ADAPTIVE WAVELET COLLOCATION METHODS FOR IMAGE SEGMENTATION USING TV-ALLEN-CAHN TYPE MODELS

ZHIJIAN RONG[1], LI-LIAN WANG[2] AND XUE-CHENG TAI[3]

Abstract. An adaptive wavelet-based method is proposed for solving TV(total variation)-Allen-Cahn type models for multi-phase image segmentation. The adaptive algorithm integrates (i) grid adaptation based on a threshold of the sparse wavelet representation of the locally-structured solution; and (ii) effective finite difference on irregular stencils. The compactly supported interpolating-type wavelets enjoys very fast wavelet transforms, and act as a piecewise constant function filter. These lead to fairly sparse computational grids, and very efficient algorithms for multi-phase image segmentation using piecewise constant level set methods (PCLSM). The proposed method is also applied to image restoration and similar advantages are observed.

1. Introduction

The solutions of many evolution partial differential equations (PDEs) dynamically exhibit localized structures and sharp transitions such as spikes, shock waves and singular layers. A successful implementation of adaptive strategies for such PDEs can significantly reduce the computational cost and increase the accuracy of numerical approximations. Various adaptive techniques based on domain and/or solution adaptivity have been developed, which include moving mesh methods (see, e.g., [10, 27, 33]), mesh refinement methods (see, e.g., [5, 6, 10, 19, 42]), and wavelet-based methods. Typically, a wavelet-based method takes the advantage of the compression capability of wavelets [18] and dynamically evolves the sparse representation of the locally-structured solution through certain thresholding techniques. The existing approaches can be roughly classified into wavelet-Galerkin (see, e.g., [2, 7, 26, 57]) and wavelet-collocation (see, e.g., [8, 9, 24, 25, 53, 54]) methods. The former approach solves the problems in frequency space, so one needs to transform back and forth between the physical and wavelet domains. The wavelet-collocation method is implemented in physical space on a dynamically adaptive grid, so it is well-suited for solving nonlinear problems and treating general boundary conditions. In contrast with many approaches based on purely manipulating the wavelet bases, another more effective method is to merely use wavelets as a tool to analyze and update the solution representation, and to perform numerical differentiations by finite difference (see, e.g., [24, 25, 28, 29, 53]). Our approach falls into this category, and the main purpose is to develop efficient adaptive algorithms for solving nonlinear PDEs arisen from image segmentation and restoration.

Image segmentation is a fundamental task in image processing and computer vision, and it is also a core component of many applications such as medical imaging. Various variational and PDE-based

Date: October 12, 2009.

1991 *Mathematics Subject Classification.* 65N35, 65N22, 65F05, 35J05.

Key words and phrases. Interpolating wavelets, adaptivity, total variation, image segmentation and restoration, piecewise constant level-set methods.

This research is supported by Singapore MOE Grant T207B2202 and Singapore NRF2007IDM-IDM002-010.

[1] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 637371, Singapore, and School of Mathematical Sciences, Xiamen University, 361005, China.

[2] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 637371, Singapore.

[3] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, 637371, Singapore, and Department of Mathematics, University of Bergen, Bergen, Norway.

models have been proposed for such a task, that is, to extract important objects from an image [30,41]. The level set method [44] has become a very useful tool for the formulation and solution of segmentation problems [11, 55]. The main idea is to represent and evolve a contour or a surface implicitly by imbedding them into a higher dimensional level set function, which allows for automatic change of topology, such as merging and breaking, and use of fixed regular grids for computations. Recently, a piecewise constant level-set method (PCLSM) was proposed in [35, 36] for image segmentation and interface problems. The key point is to represent and identify an evolving curve or surface through discontinuities of a piecewise constant function. The use of PCLSM allows for representing arbitrary number of phases by one single level set function, and avoiding the reinitialization required in classical level set methods. The application of PCLSM for Mumford and Shah image segmentation [41] leads to models closely related to phase-field models (see e.g., [1, 32, 36]). Compared with the Allen-Cahn model [1]), the Laplacian operator is replaced by the nonlinear total-variation (TV) operator in [32,36]). Due to this nonlinear nature of the model, it will add some difficulties to solve the corresponding equtions. However, this new model will avoid the difficulties with mesh refinement around the singular layers [23, 47]. Moreover, both models can approximate the TV-norm of the function. To distinguish it from the Allen-Cahn model, we term this new models as TV-Allen-Cahn model in case of two-phase problems.

