0.00074	–	–
2	8	0.00073	0.00078	–	–
3	7	0.00070	0.00068	–	–
4	6	0.00073	0.00190	–	–
5	5	0.00071	–	0.00069	0.00069

Table 15: The L^2 error of $u - \mathcal{P}^J u$, where $u = \chi_{[1/4,3/4]}$ on the periodic domain $[0,1]$ and $J = 10$.

r	$\tilde{r}$	Diff	CDF	CW	Daub
1	5	0.19	0.17	–	–
2	4	0.10	0.19	–	–
3	3	0.09	–	0.10	0.13
1	7	0.20	0.18	–	–
2	6	0.09	0.11	–	–
3	5	0.10	0.13	–	–
4	4	0.14	–	0.10	0.16
1	9	0.20	0.19	–	–
2	8	0.09	0.10	–	–
3	7	0.12	0.11	–	–
4	6	0.13	1.00	–	–
5	5	0.12	–	0.09	0.11

Table 16: This table is the magnitudes of overshoot or undershoot of $\mathcal{P}^J u$, where $u = \chi_{1/4,3/4]}$ and $J = 10$.

r	$\tilde{r}$	Diff	CDF	Daub
1	5	12.8	6.2	–
2	4	13.0	6.6	–
3	3	13.0	6.7	4.3
1	7	15.4	7.9	–
2	6	17.0	8.2	–
3	5	18.6	8.5	–
4	4	18.8	9.6	5.2
1	9	19.3	8.5	–
2	8	21.1	8.7	–
3	7	22.8	9.5	–
4	6	23.0	11.0	–
5	5	24.8	12.7	6.2

Table 17: Comparison of the length of the "essential support"(where $|\tilde{\phi}(x)| \geq 10^{-3}$)

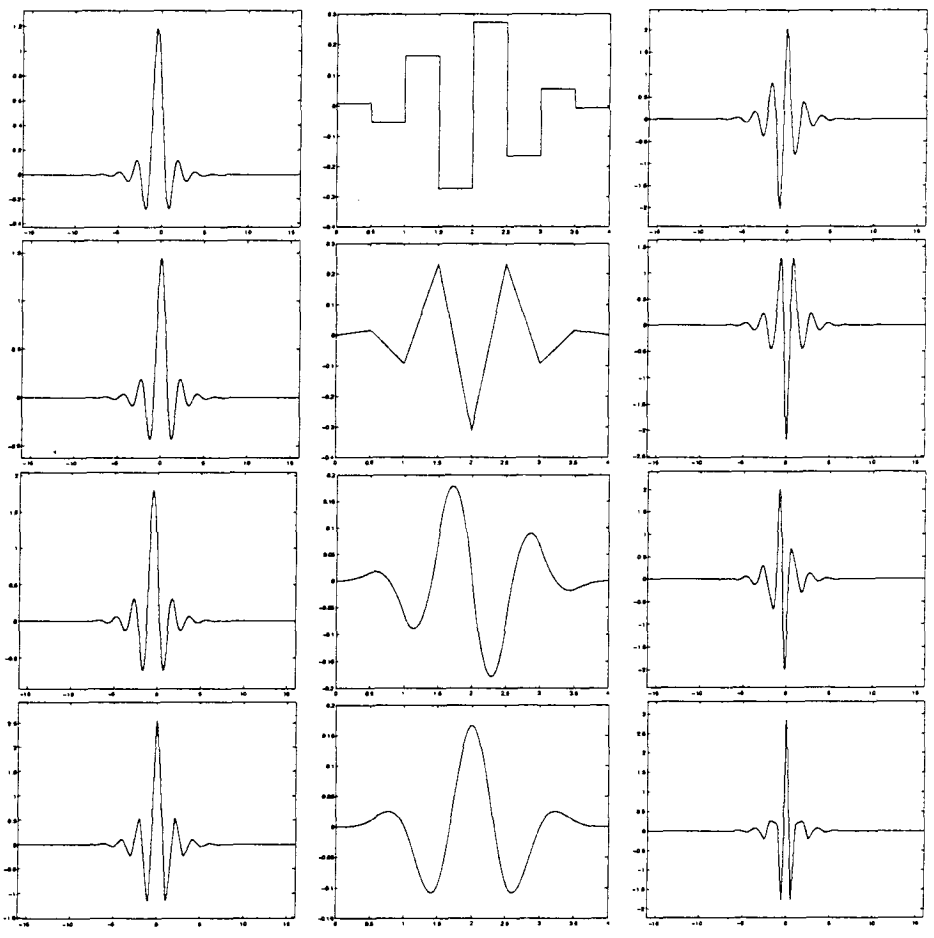


Figure 3: The graphes of $\tilde{\phi}$, ψ and $\tilde{\psi}$ (left to right) of the difference wavelet for $K = 4$ and $r = 1, 2, 3, 4$ (top to bottom).

Wavelets On Interval Over Regular Or Irregular Meshes *

I-Liang Chern † Chien-Chang Yen ‡

Abstract

We construct compact supported biorthogonal wavelets on intervals with underlying meshes being uniform or nonuniform. At the first step, a sequence of fundamentals are constructed based on the Lagrange interpolation through a subdivision scheme. These fundamentals are interpolating and supported inside the interval. The boundary fundamentals are constructed based on one-side interpolation instead. At the second step, two multi-resolution analyses are constructed. Its primals consist of B-splines, while the duals are the summations of derivatives of the above fundamentals. The biorthogonality of these two multi-resolution are followed by a commutation formula. A pair of biorthogonal wavelets are thus constructed. Riesz basis property is investigated. Regularity of these wavelets over irregular meshes is also studied for the case when the above Lagrange interpolation is cubic. Application on fast Poisson solver on box by using these wavelets will be given.

Biorthogonal wavelets $(\psi, \tilde{\psi})$ on R

- wavelet ψ and dual wavelet $\tilde{\psi}$ are compactly supported, local oscillatory functions:

$$\int x^m \psi(x)\, dx = 0, \ \forall\ 0 \le m \le \tilde{r}$$

$$\int x^m \tilde{\psi}(x)\, dx = 0, \ \forall\ 0 \le m \le r$$

*This work was supported by the National Science Council of the Republic of China under Contract NSC87-2115-M002-001 MS.

†Department of Mathematics, National Taiwan University, Taipei, Taiwan

‡Department of Mathematics, National Taiwan University, Taipei, Taiwan

- Riesz bases: $\psi_{j,i} := 2^{j/2}\psi(2^j x - i)$, $\tilde{\psi}_{j,i} := 2^{j/2}\tilde{\psi}(2^j x - i)$ Then $\{\psi_{j,i}\}_{j,i\in Z}$ and $\{\psi_{j,i}\}_{j,i\in Z}$ constitute Riesz bases in $L^2(R)$

$$\begin{aligned}\sum_{j,i} |d_{j,i}|^2 &\sim \|\sum_{j,i} d_{j,i}\psi_{j,i}\|_{L^2}^2 \\ \sum_{j,i} |d_{j,i}|^2 &\sim \|\sum_{j,i} d_{j,i}\tilde{\psi}_{j,i}\|_{L^2}^2\end{aligned}$$

- Duality:

$$(\psi_{j,i}, \tilde{\psi}_{j',k'}) = \delta_{j,j'}\delta_{i,i'}$$

Hence

$$\begin{aligned} f &= \sum_{j,i} (f, \psi_{j,i})\tilde{\psi}_{j,i} \\ &= \sum_{j,i} (f, \tilde{\psi}_{j,i})\psi_{j,i}\end{aligned}$$

Advantages of Biorthogonal Wavelets

- A Tool for Space-Scale Analysis:
  - $(f, \psi_{j,i})$ characterize local regularity of f
  - characterization of function space:
    - Sobolev space H^s:

$$\sum_{j,i} 2^{2sj} |(f, \psi_{j,i})|^2 < \infty$$

- An efficient tool for representing functions

$$f \sim \sum_{j,i} (f, \psi_{j,i}) \tilde{\psi}_{j,i}$$

  Many of the wavelet coefficients $(f, \psi_{j,i})$ are negligible.

- An efficient tool to respresent operators:
  - pseudo-differential operators
    - differential operators
    - smooth integral operators
    - Calderón-Zygmund operators
  - The matrix representation of them are "nearly" diagonal (sparse)
  - easy for preconditioning
- Fast wavelet transform is available.

A Simple Application

- Problem:

$$\begin{aligned} -u'' &= f, \quad x \in (0,1) \\ u(0) &= u(1) = 0 \end{aligned}$$

- Representation of u:

$$u \sim \sum_{k=1}^{2^j-1} u(x_{j,k}) \tilde{\phi}_{j,k}^{[0]}$$

  boundary condition is satisfied.

- Projection of equation:

$$(u'' + f, \phi_{j,k}^{[2]}) = 0, \ k = 1, \cdots 2^j - 1$$

- The discrete linear system: $AU = F$, where

$$\begin{aligned} A &= \text{diag}(-1, 2, -1) = D^2 \\ U &= (u(x_{j,k}))_{k=1}^{2^j-1} \\ F &= ((f, \phi_{j,k}^{[2]}))_{k=1}^{2^j-1} \end{aligned}$$

- Wavelet transforms:
  - Domain: $\tilde{W}^{[0]} : \tilde{\phi}^{[0]} \mapsto \tilde{\psi}^{[0]}$
  - Range: $\tilde{W}^{[2]} : \tilde{\phi}^{[2]} \mapsto \tilde{\psi}^{[2]}$

- From

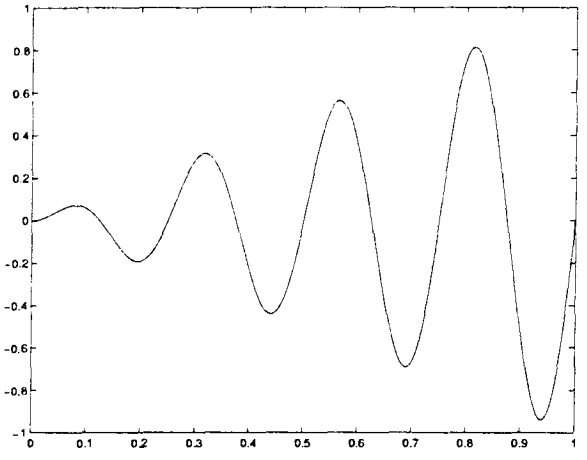
$$\begin{aligned} \frac{d^2}{dx^2} \tilde{\phi}^{[0]} &= D^2 \tilde{\phi}^{[2]} \\ \frac{d^2}{dx^2} \tilde{\psi}^{[0]} &= \tilde{\psi}^{[2]} \end{aligned}$$

  We obtain

$$(u(x_{j,k})) = (\tilde{W}^{[0]})^{-1} \tilde{W}^{[2]} ((F_k))$$

- Spectral accuracy: $\tilde{\phi}^{[0]}$ can be arbitrarily high order

Example: $-u'' = f$



$$u(x) = x\sin(8\pi x)$$

j	$K=1$	$K=2$	$K=3$	$K=4$
4	2.0000e-03	8.9173e-04	4.9675e-04	8.5731e-04
5	5.0778e-04	6.2223e-05	1.3497e-05	2.3870e-06
6	1.2822e-04	3.4720e-06	1.8147e-07	1.6794e-08
7	3.2134e-05	2.0896e-07	2.0843e-09	4.2802e-11
8	8.0376e-06	1.2965e-08	2.8382e-11	1.0537e-13
9	2.0060e-06	8.2074e-10	4.2747e-13	1.9069e-15

L^2-error

j	$K=1$	$K=2$	$K=3$	$K=4$
4	-8.9999	-10.1311	-10.9752	-10.1879
5	-10.9435	-13.9722	-16.1770	-18.6764
6	-12.9291	-18.1358	-22.3938	-25.8274
7	-14.9255	-22.1903	-28.8378	-34.4435
8	-16.9248	-26.2008	-35.0362	-43.1096
9	-18.9272	-30.1824	-41.0892	-48.8977

$\log_2(L^2\text{-error})$

Goal:

- **To construct biorthogonal wavelets $\{\psi_{j,i}\}$ and $\{\tilde{\psi}_{j,i}\}$ on $[0,1]$ with the above properties.**

Motivation

- To construct wavelets on boxes, general domains, and manifolds
- to develop spectral-element like methods for solving PDEs

Remarks.

- Many previous works for wavelets on $[0,1]$
  - Meyer (1992): Orthogonal wavelets
  - Chui-Quak (1992): Spline wavelets
  - Cohen-Daubechies-Vial (1993): Orthogonal wavelets
  - Andersson-Hall-Jawerth-Peters (1993): Orthogonal & biorthogonal wavelets
  - Jouini-Lemarié-Rieusset (1993): Biorthogonal
  - Dahmen-Kunoth-Urban (1997): Biorthogonal
- Yet, this one is
  - *simple*: no biorthogonalization
  - *natural*: the primal $\phi_{j,i}$ are just the B-splines
  - can be extended to *nonuniform grids*

Standard procedure to construct biorthogonal wavelets

(i) construct two multiresolution analyses $\{V_j\}_{j\in Z}$ and $\{\tilde{V}_j\}_{j\in Z}$:

- nested:

$$\begin{aligned} \cdots \subset V_{j-1} &\subset V_j \subset \cdots \subset L^2(R) \\ \cdots \subset \tilde{V}_{j-1} &\subset \tilde{V}_j \subset \cdots \subset L^2(R) \end{aligned}$$

- complete:

$$\begin{aligned} \cap_j V_j &= \{0\},\ \overline{\cup_j V_j} = L^2(R) \\ \cap_j \tilde{V}_j &= \{0\},\ \overline{\cup_j \tilde{V}_j} = L^2(R) \end{aligned}$$

- translation and dilation invariance:

$$\begin{aligned} f(\cdot) &\in V_0 \Leftrightarrow f(\cdot - k) \in V_0,\ \forall k \in Z \\ f(\cdot) &\in V_0 \Leftrightarrow f(2^j \cdot) \in V_j,\ \forall j \in Z \end{aligned}$$

- Riesz bases: $\exists\ \phi$ and $\tilde{\phi}$ such that

  $\{\phi_{j,i}\}_i$ is a Riesz basis in V_j,

  where $\phi_{j,i}(x) := 2^{j/2}\phi(2^j x - i)$

  Also $\{\tilde{\phi}_{j,i}\}_i$ is a Riesz basis in $\tilde{V}_j$,

$$\begin{aligned} A\sum_i |c_i|^2 &\le \|\sum_i c_i \phi_{j,i}\|_{L^2}^2 \le B\sum_i |c_i|^2 \\ \tilde{A}\sum_i |c_i|^2 &\le \|\sum_i c_i \tilde{\phi}_{j,i}\|_{L^2}^2 \le \tilde{B}\sum_i |c_i|^2 \end{aligned}$$

(ii) V_j is dual to $\tilde{V}_j$ and $(\phi_{j,i}, \tilde{\phi}_{j,k}) = \delta_{i,k},\ \forall j$

(iii)

$$V_j = V_{j-1} \oplus W_{j-1},\ \tilde{V}_j = \tilde{V}_{j-1} \oplus \tilde{W}_{j-1}$$

$$W_{j-1} \perp \tilde{V}_{j-1},\ \tilde{W}_{j-1} \perp V_{j-1},\ \forall j$$

$$L^2(R) = \bigoplus_j W_j = \bigoplus_j \tilde{W}_j$$

(iv) $\{\psi_{j,i}\}$ and $\{\tilde{\psi}_{j,i}\}$ are dual Riesz bases in $L^2(R)$:

$$(\psi_{j,i}, \tilde{\psi}_{\ell,k}) = \delta_{j,\ell}\delta_{i,k}$$

Construction of ϕ, $\tilde{\phi}$, ψ and $\tilde{\psi}$

1. Construct scaling functions: from

   $V_0 \subset V_1$, $\tilde{V}_0 \subset \tilde{V}_1$, we have

$$\begin{aligned} \phi(x) &= 2\sum_k h_k \phi(2x-k) \\ \tilde{\phi}(x) &= 2\sum_k \tilde{h}_k \tilde{\phi}(2x-k) \end{aligned}$$

   ϕ and $\tilde{\phi}$ can be constructed by cascade algorithm or subdivision scheme.

   E.g. Let, $h(z) = \sum_k h_k z^k$

$$\begin{aligned} h(z) &= \left(\frac{1+z}{2}\right)^r \\ \phi(x) &= \underbrace{1_{[0,1)} * \cdots * 1_{[0,1)}}_{r \text{ times}} \end{aligned}$$

2. **Wavelets**: from $W_0 \subset V_1$ and $\tilde{W}_0 \subset \tilde{V}_1$:

$$\begin{aligned} \psi(x) &= 2\sum_k g_{k-1} \phi(2x-k) \\ \tilde{\psi}(x) &= 2\sum_k \tilde{g}_{k-1} \tilde{\phi}(2x-k) \end{aligned}$$

General Procedure to construct Filter banks

1. Find Laurent polynomials $h(z)$ and $\tilde{h}(z)$ such that $h(1) = \tilde{h}(1) = 1$ and

$$h(z)\tilde{h}(z^{-1}) + h(-z)\tilde{h}(-z^{-1}) = 1$$

2. Define g and $\tilde{g}$ by

$$\begin{aligned} g(z) &= \tilde{h}(-z) \\ \tilde{g}(z) &= h(-z) \end{aligned}$$

Design criteria for filter banks

- **Fast**: the following are equivalent
  - The filter banks $\{h_k\}_k$, etc. have only finite many nonzero elements
  - ϕ, etc. are compactly supported
  - discrete wavelet transform and its inverse are of linear complexity
- **Stable**:
  - the following are equivalent
    - $\mathcal{W}_j$ and $\mathcal{W}_j^{-1}$ are bounded, independent of j
    - $\{\psi_{j,i}\}$ and $\{\tilde{\psi}_{j,i}\}$ are dual Riesz bases in $L^2(R)$
  - need some regularity condition for ϕ and $\tilde{\phi}$
- **Efficient**:
  - If $u \in H^s$, then

$$\|u - \tilde{Q}_J u\|_{L^2} = O(2^{-Js})\|u\|_{H^s}, \text{ for all } s \leq \tilde{r}$$

  - need polynomial exactness:

$$\pi_{\tilde{r}-1} \subset \tilde{V}_j$$

    where $\pi_{\tilde{r}-1}$ is the set of polynomials of degree $< \tilde{r}$.

Cohen-Daubechies-Feauveau (CDF) wavelet

an important example of biorthogonal wavelets:

1. Given a parameter $K > 0$, find polynomial $Q(z)$ of smallest degree such that $H(z) = z^{-K}\left(\frac{1+z}{2}\right)^{2K} Q(z)$ satisfies

$$H(z) + H(-z) = 1.$$

   Then $Q = Q_K$:

$$Q_K(z) = \sum_{n=0}^{K-1} \binom{K+n-1}{n} \left(\frac{2 - z - z^{-1}}{4}\right)^n$$

2. Split H into $h(z)\tilde{h}(z^{-1})$ by

$$H(z) = \underbrace{\left(\frac{1+z}{2}\right)^r}_{h(z)} \underbrace{z^{-K}\left(\frac{1+z}{2}\right)^{\tilde{r}} Q_K(z)}_{\tilde{h}(z^{-1})}$$

   where $r + \tilde{r} = 2K$.

3. $\{\phi_{j,i}\}_{i\in Z}$ and $\{\tilde{\phi}_{j,i}\}_{i\in Z}$ are Riesz bases if $\tilde{r} \geq r$.