Some algorithms have been developed and tested for the TV-Allen-Cahn model [3, 4, 15, 16, 32, 35, 36, 39]. In [35, 36], the traditional Augmented Lagrangian method have been used. The numerical convergence is normally slow. In [16, 32, 39], the AOS scheme of [37, 38, 56] was used to get faster iterative solvers for the nonlinear TV operator. In addition, multigrid method was tested in [15] and graph-cut methods were proposed in [3, 4]. For real applications, there is a great demand for even faster and stable algorithms. It is important to notice that the TV-Allen-Cahn models evolve piecewise constant level set functions. Hence, such locally-structured solutions will be well-suited for exploiting the sparsity and compression capability of the wavelet methods so as to design efficient adaptive wavelet collocation methods for these underlying highly nonlinear stiff PDEs. Motivated by several earlier works, the adaptive algorithm to be developed consists of two main components: (i) grid adaptation using a threshold of the locally compact interpolating-type wavelet decomposition of the piecewise constant dominant solutions, and (ii) a fast finite difference solver on irregular stencils. A key to efficiency is to associate every wavelet uniquely with a collocation point, so the threshold of wavelet representations adapts the computational grids automatically. Moreover, the wavelets act as a piecewise constant function pass filter, which favors functions with piecewise constant structures such as edges, frames and boundaries. These lead to very sparse solution representations and nearly-optimal computational grids. In this context, the wavelets shall be used to select the significant grids, although the grid refinement can be easily done. Indeed, this can relax the stiffness of the models. Similar approaches have been discussed and applied in applications such as fluid dynamics. To use this for image processing and nonlinear degenerate diffusion operators, some novel approach are needed. It is well known that wavelets have many remarkable applications in signal and image processing such as compression and denoising, and there has been a large body of literature on this topic. Interestingly, some recent work (see, e.g., [14]) has shown that the algorithm combines the denoising property of wavelet methods with that of TV model, produces results superior to each method when implemented alone. Although different types of wavelets were used in e.g., [14], in this aspect, our algorithm also inherits such a merit for simultaneous image segmentation and denoising.

The rest of the paper is organized as follows. In section 2, we briefly review the PCLSM for Mumford and Shah image segmentation, and present the TV-Allen-Cahn models. In section 3, we introduce the multi-resolution wavelet analysis and interpolating-type wavelet transforms based on the subdivision scheme and lifting techniques. We describe the adaptive wavelet collocation algorithms in section 4, and the final section is for various numerical experiments on multi-phase image segmentation and restoration.

2. Image segmentation using PSLSM

Image segmentation is one of the fundamental tasks in image processing and computer vision, and it aims to extract objects of interest by partitioning a given image into a finite number of important regions. Consider for example the partition of an open bounded domain $\Omega \in \mathbb{R}^2$ into two disjunctive subdomains by a closed curve Γ. It is known that the classical level-set approach is to embed Γ into a two-dimensional surface $z = \phi(x, y)$, and identify Γ as the zero level set: $\Gamma = \{(x, y) \in \Omega : \phi(x, y) = 0\}$. A typical choice of ϕ is the signed distance function:

$$\phi(x,y) = \begin{cases} \mathrm{dist}(\boldsymbol{x}, \Gamma), & \boldsymbol{x} = (x,y) \in \mathrm{inside}(\Gamma), \\ -\mathrm{dist}(\boldsymbol{x}, \Gamma), & \boldsymbol{x} = (x,y) \in \mathrm{outside}(\Gamma). \end{cases} \tag{2.1}$$

As a variance, the binary level set method in [35] identifies the contour Γ as the discontinuity of the piecewise constant level set function:

$$\phi(x,y) = \begin{cases} 1, & (x,y) \in \mathrm{inside}(\Gamma), \\ -1, & (x,y) \in \mathrm{outside}(\Gamma). \end{cases} \tag{2.2}$$

We plot in Figure 2.1 an illustration of two level set functions.