4. ϕ is the splines

$$\phi = \phi^{[r]} := \underbrace{1_{[0,1)} * \cdots * 1_{[0,1)}}_{r \text{ times}}$$

5. polynomial exactness: $\pi_{\tilde{r}-1} \subset \tilde{V}_j$

6. $(\phi_{j,i}, \tilde{\phi}_{j,k}) = \delta_{i,k}$

7. $\tilde{\phi}$ has the shortest support that has the above property

Multi-resolution Analysis

- Define $\phi_{j,i} := 2^{j/2}\phi(2^j x - i)$, etc., and

$$\begin{aligned} V_j &:= \text{span } \{\phi_{j,i}\}_{i\in Z}, \quad \tilde{V}_j := \text{span } \{\tilde{\phi}_{j,i}\}_{i\in Z} \\ W_j &:= \text{span } \{\psi_{j,i}\}_{i\in Z}, \quad \tilde{W}_j := \text{span } \{\tilde{\psi}_{j,i}\}_{i\in Z} \end{aligned}$$

- Under mild regularity condition, we have

  (i) $\{V_j\}_j$ and $\{\tilde{V}_j\}_j$ constitute two multi-resolution analyses:

  * nested:

$$\begin{aligned} \cdots \subset V_{j-1} &\subset V_j \subset \cdots \\ \cdots \subset \tilde{V}_{j-1} &\subset \tilde{V}_j \subset \cdots \end{aligned}$$

  * complete:

$$\begin{aligned} \cap_j V_j &= \{0\}, \overline{\cup_j V_j} = L^2(R) \\ \cap_j \tilde{V}_j &= \{0\}, \overline{\cup_j \tilde{V}_j} = L^2(R) \end{aligned}$$

  * Riesz bases:

    $\{\phi_{j,i}\}_i$ is a Riesz basis in V_j,

    $\{\tilde{\phi}_{j,i}\}_i$ is a Riesz basis in $\tilde{V}_j$,

$$\begin{aligned} A\sum_i |c_i|^2 &\le \|\sum_i c_i \phi_{j,i}\|_{L^2} \le B\sum_i |c_i|^2 \\ \tilde{A}\sum_i |c_i|^2 &\le \|\sum_i c_i \tilde{\phi}_{j,i}\|_{L^2} \le \tilde{B}\sum_i |c_i|^2 \end{aligned}$$

  (ii) V_j is dual to $\tilde{V}_j$ and

$$(\phi_{j,i}, \tilde{\phi}_{j,k}) = \delta_{i,k}, \ \forall j$$

(iii)

$$V_j = V_{j-1} \oplus W_{j-1}, \tilde{V}_j = \tilde{V}_{j-1} \oplus \tilde{W}_{j-1}$$

$$W_{j-1} \perp \tilde{V}_{j-1}, \ \tilde{W}_{j-1} \perp V_{j-1}, \ \forall j$$

$$L^2(R) = \bigoplus_j W_j = \bigoplus_j \tilde{W}_j$$

(iv) $\{\psi_{j,i}\}$ and $\{\tilde{\psi}_{j,i}\}$ are dual Riesz bases in $L^2(R)$:

$$(\psi_{j,i}, \tilde{\psi}_{\ell,k}) = \delta_{j,\ell}\delta_{i,k}$$

- Wavelet decomposition:

$$\begin{aligned}
\tilde{Q}_j u &= \sum_i (u, \phi_{j,i})\tilde{\phi}_{j,i} \\
&= \sum_i (u, \phi_{j-1,i})\tilde{\phi}_{j-1,i} + \sum_i (u, \psi_{j-1,i})\tilde{\psi}_{j-1,i} \\
&= \sum_i (u, \phi_{0,i})\tilde{\phi}_{0,i} + \sum_{\ell=0}^{j-1} \sum_i (u, \psi_{\ell,i})\tilde{\psi}_{\ell,i} \\
u &= \lim_{j\to\infty} \tilde{Q}_j u \\
&= \sum_i (u, \phi_{0,i})\tilde{\phi}_{0,i} + \sum_{\ell=0}^{\infty} \sum_i (u, \psi_{\ell,i})\tilde{\psi}_{\ell,i} \\
&= \sum_{\ell=-\infty}^{\infty} \sum_i (u, \psi_{\ell,i})\tilde{\psi}_{\ell,i}
\end{aligned}$$

Wavelet transform

- discrete wavelet transform:

$$\mathcal{W}_j : (u_{j,i})_i \longmapsto ((u_{0,i})_i, (w_{0,i})_i, (w_{1,i})_i, \cdots, (w_{j-1,i})_i)$$

  where $u_{j,i} = (u, \phi_{j,i})$ and $w_{j,i} = (u, \psi_{j,i})$.

- Fast wavelet transform:

$$\begin{aligned} u_{j-1,i} &= \sqrt{2} \sum_k h_k u_{j,2i+k} \quad \text{(low-pass)} \\ w_{j-1,i} &= \sqrt{2} \sum_k g_k u_{j,2i+1+k} \quad \text{(high-pass)} \end{aligned}$$

- Inverse wavelet transform:

$$u_{j,i} = \sqrt{2} \sum_k [\tilde{h}_{i-2k} u_{j-1,k} + \tilde{g}_{i-2k-1} w_{j-1,k}]$$

func	supp	dim (interior)	dim (restricted)
ϕ	$[0, r]$	$2^j - r + 1$	$2^j + r - 1$
$\tilde{\phi}$	$[0, 2K - 2 + \tilde{r}]$	$2^j - 2K - 1 + \tilde{r}$	$2^j + 2K - 3 + \tilde{r}$
ψ	$[0, 4K - 2]$	$2^j - 4K - 1$	$2^j + 4K - 3$
$\tilde{\psi}$	$[0, 4K - 2]$	$2^j - 4K - 1$	$2^j + 4K - 3$

Problems for wavelets on $[0, 1]$

- Given r and $\tilde{r}$ with $\tilde{r} \geq r$ and $r + \tilde{r} = 2K$. Let.

$$\begin{aligned} \mathbf{v}_j[0,1] &= \text{span } \{\phi_{j,i} | \text{ support } \phi_{j,i} \subset [0,1]\} \\ V_j[0,1] &= \text{span } \{\phi_{j,i} | \text{ support } \phi_{j,i} \cap (0,1) \neq \phi\} \end{aligned}$$

We have

- dimension problem
  1. dim $\mathbf{v}_j[0,1] \neq$ dim $\tilde{\mathbf{v}}_j[0,1]$,
     dim $V_j[0,1] \neq$ dim $\tilde{V}_j[0,1]$
  2. dim $\mathbf{w}_j[0,1] \neq$ dim $\tilde{\mathbf{w}}_j[0,1]$,
     dim $W_j[0,1] \neq$ dim $\tilde{W}_j[0,1]$,
  3. dim W_j, $\tilde{W}_j$ should be 2^j
- nested problem:
  1. $V_j[0,1] \subset V_{j+1}[0,1]$
     $\tilde{V}_j[0,1] \subset \tilde{V}_{j+1}[0,1]$
  2. $\mathbf{v}_j[0,1] \not\subset \mathbf{v}_{j+1}[0,1]$
     $\tilde{\mathbf{v}}_j[0,1] \not\subset \tilde{\mathbf{v}}_{j+1}[0,1]$

- Previous approaches: want to find $\{\phi^b_{j,i}\}$, $\{\tilde{\phi}^b_{j,i}\}$ such that

  1. $V_j = \mathbf{v}_j[0,1] +$ span $\{\phi^b_{j,i}\}$
     $\tilde{V}_j = \tilde{\mathbf{v}}_j[0,1] +$ span $\{\tilde{\phi}^b_{j,i}\}$
  2. dim V_j = dim $\tilde{V}_j$
  3. $V_j \subset V_{j+1}$, $\tilde{V}_j \subset \tilde{V}_{j+1}$
  4. Duality $(\phi_{j,i}, \tilde{\phi}_{j,k}) = \delta_{i,k}$
  5. V_j contains splines of order $r-1$.
  6. $\tilde{V}_j$ has $\tilde{r}$ polynomial exactness
     (i.e. reproduce polynomials)
  7. Riesz basis property

The approach of Dahmen-Kunoth-Urban

1. Inserting $\{1, x, \cdots, x^{r-1}\}$ into subspace of $\mathbf{v}_j[0,1]$:

   (a) Define left boundary scaling functions $\phi^L_{j,-p}$, $0 \le p < r$

$$\begin{aligned} \phi^L_{j,-p} &:= \sum_{m<i_L} (x^p, \tilde{\phi}_{j,m}) \phi_{j,m}|_{[0,1]} \\ &= x^p - \sum_{m \ge i_L} (x^p, \tilde{\phi}_{j,m}) \phi_{j,m}|_{[0,1]} \end{aligned}$$

   (b) Define left boundary scaling functions $\tilde{\phi}^L_{j,-p}$, $0 \le p < \tilde{r}$

$$\begin{aligned} \tilde{\phi}^L_{j,-p} &:= \sum_{m<\tilde{i}_L} (x^p, \phi_{j,m}) \tilde{\phi}_{j,m}|_{[0,1]} \\ &= x^p - \sum_{m \ge \tilde{i}_L} (x^p, \phi_{j,m}) \tilde{\phi}_{j,m}|_{[0,1]} \end{aligned}$$

   (c) i_L and $\tilde{i}_L$ are chosen such that

$$\dim V_j = \dim \tilde{V}_j$$

2. Perform biorthogonalization.

Disadvantage

- difficult to compute $(u, \phi_{j,k})$ for boundary scaling function
  1. perform $(u, \phi_{j,m}|_{[0,1]})$
  2. compute $(u, \phi^L_{j,-p})$
  3. perform biorthogonalization
- No intuitive idea about $(u, \psi_{j,i})$ for boundary wavelets

Present approach

1. $V_j = V_j[0,1]$

   hence

   (a) $\phi_{j,k}$ are the splines restricted on $[0,1]$

   (b) $(u, \phi_{j,k})$ are local spline averaging

2. Interior wavelets are the CDF wavelets (which have the smallest supports)

3. Construct the duals $\tilde{\phi}_{j,k}$ using two techniques

   (a) subdivision algorithm

   (b) commutation formula

4. No biorthogonalization, simple filter coefficients

5. $(u, \psi_{j,k})$ are the interpolation error of the integrated function

6. Works also on **nonuniform grid**

Sketch our Method (for left boundary)

1. Given r, $\tilde{r}$ with $r \leq \tilde{r}$ and $r + \tilde{r} = 2K > 0$. We start to construct auxiliary functions

   - $\phi_{j,i}^{[0]} = \delta(\cdot - x_{j,i})$
   - $\tilde{\phi}_{j,i}^{[0]}$: constructed by subdivision scheme based on Lagrange or Hermite interpolation, with boundary condition

     $$\tilde{\phi}_{j,i}^{[0]}(0) = \cdots = \frac{d^{r-1}}{dx^{r-1}}\tilde{\phi}_{j,i}^{[0]}(0) = 0$$

     and such that

     $$(\tilde{\phi}_{j,i}^{[0]}, \phi_{j,k}^{[0]}) = \delta_{i,k}$$

2. One can construct auxiliary functions $\phi_{j,i}^{[s]}$ and $\tilde{\phi}_{j,i}^{[s]}$ for $s = 1, \cdots r$ by commutation formula

   $$\begin{aligned} \frac{d}{dx}\phi_{j,i}^{[s]} &= \phi_{j,i}^{[s-1]} - \phi_{j,i+1}^{[s-1]}, \quad \phi[s]_{j,i}(\infty) = 0 \\ \frac{d}{dx}\tilde{\phi}_{j,i}^{[s-1]} &= \tilde{\phi}_{j,i-1}^{[s]} - \tilde{\phi}_{j,i}^{[s]} \end{aligned}$$

   Then

   $$(\tilde{\phi}_{j,i}^{[s]}, \phi_{j,k}^{[s]}) = \delta_{i,k}$$

3. $\phi_{j,i}^{[r]}$ and $\tilde{\phi}_{j,i}^{[r]}$ are what we want.

4. It turns out that $\phi_{j,i}^{[r]}$ are the B-splines restricted on $[0, 1]$.

Subdivision Scheme Interior case

- Given a hierarchical grid systems $X_j = \{x_{j,k}\}_{k \in \Lambda_j}$ such that
  - $x_{j+1,k} = x_{j,2k}$
  - homogeneous:

$$\sup_{j,k} \frac{\max(\Delta^{[1]}_{j,k+1}, \Delta^{[1]}_{j,k-1})}{\Delta^{[1]}_{j,k}} < \infty$$

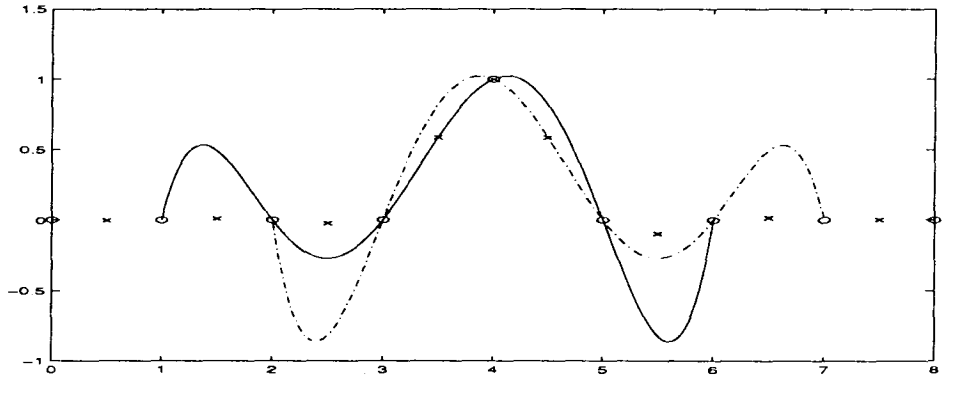
  where $\Delta^{[r]}_{j,k} = x_{j,k+r} - x_{j,k}$

- We can construct a function $\tilde{\phi}$ by

$$\begin{aligned} \tilde{\phi}(x_{0,k}) &= \delta_{0,k} \\ \tilde{\phi}(x_{j+1,k}) &= \sum_{m=-K+1}^{K-1} L_{j,k,m}(x_{j+1,k}) \tilde{\phi}(x_{j,k+m}) \\ L_{j,k,m}(x) &= \prod_{n=-K+1, n \neq m}^{K-1} \frac{x - x_{j,k+n}}{x_{j,k+m} - x_{j,k+n}} \end{aligned}$$

- For uniform grids, it is known that $\tilde{\phi}^{[0]} \in C^{K-\epsilon}$ and

$$(\tilde{\phi}^{[0]}_{j,i}, \delta(\cdot - x_{j,k})) = \delta_{i,k}$$



Subdivision scheme for left auxiliary function $\tilde{\phi}^{[0]}$

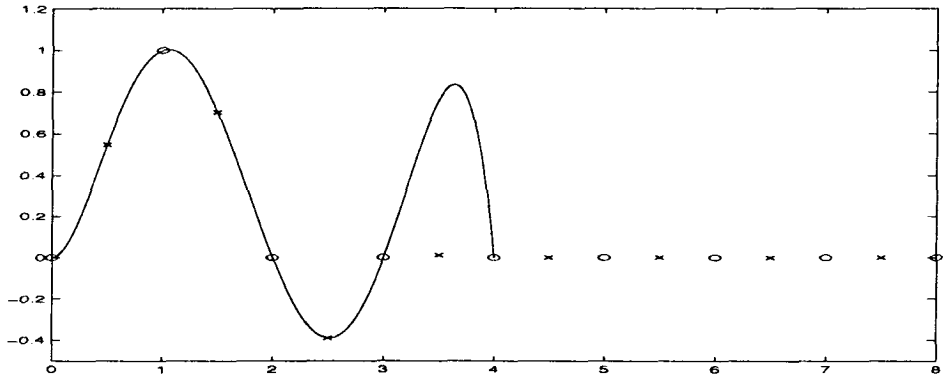
- $\tilde{\phi}^{[0]}_{j,i}$ is require to satisfy

  1. $\tilde{\phi}^{[0]}_{j,i}(x_{j,k}) = \delta_{i,k}$
  2. $\frac{d^s}{dx^s}\tilde{\phi}^{[0]}_{j,i}(0) = 0$ for $s = 0, \cdots, r-1$

- the subdivision scheme is based on $2K$-Lagrange interpolation:

$$\begin{aligned}
\tilde{\phi}^{[0]}_{j,i}(x_{j,k}) &= \delta_{i,k} \\
\tilde{\phi}^{[0]}_{j,i}(x_{\ell+1,k}) &= \sum_{m=-K+1}^{K-1} L_{\ell,k,m}(x_{\ell+1,k})\tilde{\phi}^{[0]}_{j,i}(x_{\ell,k+m}) \\
L_{j,k,m}(x) &= \prod_{n=-K+1, n\neq m}^{K-1} \frac{x - x_{j,k+n}}{x_{j,k+m} - x_{j,k+n}}
\end{aligned}$$

- $x_{j,0} = x_{j,-1} = \cdots = x_{j,-r+1} = 0$



The primal $\phi^{[0]}$

- $\phi_{j,i}^{[0]} := \delta(x - x_{j,i})$ for $i \geq 1$
- We have

$$(\phi_{j,i}^{[0]}, \tilde{\phi}_{j,k}^{[0]}) = \delta_{i,k}, \ i,k \geq 1$$

The auxiliary wavelets at $s = 0$

- We choose $\psi_{j,i}^{[0]}$ and $\tilde{\psi}_{j,i}^{[0]}$ so that

$$\begin{aligned}
(\psi_{j,i}^{[0]}, \tilde{\phi}_{j,k}^{[0]}) &= 0, i \geq 0, k \geq 1 \\
(\tilde{\psi}_{j,i}^{[0]}, \phi_{j,k}^{[0]}) &= 0, i \geq 0, k \geq 1 \\
(\psi_{j,i}^{[0]}, \tilde{\psi}_{j,k}^{[0]}) &= \delta_{i,k}, i,k \geq 0
\end{aligned}$$

- A natural choice is

$$\begin{aligned}
\psi_{j,k}^{[0]} &= \delta(x - x_{j+1,2k+1}) - \sum_{n \geq 1} \tilde{\phi}_{j,n}^{[0]}(x_{j+1,2k+1}) \delta(x - x_{j,n}) \\
\tilde{\psi}_{j,k}^{[0]} &:= \tilde{\phi}_{j+1,2k+1}^{[0]}
\end{aligned}$$

The filter banks

$$\begin{aligned}
h_{j,k,n}^{[0]} &= \delta_{2k,n} \\
\tilde{h}_{j,k,n}^{[0]} &= \tilde{\phi}_{j,k}^{[0]}(x_{j+1,n}) \\
g_{j,k,2\ell}^{[0]} &= -\tilde{h}_{j,\ell,2k+1} \\
g_{j,k,2\ell+1}^{[0]} &= \delta_{k,\ell} \\
\tilde{g}_{j,k,n}^{[0]} &= \delta_{2k_1,n}
\end{aligned}$$

Commutation formula uniform grid

- for primal scaling function: the following are equivalent

$$\begin{aligned} h^{[s]}(z) &= \left(\frac{1+z}{2}\right) h^{[s-1]}(z) \\ \frac{d}{dx}\phi_{j,i}^{[s]} &= \phi_{j,i}^{[s-1]} - \phi_{j,i+1}^{[s-1]} \end{aligned}$$

- for the dual $\tilde{\phi}$: the following are equivalent

$$\begin{aligned} \tilde{h}^{[s]}(z) &= \left(\frac{1+z}{2}\right)^{-1} \tilde{h}^{[s-1]}(z) \\ \frac{d}{dx}\tilde{\phi}_{j,i}^{[s-1]} &= \tilde{\phi}_{j,i-1}^{[s]} - \tilde{\phi}_{j,i}^{[s]} \end{aligned}$$

- for primal wavelet:

$$\begin{aligned} g^{[s]}(z) &= \left(\frac{1-z^{-1}}{2}\right)^{-1} g^{[s-1]}(z) \\ \frac{d}{dx}\psi_{j,i}^{[s-1]} &= -\psi_{j,i}^{[s]} \end{aligned}$$

- for dual wavelet:

$$\begin{aligned} \tilde{g}^{[s]}(z) &= \left(\frac{1-z}{2}\right) \tilde{g}^{[s-1]}(z) \\ \frac{d}{dx}\tilde{\psi}_{j,i}^{[s]} &= \tilde{\psi}_{j,i}^{[s-1]} \end{aligned}$$

Polynomial exactness: $\pi_{\tilde{r}}|_{[0,1]} \subset \tilde{V}_j$

1. $\tilde{\pi}^{[0]} := \left\{ \int_0^x (x-t)^{r-1} p(t)\, dt \;\mid\; p \in \pi_{\tilde{r}} \right\} \subset \pi_{2K}$

2. $\tilde{\pi}^{[0]} \subset \tilde{V}_j^{[0]}$

3. Given a function $u \in L^2[0,1]$, let

$$u^{[0]}(x) = \int_0^x (x-t)^{r-1} u(t)\, dt$$

Let

$$\begin{aligned} \tilde{Q}_j^{[0]} u^{[0]} &:= \sum_{k \geq 1} (u^{[0]}, \phi_{j,k}^{[0]}) \tilde{\phi}_{j,k}^{[0]} \\ \tilde{Q}_j u &:= \sum_{k \geq -r+1} (u, \phi_{j,k}^{[r]}) \tilde{\phi}_{j,k}^{[r]} \end{aligned}$$

Then

$$\frac{d^r}{dx^r} \left(\tilde{Q}_j^{[0]} u^{[0]} \right) = \tilde{Q}_j u$$

4. In particular, $\frac{d^r}{dx^r} \tilde{\pi}^{[0]} \subset \tilde{V}_j$

Regularity of $\tilde{\phi}^{[0]}$

- When the grid is uniform, $\tilde{\phi}^{[0]} \in C^{K-\epsilon}$ (Deslauriers and Dubuc)
- For nonuniform grid, $K = 2$, $\tilde{\phi}^{[0]} \in C^{2-\epsilon}$ (Daubechies,Guskov, Sweldens)
- The boundary scaling functions $\tilde{\phi}_{j,i}^{[0]}$ are polynomials near the boundary.
- The regularity of $\tilde{\phi}_{j,i}^{[r]}$ is the same as the regularity of CDF's

Riesz basis property

- $\{\psi_{j,i}^{[r]}\}_{j,i}$ and $\{\tilde{\psi}_{j,i}^{[r]}\}_{j,i}$ form a dual Riesz bases in $L^2[0,1]$ for $\tilde{r} \geq r$.