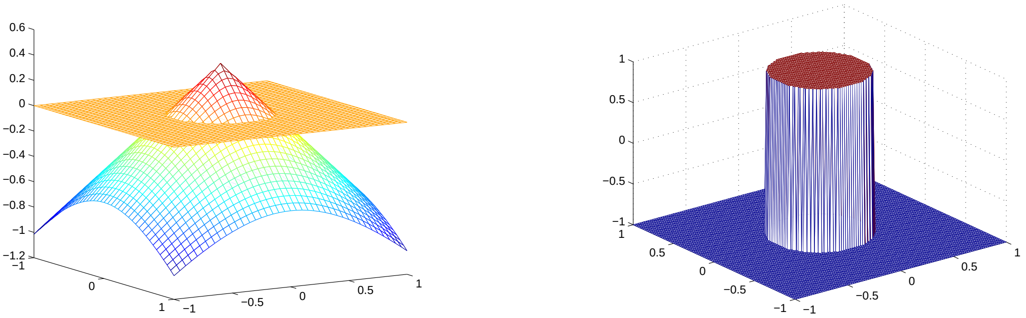


Figure 2.1. Signed distance function(left) versus piecewise constant function(right)

The original Mumford and Shah model [41] for image segmentation can be formulated as: given an image u_0, find a partition Ω_i of Ω, and an optimal piecewise smooth approximation u of u_0 such that u varies smoothly within each region Ω_i, and rapidly and discontinuously across the boundaries of Ω_i. More precisely, this task can be carried out by solving the minimization problem:

$$\inf_{u,\Gamma} \Big\{ F^{MS}(u,\Gamma) = \int_\Omega |u - u_0|^2 d\boldsymbol{x} + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 d\boldsymbol{x} + \nu |\Gamma| \Big\}, \tag{2.3}$$

where $|\Gamma|$ is the length of Γ, and $\mu, \nu > 0$ are fixed parameters to weight the terms in the cost functional.

Assuming that u_0 consists of two phases of approximately piecewise constant intensities c_1 and c_2, we seek the approximation u of u_0 :

$$u = c_1 \frac{1+\phi}{2} + c_2 \frac{1-\phi}{2} \approx u_0, \tag{2.4}$$

characterized by the piecewise constant level set function (2.2). Accordingly, the Mumford and Shah model (2.3) is reduced to

$$\inf_{c_1, c_2,\, \phi} \Big\{ F(c_1, c_2, \phi) = \int\limits_\Omega |u - u_0|^2 d\boldsymbol{x} + \beta \int\limits_\Omega |\nabla \phi| d\boldsymbol{x} \Big\}, \quad \beta > 0, \tag{2.5}$$

subject to the constraint: $\phi^2 = 1$. In [36], the constraint was treated by the augmented Lagrangian technique. Here, we merely use the penalization method, so the corresponding model becomes

$$\inf_{c_1, c_2,\, \phi} \Big\{ L(c_1, c_2, \phi) = \int\limits_\Omega |u - u_0|^2 d\boldsymbol{x} + \beta \int\limits_\Omega |\nabla \phi| d\boldsymbol{x} + \frac{\gamma}{4} \int_\Omega (\phi^2 - 1)^2 d\boldsymbol{x} \Big\}. \quad \beta, \gamma > 0, \tag{2.6}$$

The associated Euler-Lagrange equation in gradient descent formulation reads

$$\frac{\partial \phi}{\partial t} = \beta \mathrm{div}\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - (u - u_0) \frac{\partial u}{\partial \phi} + \gamma \phi (\phi^2 - 1), \quad \text{in } \Omega. \tag{2.7}$$

It is closely related to the phase field models [1, 32]. To distinguish it from the original Allen-Cahn model, we term this binary model as the TV-Allen-Cahn equation. In contrast to the traditional diffusive phase-field models, the equation (2.7) contains a nonlinear TV-diffusion operator rather than a linear Laplacian diffusion. This induces some difficulties for numerical solutions, but since the TV-term has the capability to preserve discontinuities and edges, the TV-model might avoid mesh refinement as usually required for phase-field models [23, 47].