This mainly due to (BPX, Dahmen)

1. $\{\phi_{j,i}^{[r]}\}_i$ forms a Riesz basis in $V_j[0,1]$.
2. $\{\tilde{\phi}_{j,i}^{[r]}\}_i$ forms a Riesz basis in $\tilde{V}_j[0,1]$ for $\tilde{r} \geq r$.
3. Jackson inequality
4. Inverse inequality

Conclusion

1. We have constructed biorthogonal wavelets on $[0, 1]$.

2. Basic properties:

   - stability: Riesz bases properties
   - approximation ability
   - fast wavelet transform

3. What are special:

   - simple: no biorthogonalization is needed
   - natural:
     - the primal scaling functions are the B-splines restricted to $[0, 1]$
     - the interior dual scaling functions are the same as CDF
   - also work for nonuniform grids

4. The key ingredients are

   - commutation formula
   - subdivision scheme
   - Lagrange and Hermite interpolation

5. Useful for constructing wavelets on manifolds

ACCELERATION METHODS FOR TOTAL VARIATION-BASED IMAGE RESTORATION

QIANSHUN CHANG * and I-LIANG CHERN †

September 30, 2002

Abstract

In this paper, we apply a fixed point method to solve the total variation-based image restoration problem. An algebraic multigrid method is used to solve the corresponding linear equations. Krylov subspace acceleration is adopted to improve convergence in the fixed point iteration. A good initial guess for this outer iteration at finest grid is obtained by combining fixed point iteration and geometric multigrid interpolation successively from the coarsest grid to the finest grid. Numerical experiments demonstrate that this method is efficient and robust.

Keywords. Image restoration, total variation, fixed point method, algebraic multigrid method, Krylov acceleration, initialization

AMS subject classification. 65F30

1 Introduction

The image restoration problem is to recover a "true" image u from an observed image z. The latter is usually noisy and blurred, and is modeled by $z = Ku + n$. Here, K is a known linear blur operator, and n is a Gaussian white noise. In recent years, one main method for noise removal and deblurring is the total variation based restoration method, proposed by L. Rudin, S. Osher and E. Fatemi [14]. In this method, the total variation of u is used as a regularization penalty functional for a corresponding minimization problem (see (1.1) below). The main advantage of this method is that it can maintain sharpness of edges of images. The drawback is that the corresponding partial differential equation (PDE) is harder to solve because the degeneracy of the diffusion coefficients on edges. This paper is devoted to an efficient algorithm for solving this nonlinear PDE.

*Institute of Applied Mathematics, Academy of Mathematics and Systems Sciences, Chinese academy of Sciences, Beijing, China. Email: *cqs@amath8.amt.ac.cn*

†Department of Mathematics, National Taiwan University, Taipei, Taiwan. Email: *chern@math.ntu.edu.tw.*
Supported by grant NSC90-2115-M-002-020.

Using the Tikhonov penalty method and a diffusion regularization, the total variation based restoration method can be formulated as an unconstrained minimization problem:

$$\min_u \left(\alpha \int_\Omega \sqrt{|\nabla u|^2 + \beta}\, dxdy + \frac{1}{2}\|Ku - z\|_{L^2}^2 \right). \tag{1.1}$$

Here, $\alpha > 0$ is the penalty parameter and $\beta > 0$ is the regularization parameter and is usually small. The functional in (1.1) is strictly convex. Its global minimizer is unique. The well-posedness of problem (1.1) with $\beta \to 0+$ has been discussed in [1].

The corresponding Euler-Lagrange equation for (1.1) is

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) + K^*(Ku - z) = 0, \tag{1.2}$$

where K^* is the adjoint operator of K with respect to the L_2 inner product. Numerically, solving equation (1.2) is not an easy task for its nonlinearity and singularity in the diffusion term. Many methods have been proposed. The time marching scheme to reach the steady state of the corresponding parabolic equation of (1.2) was suggested in [3, 14]. An affine scaling algorithm was proposed in [10]. Vogel and Oman [17] applied a fixed point method to solve equation (1.2). Newton's method with a continuation procedure on the regularization parameter β was used in [6]. T. Chan, G. Golub, and P. Mulet [7] proposed a nonlinear primal-dual method. A multigrid method was proposed to solve the linearization part of equation (1.2) in [16, 12]. However, the convergence rate there was slow.

In this paper, we suggest to use the fixed point method [17] with Krylov's acceleration procedure instead of Newton's method, despite the latter is of quadratic convergence. From our experience, for practical images with reasonable accuracy, the number of iterations needed in Newton's method is about the same as that of a fixed point method with algebraic multigrid acceleration. The reason is that, the corresponding linear system in Newton's method is much harder to solve, because its diffusion coefficient (which is a matrix) is always degenerate in certain direction (the normal direction of the level set of the underlying function u), while the diffusion coefficient of (1.2) is a scalar, thus no preferential direction of degeneracy. Although this diffusion coefficient may vary dramatically, it can still be manageable. An algebraic multigrid (AMG) method is suitable to solve the corresponding linear system. We will show that the number of inner iterations needed is much less than that of a preconditioned conjugate gradient method used in Newton's method.

Multigrid methods have been successful for solving elliptic or parabolic problems numerically. The method is nearly optimal in the sense that the computational work required to achieve a fixed accuracy is proportional to the number of discrete unknowns [11]. The algebraic multigrid (AMG) method is designed to utilize the principle of the geometrically oriented multigrid (GMG) method to obtain a fast and automatic solution procedure for linear algebraic systems of equations (see [8, 9, 15]). This method is particularly suitable to our problem, where the coefficients may vary dramatically. In this paper, we adopt an improved version of ordinary AMG method [8]. The number of inner iterations needed is just one per each outer iteration. The convergent factor about 0.05. This is much better than Newton's method with PCG, where the numbers of inner iteration are between 4 to 7 [7].

By noticing the convexity of the functional (1.1), or equivalently, the monotonicity of equation (1.2), we adopt the Krylov subspace method [5, 13] to accelerate the outer iteration in our fixed point method. The improvement is about double in our numerical experiment.

A nice initial data in a fixed point iteration can reduce the number of iteration. Further, it can also allow us to choose a very small β. Thus, no continuation procedure is needed as that in [6]. To produce such a nice initial guess at finest grid, we first solve (1.2) at coarsest grid (by using the fixed point method with a direct solver), then we interpolate to the next fine grid level, then solve equation (1.2) again at that level with algebraic multigrid solver, and so on, until we reach the finest grid. In our experiment, the improvement of this acceleration is about 40%. But it can allow us to choose $\beta = 10^{-32}$, the smallest positive machine number.

This accelerated fixed point method with algebraic multigrid procedure and a nice initialization procedure produces a valuable computational method for image restoration. Computational experiments demonstrate that the method is efficient and robust.

The paper is organized as follows. Section 2 is the finite difference scheme and the fixed point algorithm. In section 3, we introduce the algebraic multigrid algorithm to solve the corresponding linear systems. The combination of Krylov subspace acceleration and the fixed point iteration is given in section 4. And section 5 is the initialization procedure to produce a nice initial data for fixed point iteration. Finally, numerical results and discussion are given in section 6.

2 Difference Scheme and Fixed Point Method

Let us consider the Euler-Lagrange equation

$$-\alpha\nabla\cdot\left(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right)+K^*(Ku-z)=0 \text{ in } \Omega=(0,1)\times(0,1), \tag{2.1}$$

with zero Neumann (no flux) boundary condition. We partition the domain $(0,1)\times(0,1)$ into $L\times L$ uniform cells. Denote $1/L$ by h. The cell centers are $(x_l,y_k)=((l-1/2)h,(k-1/2)h)\,, l,k=1,\cdots,L$. The value $u(x_l,y_k)$ is approximated by $u_{l,k}$. Following [17], we discretize (1.2) by a standard five-point finite difference scheme:

$$\begin{aligned}&\tfrac{1}{h^2}\Big[\left(D_{l+1/2,k}+D_{l-1/2,k}+D_{l,k+1/2}+D_{l,k-1/2}\right)u_{l,k}\\&\quad -D_{l+1/2,k}u_{l+1,k}-D_{l-1/2,k}u_{l-1,k}-D_{l,k+1/2}u_{l,k+1}-D_{l,k-1/2}u_{l,k-1}\Big]\\&\quad +\left(K^*(KU-Z)\right)_{l,k}=0,\quad l,k=1,\cdots,L,\end{aligned} \tag{2.2}$$

where

$$D_{l+1/2,k}=\frac{\alpha}{\sqrt{|(u_{l+1,k}-u_{l,k})/h|^2+\beta}}, \tag{2.3}$$

and $U=(u_{1,1},u_{1,2},\cdots,u_{1,L},u_{2,1},\cdots,u_{2,L},\cdots,u_{L,L}), Z=(z_{1,1},z_{1,2},\cdots,z_{1,L},z_{2,1},\cdots,z_{L,L})$. The discrete Neumann boundary conditions is

$$u_{0,k}=u_{1,k},\quad u_{L+1,k}=u_{L,k},\quad u_{l,0}=u_{l,1},\quad u_{l,L+1}=u_{l,L}. \tag{2.4}$$

We abbreviate (2.2) by

$$A(U)U+K^*(KU-Z)=0. \tag{2.5}$$

Following [17], we use the following fixed point method to solve the above finite difference equation.

$$A(U^s)U^{s+1} + K^*(KU^{s+1} - Z) = 0. \tag{2.6}$$

In order to solve system (2.6) for u^{s+1} efficiently, Vogel [16] applied the geometric multigrid (GMG) method. Unfortunately, he found that the GMG converges slowly, because the diffusion coefficients $1/\sqrt{|\nabla u^s|^2 + \beta}$ varies too much. In this paper we use the algebraic multigrid (AMG) method instead, where the information of these large variation of coefficients are built in the interpolation operator and coarse grid equation. Near edges (where the coefficients $1/\sqrt{|\nabla u^s|^2 + \beta}$ is very small), it maintains more grid points than the GMG method in the coarse grid equation. Hence, the coarse grid equation is more accurate and the convergence rate is improved.

3 Basic AMG Algorithm

Now, we describe our version of the AMG algorithm [8, 15] briefly. We consider the following $n \times n$ system of linear equations

$$AU = F. \tag{3.1}$$

An AMG method breaks this equation into a sequence of smaller and smaller equations: $A^m U^m = F^m$, $m = 1, \cdots, M$, where $A^m = (a^m_{i,j})_{n_m \times n_m}$, $U^m = (u^m_1, u^m_2, \cdots, u^m_{n_m})^T$, and $F^m = (f^m_1, f^m_2, \cdots, {}^m_{n_m})^T$, with $n = n_1 > n_2 > \cdots > n_M$, $A^1 = A, U^1 = U, F^1 = F$. These equations formally play the same role as the coarse grid equations in the GMG method.

In a standard multigrid process, one needs to define the coarse grids, the interpolation operator I^m_{m+1}, the restriction operator I^{m+1}_m, and the coarse grid operator A^{m+1}. With these, at each level, a smoothing process, say Gauss-Seidel, is applied to the equation $A^m U^m = F^m$ to find an approximate solution $\widehat{U}^m$. The high frequency errors of the residual $r^m := F^m - A^m \widehat{U}^m$ are usually reduced in this smoothing process. The correction for low frequency errors is approximated by the following procedure. First, the correction equation $A^m e^m = r^m$ is restricted to the next coarser grid by the restriction operator. The resulting equation is solved to obtain the coarse grid correction e^{m+1}. This correction e^{m+1} is then interpolated back to level m by the interpolation operator to obtain approximate solution e^m.

We shall adopt Galerkin type algorithm, where $I^{m+1}_m = (I^m_{m+1})^T$ and $A^{m+1} = I^{m+1}_m A^m I^m_{m+1}$. Thus, we will only need to define the coarse grids and interpolation operators. We follow the approach in [8, 15] to define the grid Ω^m and its coarse grid C^m. The grid Ω^m is regarded as the indices $\{1, \cdots, n_m\}$ of the unknowns e^m_j, $1 \le j \le n_m$. The coarse grid C^m is a subset of Ω^m. The grid Ω^{m+1} is nothing but a re-indexing of C^m. We denote $\Omega^m - C^m$ by F^m, the fine grid. Criteria to determine C^m will be discussed later.

The interpolation operator I^m_{m+1} maps data on Ω^{m+1} to data on Ω^m. Namely, for $i \in C^m$, the datum e^m_i is taken to be the datum on the corresponding index on Ω^{m+1}; while for $i \in F^m$, e^m_i is interpolated from data on C^m. Roughly speaking, this interpolation formula is derived so that the i^{th} equation

$$a^m_{i,i} e^m_i + \sum_{j \in N^m_i} a^m_{i,j} e^m_j = r^m_i \approx 0 \tag{3.2}$$

is almost satisfied. Here, $N_i^m = \{j \in \Omega^m \mid a_{i,j}^m \neq 0,\ j \neq i,\}$, which can be thought as the neighbors of i.

In order to solve (3.2) approximately, we classify the neighbors of the point i into two classes. A point $j \in N_i^m$ is said to be strongly connected to i if

$$\mid a_{i,j}^m \mid \geq \theta \cdot \max_{k \neq i} \mid a_{i,k}^m \mid$$

for some fixed $0 < \theta \leq 1$, and weakly connected if otherwise. We denote the collection of these neighboring points by S_i^m (strong) and W_i^m (weak), respectively. We also denote $C^m \bigcap S_i^m$ by C_i^m. Our goal is to derive an interpolation formula

$$e_i^m = \sum_{j \in C_i^m} \omega_{i,j} e_j^m, \text{ for } i \in F^m$$

so that the i^{th} correction equation is almost satisfied:

$$a_{i,i}^m e_i^m + \sum_{j \in N_i^m} a_{i,j}^m e_j^m = 0. \tag{3.3}$$

Noting that $N_i^m = S_i^m \cup W_i^m = (S_i^m \cap C^m) \cup (S_i^m \cap F^m) \cup W_i^m$, the issue here is how to approximate e_j^m with $j \in S_i^m \cap F^m$ or $j \in W_i^m$ in terms of e_i or e_k^m with $k \in C_i^m$. Before going to the discussion of this issue, let us describe how to choose the coarse grid C^m for a moment.

The coarse grid C^m is chosen such that the following criteria are satisfied:

(C1) For each point $i \in F^m$, every point $j \in S_i^m$ is either in C_i^m or strongly connected to at least one point in C_i^m (i.e. $S_j^m \cap C_i^m \neq \phi$).

(C2) C^m should be the maximal subset of all points with the property that any two points in C^m are not strongly connected to each other.

Condition (C1) ensures that for $i \in F^m$, e_i^m can be constructed from the values e_k^m with $k \in C_i^m$ with certain accuracy. Condition (C2) means that C^m is chosen as smaller as possible to gain efficiency. In general, it is difficult to construct C^m to satisfy (C2) strictly. Ruge and Stüben [15] provided an $O(n_m)$ algorithm to construct the coarse grid C^m which is small enough and leads to linear computational complexity of the overall algorithm practically.

Let us go back to the issue: how to approximate e_j^m with $j \in S_i^m \cap F^m$ or $j \in W_i^m$ in terms of e_i^m or e_k^m with $k \in C_i^m$? For $j \in W_i^m$, we may simply approximate e_j^m by

$$e_j^m = e_i^m, \tag{3.4}$$

based on the smoothness of e^m which we do expect. For $j \in S_i^m \cap F^m$, we look into the j^{th} equation:

$$a_{j,j}^m e_j^m + \sum_{k \in C_i^m \cap N_j^m} a_{j,k}^m e_k^m + \cdots \approx 0.$$

The part "..." is secondary error and thus negligible. A natural approximation of e_j^m is the following average formula:

$$e_j^m = \sum_{k \in C_i^m \cap N_j^m} g_{j,k}^m e_k^m, \quad g_{j,k}^m = \frac{|a_{j,k}^m|}{\sum_{\ell \in C_i^m \cap N_j^m} |a_{j,\ell}^m|} \tag{3.5}$$

The condition (C1) (i.e. $C_i^m \cap S_j^m \neq \phi$) guarantees that $\sum_{k\in C_i^m} |a_{j,k}^m|$ is not too small.

The above interpolation formula was given in [15] when A^1 is an M-matrix. An improved interpolation formula using some geometric assumptions was proposed in [8]. These assumptions are as below.

(G1) Elements in N_i^m are the neighbors of a point i in Ω^m. Further, the larger the quantity $|a_{i,j}^m|$ is, the closer the point j is to the point i.

(G2) If $a_{i,j}^m < 0$ or $|a_{i,j}^m|$ is small, we say that the error between i and j is geometrically smooth. Otherwise, we call it geometrically oscillating. Here, we have normalized $a_{i,i} > 0$.

Roughly speaking,"geometrically," the average location of points in $C^m \cap S_i^m \cap S_j^m$ is somewhere between i and j. Therefore the error e_j^m can be approximated more accurately by an extrapolation formula using e_i and $\sum_{k\in C_i^m\cap S_j^m} g_{j,k}e_k^m$. More precisely, let us define

$$\zeta_{i,j}^m = \frac{-\sum_{k\in C_i^m\cap N_j^m} a_{j,k}^m}{\sum_{k\in C_i^m\cap N_j^m} | a_{j,k}^m |}, \qquad \eta_{i,j}^m = \frac{| a_{i,j}^m |}{\frac{1}{|C_i^m\cap N_j^m|}\sum_{k\in C_i^m\cap N_j^m} | a_{j,k}^m |}.$$

The quantity $\zeta_{i,j}^m$ indicates whether there is a large negative entry $a_{j,k}^m$ for $k \in C_i^m \cap N_j^m$. When $\zeta \geq 1/2$ and $a_{i,j}^m < 0$, it can be shown that the errors between the point i and the point j are geometrically smooth. The quantity $\eta_{i,j}^m$ roughly gives the "inverse ratio" of the distance between j and i to the average distance between the point j and the points in $C_i^m \cap N_j^m$. If $\eta_{i,j}^m < 3/4$, we think the "average location" of the points in $C_i^m \cap N_j$, denoted by $\bar{k}_{j,i}$, is closer to j than that of i. That is, $\bar{k}_{j,i}$ lies between i and j, and thus, an extrapolation formula for e_j in terms of e_i and $\sum_{k\in C_i^m\cap N_j^m} g_{j,k}e_k^m$ can be applied. When $\eta_{i,j}^m > 2$, we think i is closer to j than that of $\bar{k}_{j,i}$. In this case, we use an interpolation formula instead. Otherwise, we think $\bar{k}_{j,i}$ is very close to j and we should just use the average formula $\sum_{k\in C_i^m\cap N_j^m} g_{j,k}e_k^m$ to approximate e_j.

In summary, we use the following interpolation formulae.

(1) For $j \in S_i^m \cap F^m$, we have

$$e_j^m = \begin{cases} 2\sum_{k\in C_i^m} g_{j,k}^m e_k^m - e_i^m, & \text{if } \eta_{i,j}^m < 3/4, \zeta_{i,j} \geq 1/2 \text{ and } a_{i,j}^m < 0 \\ \frac{1}{2}(\sum_{k\in C_i^m} g_{j,k}^m e_k^m + e_i^m), & \text{if } \eta_{i,j}^m > 2, \zeta_{i,j}^m \geq 1/2 \text{ and } a_{i,j}^m < 0 \\ \sum_{k\in C_i^m} g_{j,k}^m e_k^m, & \text{otherwise.} \end{cases} \tag{3.6}$$

(2) For $j \in W_i^m$, we have

$$e_j^m = \begin{cases} e_i^m, & \text{if } C_i^m \cap S_j^m = \phi,\ a_{i,j}^m < 0 \\ -e_i^m, & \text{if } C_i^m \cap S_j^m = \phi,\ a_{i,j}^m > 0 \\ 2\sum_{k\in C_i^m} g_{j,k}^m e_k^m - e_i^m, & \text{if } C_i^m \cap S_j^m \neq \phi,\ \zeta_{i,j}^m \geq 1/2 \text{ and } a_{i,j}^m < 0 \\ \sum_{k\in C_i^m} g_{j,k}^m e_k^m, & \text{otherwise.} \end{cases} \tag{3.7}$$

The convergence proof for this improved AMG method was given in [8] when A^m is symmetric positive definite.