This model can also be used for multi-phase segmentation [35,36]. In classical level set methods, the sign of n level set functions are used to identity up to 2^n phases [55]. Here, we use one single piecewise constant function to locate arbitrary number of phases. Basically, in order to decompose the image domain Ω into n separated regions $\{\Omega_i\}_{i=1}^n$, we label each one by an integer and define the piecewise constant level set function as

$$\phi(x, y) = i, \quad \text{if } \ (x, y) \in \Omega_i, \quad i = 1, 2, \cdots, n. \tag{2.8}$$

The characteristic function of Ω_i is given by the Lagrange basis polynomial:

$$\chi_i = \frac{1}{\alpha_i} \prod_{\substack{j=1 \\ j \neq i}}^n (\phi - j) \ \text{ and } \ \alpha_i = \prod_{\substack{k=1 \\ k \neq i}}^n (i - k). \tag{2.9}$$

Like (2.4), we seek an n-phase piecewise constant function u to approximate the original image u_0,

$$u = \sum_{i=1}^n c_i \chi_i \approx u_0, \tag{2.10}$$

where the unknown constants $\{c_i\}$ are the approximations of intensities of u_0 in Ω_i. Similarly, the constraint $\phi^2 = 1$ in (2.5) is replaced by

$$K(\phi) \equiv (\phi - 1)(\phi - 2) \cdots (\phi - n) \equiv \prod_{i=1}^n (\phi - i) = 0. \tag{2.11}$$

This guarantees there is no vacuum and overlap between different phases, that is, each point $x \in \Omega$ can take one and only one phase value. With the above setup, we reduce the Mumford and Shah model (2.3) to

$$\min_{\substack{\mathbf{c},\, \phi \\ K(\phi)=0}} \Big\{ F(\mathbf{c}, \phi) = \int\limits_\Omega |u - u_0|^2 d\boldsymbol{x} + \beta \int\limits_\Omega |\nabla \phi| d\boldsymbol{x} \Big\}, \quad \beta > 0, \tag{2.12}$$

where vector $\mathbf{c} = (c_1, c_2, \cdots, c_n)$. The penalization method and the gradient descent technique give us the flow (cf. [52]):

$$\frac{\partial \phi}{\partial t} = \beta \text{div}\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - (u - u_0)\frac{\partial u}{\partial \phi} - \frac{\gamma}{2} K(\phi)K'(\phi), \quad \text{in } \Omega, \tag{2.13}$$

which we call the multi-phase TV-Allen-Cahn model. For a given ϕ, the phase values $\{c_i\}$ are determined by solving the $n \times n$ linear system:

$$\sum_{j=1}^{n} c_i \int_{\Omega} \chi_i \chi_j \, d\boldsymbol{x} = \int_{\Omega} u_0 \chi_i \, d\boldsymbol{x}, \quad \text{for } i = 1, 2, \cdots n, \tag{2.14}$$

and u is updated by using (2.10).

Several numerical methods have been proposed to solve this nonlinear stiff model (2.13)-(2.14), see [3,4,15,16,32,39]. Here, we adopt a splitting scheme (or operator splitting, cf. [40,48]) simialr to the ones of [16,32,39], which allows to solve the following two problems consecutively and recursively at each time step:

$$\phi_t = \beta \nabla \cdot \Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - (u - u_0)\frac{\partial u}{\partial \phi}, \tag{2.15}$$

and

$$\phi_t = -\frac{\gamma}{2} K(\phi)K'(\phi). \tag{2.16}$$

The latter is an ordinary differential equation (2.16), which can be solved efficiently (analytically for two-phase case), so we focus on the adaptive algorithm for solving (2.15). In view of this, the proposed method can easily be applied to the ROF model [45] for image denoising:

$$\phi_t = \beta \nabla \cdot \Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) - (\phi - u_0), \tag{2.17}$$

where u_0 is a given noisy image, and ϕ at the steady state gives the restored image.

More generally, we consider the curvature equation [43]:

$$\begin{aligned} &\frac{\partial \phi}{\partial t} = \beta \text{div}\Big(\frac{\nabla \phi}{|\nabla \phi|}\Big) + F(\phi), \quad \text{in } \Omega, \\ &\frac{\partial \phi}{\partial \boldsymbol{n}} = 0, \ \text{ on } \partial\Omega; \quad \phi(\boldsymbol{x}, 0) = \phi_0, \quad \text{in } \Omega. \end{aligned} \tag{2.18}$$

where $\boldsymbol{n}$ is the unit outer normal vector of $\partial\Omega$ and $F(\phi)$ is a functional of ϕ. We next introduce multi-resolution analysis based on interpolating-type wavelet transforms, and describe the adaptive algorithms for (2.18).

3. Interpolating Wavelets, Transforms and Differentiation

In this section, we briefly review the subdivision scheme by Deslauriers and Dubuc [20,22] and introduce the multi-resolution framework. A standard wavelet multi-resolution analysis is induced by the scaling functions $\{\varphi_j^k\}$, and the wavelets $\{\psi_j^k\}$, which are usually generated from a single scaling and wavelet function through transition and dilation [18]. However, for adaptive solutions of PDEs, we use the notion of the subdivision scheme [20,22] (also called the cascade algorithm [50]) to derive interpolating wavelets on dyadic grids (see, e.g., [21,25]), which leads to more effective algorithms.

3.1. **Dyadic grids and multi-resolution analysis.** Let $I = (0,1)$, and preassign a set of dyadic grids on $\bar{I}$:

$$\mathcal{G}^j = \left\{ x_k^j = \frac{k}{2^j} \ : \ 0 \le k \le 2^j \right\}, \qquad j \ge 0, \tag{3.1}$$

where the superscript j represents the level of resolution, and the subscript k indicates the spatial location. Denote

$$\mathcal{F}^j := \mathcal{G}^{j+1} \setminus \mathcal{G}^j = \left\{ y_k^j = \frac{2k+1}{2^{j+1}} \ : \ 0 \le k \le 2^j - 1 \right\}, \qquad j \ge 0. \tag{3.2}$$

Note that we have used x_k^j and y_k^j to represent the grids in $\mathcal{G}^j$ and $\mathcal{F}^j$, respectively. It is obvious that

$$x_k^j = x_{2k}^{j+1} \in \mathcal{G}^j, \quad y_k^j = x_{2k+1}^{j+1} \in \mathcal{F}^j, \quad 0 \le k \le 2^j, \tag{3.3}$$

and $\{\mathcal{G}^j\}_{j\ge 0}$ forms a sequence of nested grid sets, i.e., $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. An illustration of one dimensional and two-dimensional tensorial dyadic grids is given in Figure 3.1.

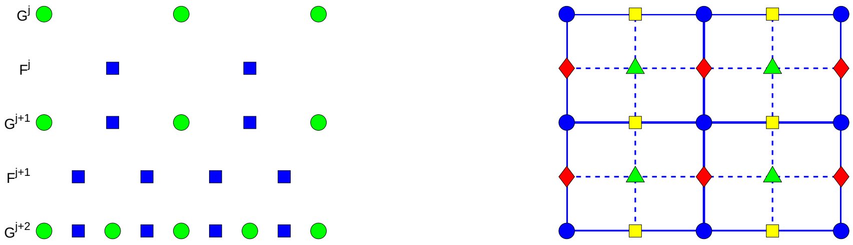


FIGURE 3.1. Illustration of dyadic grids.

Now, we are ready to introduce the multi-resolution analysis. Let $\{\varphi_k^j\}$ (the scaling basis) and $\{\psi_k^j\}$ (the wavelet basis) be the sets of basis functions (to be specified later) associated with $\mathcal{G}^j$ and $\mathcal{F}^j$, respectively. We define

$$\mathbf{V}^j = \mathrm{span}\{\varphi_k^j : k \in \mathcal{G}^j\}, \quad \mathbf{W}^j = \mathrm{span}\{\psi_k^j : k \in \mathcal{F}^j\}, \quad j \ge 0. \tag{3.4}$$

Here, the notation $k \in \mathcal{G}^j$ means the grid point $x_k^j \in \mathcal{G}^j$ and likewise for other similar expressions. Formally, by construction, we have

$$\mathbf{V}_0 \subset \mathbf{V}^1 \subset \cdots \subset \mathbf{V}^j \subset \cdots; \quad \mathbf{V}^{j+1} = \mathbf{V}^j + \mathbf{W}^j. \tag{3.5}$$

The multi-resolution decomposition from level J down to a coarser level J_0 is performed as

$$\mathbf{V}^J = \mathbf{V}^{J_0} + \mathbf{W}^{J_0} + \mathbf{W}^{J_0+1} + \cdots + \mathbf{W}^{J-1}, \quad J_0 \ge 0, \ \ J \ge 1, \tag{3.6}$$

so equivalently we have the wavelet representation:

$$f(x) \sim \sum_{k\in\mathcal{G}^{J_0}} s_k^{J_0} \varphi_k^{J_0}(x) + \sum_{J_0 \le j \le J-1} \sum_{k \in \mathcal{F}^j} d_k^j \psi_k^j(x), \quad \forall x \in \bar{I}, \tag{3.7}$$

where the wavelet coefficients $\{d_k^j\}$ encode the solution structures on successive multi-scales. The reconstruction process is carried out reversely.