4 Krylov Subspace Acceleration

The Krylov subspace method [5, 13] is an acceleration technique for general iteration methods. Basically, it uses extrapolation to accelerate convergence rate. It is particularly suitable to accelerate our fixed point iteration, because the functional (1.1) is convex, or equivalently, the operator of the corresponding Euler-Lagrange equation (1.2) is monotone. We illustrate this acceleration procedure below. First, we choose two parameters K and s, with $K \leq s$. The Krylov subspace acceleration is perform after every s steps of fixed point iterations as the follows. For integer $n > 0$, let

$$U^{new} = U^{ns} + \sum_{k=1}^{K} \alpha_k (U^{ns+1-k} - U^{ns-k}). \tag{4.1}$$

where the coefficients α_k are chosen such that the residual R^{new} for U^{new} is minimum in L_2 norm, i.e.,

$$\min_{\alpha_1,\cdots,\alpha_K} (R^{new}, R^{new}). \tag{4.2}$$

We then reset U^{ns} to be U^{new}.

Noticing

$$R^{new} = R^{ns} + \sum_{k=1}^{K} \alpha_k (R^{ns+1-k} - R^{ns-k}), \tag{4.3}$$

the coefficients $(\alpha_1, \cdots, \alpha_K)$ can be found easily. For instance, for $K = 1$, α_1 is

$$\alpha_1 = \frac{-(R^{ns},\ R^{ns} - R^{ns-1})}{(R^{ns} - R^{ns-1},\ R^{ns} - R^{ns-1})} \tag{4.4}$$

Remark. The acceleration formula with $K = 1$ is the same as the one given by Brandt and Mikulinsky in [5] when the iteration is linear. Indeed,

$$\alpha_1 = \frac{-\lambda}{\lambda - 1} \text{ with } \lambda = \frac{(R^{ns},\ R^{ns} - R^{ns-1})}{(R^{ns-1},\ R^{ns} - R^{ns-1})}.$$

And λ here is approximately the largest eigenvalue of the iteration matrix M. The reason is shown below.

Suppose the eigenvalues of M are $\lambda_1, \lambda_2, \cdots, \lambda_m = \lambda$ with the corresponding eigenvectors $v_1, v_2, \cdots, v_m$. Then

$$R^0 = r_1^0 v_1 + r_2^0 v_2 + ... + r_m^0 v_m,$$

$$R^n s = \lambda_1^{ns} r_1^0 v_1 + \lambda_2^{ns} r_2^0 v_2 + ... + \lambda_m^{ns} r_m^0 v_m \approx \lambda_m^{ns} r_m^0 v_m \equiv \lambda^{ns} r_n^0 v_m.$$

$$\frac{(R^{ns},\ R^{ns} - R^{ns-1})}{(R^{ns-1},\ R^{ns} - R^{ns-1})} \approx \frac{(\lambda^{ns} r_m^0 v_m,\quad \lambda^{ns-1}(\lambda - 1) r_m^0 v_m)}{(\lambda^{ns-1} r_m^0 v_m,\quad \lambda^{ns-1}(\lambda - 1) r_m^0 v_m)} = \lambda.$$

5 Initialization for the Nonlinear Iteration

The initial guess u^0 in the above fixed point iteration (2.6) is very important for convergence. We shall use the idea of GMG method to construct our initial guess. First, we start at the coarsest grid (4×4 grid) by performing the fixed point iteration (2.6) few times until the

residual is reduced to 10^{-2} of the original one. Here, the noisy data z is restricted to this coarsest grid through a usual restriction operator (5.1):

$$u^c_{l,k} = \frac{1}{4}(u^f_{2l-1,2k-1} + u^f_{2l-1,2k} + u^f_{2l,2k-1} + u^f_{2l,2k}). \tag{5.1}$$

Next, we interpolate the computed result to the next level grid (8×8) by the following interpolation formula: At (x_{2l-1}, y_{2k-1}), we take the same values from the coarse grid. At (x_{2l}, y_{2k}), we use

$$u^f_{2l,2k} = \frac{1}{16}(9u^c_{l,k} + 3u^c_{l+1,k} + 3u^c_{l,k+1} + u^c_{l+1,k+1}). \tag{5.2}$$

For the rest grid points, we use bilinear interpolation. This interpolated U is used as the initial guess for the fixed point iteration at this grid level (8×8). We continue this process until we reach the finest grid (256×256). At every level, we use our AMG method and the Krylov acceleration to solve the systsm (2.6).

6 Numerical Experiments and Discussions

In the numerical experiments below, we take the blur operator K to be the identity matrix (i.e. $K = I$). Two benchmark model problems are considered here [7, 3]. The original images (denoted by u^e, see Figure 1) have 256×256 pixels. Each pixel has value in $[0, 255]$. A Gaussian distribution with mean 0 and variance σ (to be determined later) is added to the original images and the resulting noisy images (denoted by z) are displayed in Figure 2, respectively. By definition, we have $\|u^e - z\|_{L_2} = \sigma$. We choose σ such that the noise-to-signal ratios of these two noisy images are

$$\frac{\|u^e - z\|_{L_2}}{\|u^e\|_{L_2}} = \begin{cases} 0.21 & \text{for model I} \\ 0.28 & \text{for model II.} \end{cases} \tag{6.1}$$

The value of σ is approximate 35.

In the restoration process, we take the parameters $\alpha = 1.18$, which was chosen by T. Chan et al. [7], and $\beta = 0.01$, 0.001 and 10^{-32}. In the AMG procedure, we apply the simple V-cycle and use the Gauss-Seidel iteration as the smoother.

6.1 Normalized residual

An important issue in image restorations is to choose a quantity to measure the quality of improvement. It is used as a stopping criterion for the fixed point iteration. Usually, the residual of the system (2.6) is chosen. But, a normalization is needed. Namely, we should use $D^{-1}(Re)$ as the normalized residual. Here, (Re) is the residual of the system (2.6) and D is the corresponding diagonal matrix. The reason for this normalization is the follows. Due to the fact that the diffusion coefficient is very large in smooth region, we observe that the unnormalized residual is very large in those components where u is smooth (thus, no more denoising is needed), and is relatively small in those where u is less smooth (thus, either it has a jump or it needs further denoising). A normalization will cure this imbalance. Numerical experiments below demonstrates that this quantity is able to measure the improvement of the denoising process. From now on, we shall denote this normalized residual by Re.

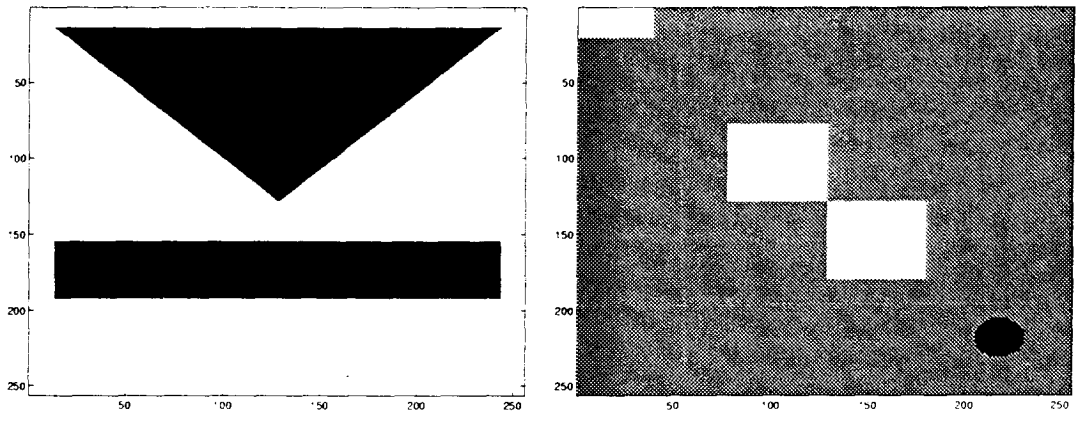


Figure 1: Original images of Model I (left) and Model II (right)

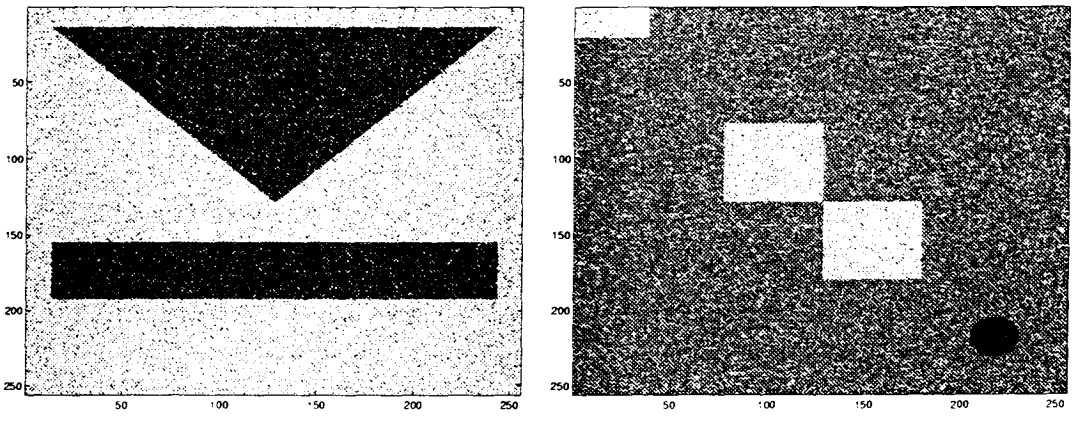


Figure 2: Noisy images of Model I (left) and Model II (right)

6.2 Fixed point iteration with AMG method

In the first set of numerical experiments, we apply the fixed point method to the model images at the finest grid directly. No particular preparation of initial data and nor the Krylov acceleration are adopted at this moment. In each fixed point iteration, only one V-cycle of the AMG method is applied for solving the corresponding linear system. There is no need to have more iteration because the dominant error is from the outer iteration. The stopping criterion for the fixed point iteration in this paper is a relative decrease of the residual by a factor of 10^{-4} for model I and of 10^{-5} for model II, namely,

$$\frac{\|Re^N\|_{L_2}}{\|Re^0\|_{L_2}} \leq \begin{cases} 10^{-4}, & \text{for Model I,} \\ 10^{-5}, & \text{for Model II.} \end{cases}$$

Table 1 is the number of fixed point iterations (denoted by N) and the "CPU" time (on Sparc 10 machine) needed. Table 2 shows the convergence factors ρ_F and ρ_A of the fixed

point iteration and of the AMG, respectively. They are defined by the ratio of the residual after and before the iteration. We observe that the AMG reduces the residual very fast, whereas the fixed point iteration becomes inefficient as N becomes large. Comparing with the result (only for Model I) of the primal-dual Newton method [7], we observe that the number of outer iteration is about the same in both methods, whereas the number of inner iteration is just one in our method versus 4 to 5 in their method. The reason is that the linear equation in fixed point method is much easier to solve than that of Newton's method.