The significance of wavelets is that most wavelet coefficients of functions with isolated small scales on a large-scale background are very small. Therefore, a large number of wavelets can be discarded through thresholding. Furthermore, we shall use interpolating-type wavelets, where each physical point uniquely associates with one wavelet, so the computational grids can be adapted via analyzing the wavelet coefficients. At this point, two important issues need to be addressed: (i) the construction of wavelets with high compression capabilities; (ii) the efficiency of wavelet transforms and grid adaption.

In this paper, we focus on some interpolating-type wavelets with an aim to achieve a good trade-off between (i) and (ii).

3.2. **Interpolating-type wavelets and transforms.** We first introduce an interpolating wavelet basis using the subdivision scheme by Deslauriers and Dubuc [20,22], followed by the second-generation wavelets obtained by a lifting technique [49–51].

3.2.1. *Scaling functions and wavelets.* We begin with the successively local polynomial interpolation on the dyadic grids $\{\mathcal{G}^j\}$. Given a sampling $\big\{f_k^j := f(x_k^j),\ \forall x_k^j \in \mathcal{G}^j\big\}$ of $f(x)$, we want to predict the physical values of f on $\mathcal{F}^j$ by polynomial interpolation of degree $2N-1$. More precisely, for a specific point $y_k^j \in \mathcal{F}^j$, we choose $2N$ closest points in $\mathcal{G}^j$ (note that y_k^j and these points are all in $\mathcal{G}^{j+1}$), which could be centered around y_k^j or one-sided distributed if y_k^j is near the endpoints of the interval. For notational convenience, we denote this set of points by $\{x_l^j \in \mathcal{G}^j : l \in \Upsilon_k^j\}$, where Υ_k^j is an index set of cardinality $2N$. Further, let $\big\{h_l^j(x) : l \in \Upsilon_k^j\big\}$ be the associated Lagrange basis polynomials. It is clear that the interpolant on $\mathcal{F}^j$ is given by

$$\tilde{f}_k^j = \sum_{l\in\Upsilon_k^j} \omega_{k,l}^j f_l^j \quad \text{with} \quad \omega_{k,l}^j = h_l^j(y_k^j), \quad \forall\, l \in \Upsilon_k^j,\ \ \forall\, k \in \mathcal{F}^j. \tag{3.8}$$

In view of $\mathcal{G}^{j+1} = \mathcal{G}^j \cup \mathcal{F}^j$, we obtain the data on $\mathcal{G}^{j+1}$ by setting

$$f_{2k}^{j+1} = f_k^j, \quad f_{2k+1}^{j+1} = \tilde{f}_k^j. \tag{3.9}$$

By performing such a procedure consecutively up to any level $J \ge j$, we can obtain the interpolating scaling functions and wavelets. More precisely, the scaling function $\varphi_k^j(x)$ at level j corresponding to $x_k^j \in \mathcal{G}^j$ is obtained by feeding the data $\{f_p^j = \delta_{pk}\}_{p=0}^{2^j}$ in the above interpolating algorithm, and running it up to an infinite high level of resolution (i.e., $J \to \infty$). Due to this construction, we find

$$\varphi_k^j(x_p^j) = \delta_{kp}, \quad \forall\ x_p^j \in \mathcal{G}^j; \quad \varphi_k^j(y_p^j) = w_{p,k}^j = h_k^j(y_p^j), \quad \forall\ y_p^j \in \mathcal{F}^j, \tag{3.10}$$

and for fixed k, one-step interpolation from level j to $j+1$ yields the data:

$$\forall p \in \mathcal{G}^j, \quad f_{2p}^{j+1} = \delta_{pk}; \qquad \forall p \in \mathcal{F}^j, \quad f_{2p+1}^{j+1} = \begin{cases} w_{p,k}^j, & \text{if } \ k \in \Upsilon_p^j\,, \\ 0, & \text{if } \ k \notin \Upsilon_p^j\,. \end{cases} \tag{3.11}$$