Model	β	N	CPU time (in second)
I	0.01	22	79.60
	0.001	22	84.11
II	0.01	48	165.97
	0.001	49	172.6

Table 1: Number of fixed point iterations N and the CPU time (on Sprac 10) needed to achieve a reduction of residual by a factor of 10^{-4} for Model I and of 10^{-5} for Model II.

N	1	2	3	4	5	6	7	8
ρ_P	0.665	0.007	0.456	0.704	0.767	0.799	0.817	0.830
ρ_A	0.209	0.009	0.088	0.069	0.064	0.063	0.058	0.052
N	9	10	11	12	13	14	15	16
ρ_P	0.842	0.850	0.851	0.860	0.867	0.872	0.876	0.881
ρ_A	0.049	0.048	0.045	0.044	0.043	0.043	0.043	0.047
N	17	18	19	20	21	22		
ρ_P	0.886	0.889	0.894	0.895	0.901	0.902		
ρ_A	0.051	0.049	0.052	0.053	0.054	0.053		

Table 2: ρ_F and ρ_A: convergence factors of the fixed point iteration and the AMG method. This result is for the Model I with $\beta = 0.01$. Notice that ρ_A is very small. Thus the number of inner iteration is just one at every outer iteration.

6.3 Improvement by the Krylov Acceleration

The slow convergence of the fixed point iteration above can further be improved by the Krylov acceleration method. In the application of Krylov acceleration, we choose the parameter $s = 4$, i.e. we apply the Krylov acceleration every four fixed point iterations. The parameter K is taken to be 1 or 2. The result is given in Table 3. The total number of iterations is reduced to about 50%. The overhead is low, because only simple algebraic operations are needed. The results demonstrate that the Krylov acceleration method is very efficient to accelerate the convergence of our fixed point method.

In numerical experiment, we observed that, in the fixed point iteration without Krylov acceleration, the convergence factor tends to 1 as N increases. As a result, the residual stays

K	Model	β	N	CPU time (in second)
1	I	0.01	12	44.14
		0.001	12	44.34
	II	0.01	21	74.48
		0.001	21	75.53
2	I	0.01	12	44.64
		0.001	12	44.82
	II	0.01	21	74.72
		0.001	20	71.26

Table 3: Number of fixed point iteration N needed with Krylov acceleration. Comparing with Table 1, the improvement is almost double.

almost constant. However, with the Krylov acceleration, the residual can further be reduced, as shown in Table 4, where Re_N denotes the residual after N iterations. After 30 iteration, the convergence factors are more than 0.94 and the residuals are reduced slowly. The Krylov acceleration is no longer efficient.

Method	Model	Re_0	Re_{20}	Re_{30}
Fixed point	I	30.54	3.43×10^{-3}	1.36×10^{-3}
	II	29.03	2.31×10^{-3}	9.39×10^{-4}
Fixed point + Krylov acceleration	I	30.54	7.10×10^{-4}	2.93×10^{-4}
	II	29.03	3.05×10^{-4}	8.00×10^{-5}

Table 4: Residuals of the fixed point method with and without the Krylov acceleration.

6.4 Further improvement by a good preparation of initial data

In previous numerical experiments, the initial data is 0. We can have less nonlinear iterations if we start from a good initial guess. As mentioned in previous section, such an initial guess is obtained by interpolation from an approximate solution at coarse grid successively. We start from 4×4 grid. The stopping criterion at each grid level is the 10% reduction of the residual. Table 5 is the number of iteration and CPU time with this initialization. There, C_i, C_f and C_{total} are the CPU times of the initialization, the fixed point iteration and the total time, respectively. We observe that the initialization is about 10% of the total work for Model I and 7% for Model II. Comparing Table 3 and Table 5, we observe this initialization gives about 20% improvement. What a surprise is that we are allowed to choose a very small regularization coefficient β, say $\beta = 10^{-32}$, the smallest positive machine number. The overall method still converges. Without a good initialization, the fixed point method is hard to converge for $\beta < 10^{-4}$. Figure 3 is the denoised images with $\beta = 10^{-32}$.

In conclusion, numerical experiments demonstrate that our algorithm, which combines the fixed point method, the AMG method, the Krylov acceleration and good initialization, is efficient and robust.

Model	β	N	C_i	C_f	C_{total}
I	0.01	9	3.16	32.46	35.62
	0.001	9	3.18	32.68	35.86
	1.0×10^{-32}	9	3.20	33.11	36.31
II	0.01	17	4.51	56.87	61.38
	0.001	17	4.48	57.75	62.23
	1.0×10^{-32}	17	4.59	59.15	63.74

Table 5: CPU time by using the combination of the fixed point method, AMG algorithm, Krylov acceleration and good preparation of initial guess. Here, C_i is the time for nitialization, C_f, the time for fixed point iteration, and $C_{total} = C_i + C_f$ is the total amount of time. The improvement of the good initialization is about 40%. It can also allow us to choose a very small β, the smallest positive machine number.

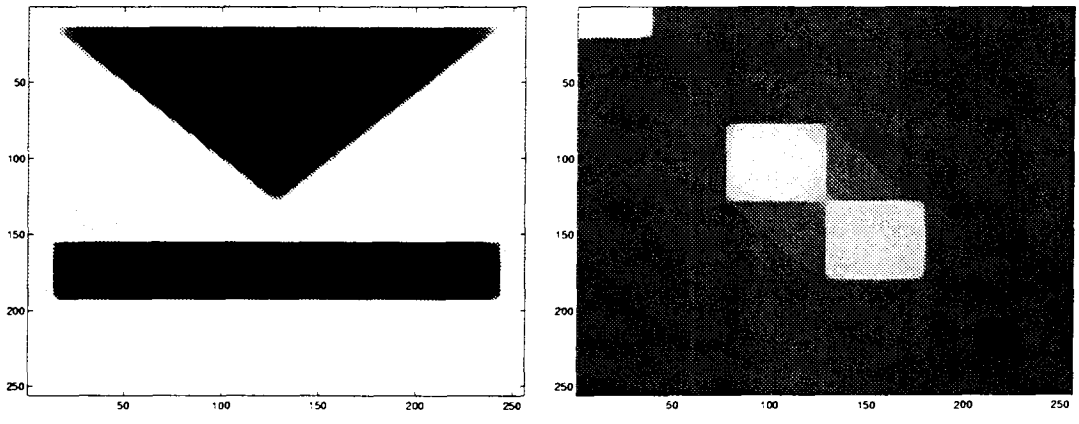


Figure 3: Restored images of Model I (left) and Model II (right)

References

[1] R. Acar and C. R. Vogel, *Analysis of total variation penalty methods for ill-psoed problems,* Inverse Problems, 10(1994), pp 1217-1229.

[2] L. Alvarez, P. -L. Lions, and J. -M. Morel, *Image selective smoothing and edge detection by nonlinear diffusion II,* SIAM J. Numer. Anal., 29(1992), pp 845-866.

[3] C. A. Z. Barcelos, and Y. Chen, *Heat flow and related minimization problem in image restoration,* Computers and Mathematics with Applications, 39(2000), pp 81-97.

[4] G. Barles, and P. E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations,* Asymptotic Analysis 4(1991), pp 271-283.

[5] A. Brandt, and V. Mikulinsky, *On recombining iterants in multigrid algorithms and problems with small islands,* SIAM Sci. Comput., 16(1995), pp 20-28.

[6] R. Chan, T. Chan, and H. Zhou, *Advanced signal processing algorithms,* In Proceedings of the International Society of Photo-Optical Instrumentation Engineers, F. Luk, ed., SPIE, 1995, pp 314-325.

[7] T. F. Chan, G. H. Golub, and P. Mulet, *A Nonlinear primal-dual method for total variation-based image restoration,* SIAM J. Sci. Compt., 20(1999), pp 1964-1977.

[8] Q. Chang, Yaushu Wong and H. Fu, J. Comput. Phys., 125(1996), pp.279-292.

[9] Q. Chang, S. Ma and G. Lei, J. of Computer Math., 70(1999), pp. 539-552.

[10] Y. Li and F. Santosa, *An affine scaling algorithm for minimizing total variation in image enhancement,* Tech. Report 12/94, Center for Theory and Simulation in Science and Engineering, Cornell University, 1994.

[11] S. McCormick, *Multigrid Methods,* SIAM, Philadelphia, 1987.

[12] M. E. Oman, *Fast multigrid techniques in total variation-based image reconstruction,* to appear in Preliminary Proceedings of the 1995 Copper Mountain Conference on Multigrid Methods.

[13] C. W. Oosterlee, and T. Washio, *Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows,* SIAM J. Sci. Comput., 21(2000), pp 1670-1690.

[14] L. Rudin, S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms,* Phys. D, 60(1992), pp 259-268.

[15] J. Ruge and K. Stüben, *Algebraic Multigrid,* in Multigrid Methods, (S. F. McCormick, ed.) 4, SIAM, Philadephia, (1987) pp. 73-130.

[16] C. R. Vogel, *A multigrid method for total variation-based image denoising,* in Computation and Control IV, Progr. Systems Control Theory 20, Birkhauser, Boston, MA, 1995, pp 323-331.

[17] C. R. Vogel, and M. E. Oman, *Iterative methods for total variation denoising,* SIAM. J. Sci. Compt., 17(1996), pp 227-238.

[18] Y. Zhou, *Applications of discrete functional analysis to the finite difference method,* International Academic Publishers, Beijing, 1991.