As a matter of fact, performing this scheme with the initial data (3.11) at level $j+1$ also converges to $\varphi_k^j(x)$ (cf. [51]), so we have the two-scale relation:

$$\varphi_k^j(x) = \varphi_{2k}^{j+1}(x) + \sum_{p\in\mathcal{F}^j} w_{p,k}^j \varphi_{2p+1}^{j+1}(x), \tag{3.12}$$

where we have defined $w_{p,k}^j = 0$ if $k \notin \Upsilon_p^j$. The wavelets are defined as

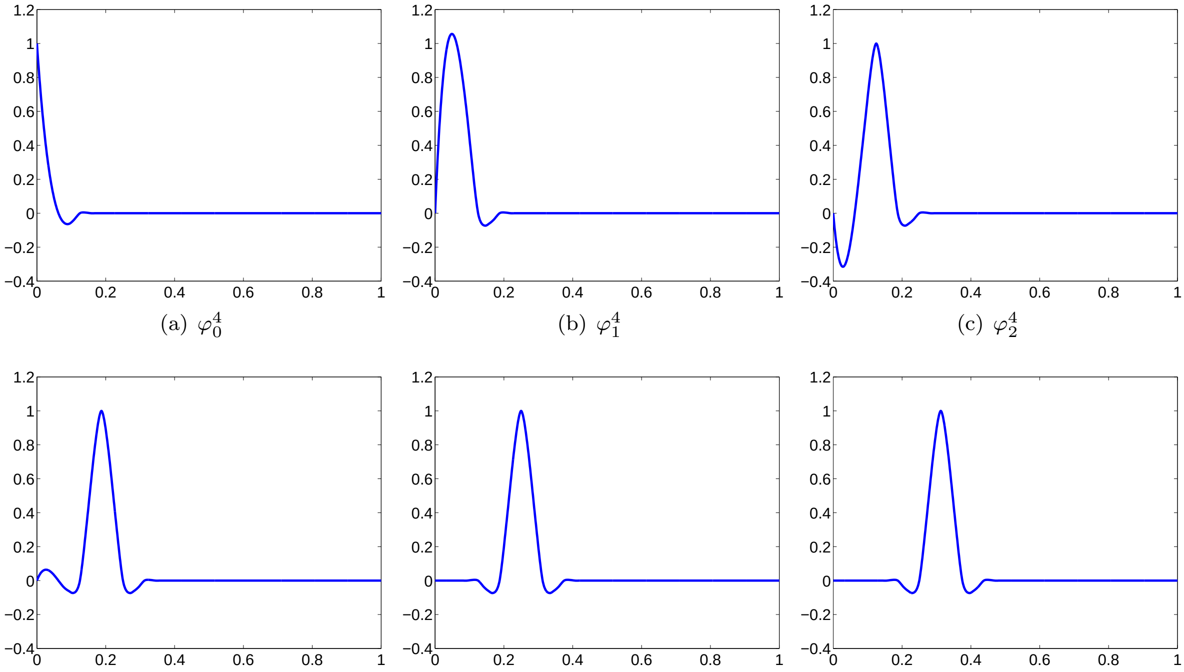
$$\psi_k^j(x) = 2\varphi_{2k+1}^{j+1}(x), \quad \forall\, k \in \mathcal{F}^j. \tag{3.13}$$

Thanks to (3.12)-(3.13), $\{\varphi_k^j, \psi_k^j\}$ generates a multi-resolution analysis as defined in (3.4)-(3.7). Some properties of $\{\varphi_k^j\}$ are summarized below:

- they are compactly supported with the support $\big[(-2N+1)2^{-j}, (2N-1)2^{-j}\big]$;
- $\varphi_k^j(x)$ satisfies the interpolation property: $\varphi_k^j(x_l^j) = \delta_{kl}$;
- linear combinations of $\varphi_k^j(x)$ reproduce the polynomials up to degree $2N-1$.

We plot in Figure 3.2 several samples of the scaling functions with $j = 4$ and $N = 2$.

Some comments on the weights $\{w_{k,l}^j\}$ are in order. Such a polynomial interpolation on uniform dyadic grids on the whole line leads to uniform weights, i.e., independent of j and k. For the finite interval, the weights are independent of the level j, and have the same values for the case when local



(a) φ_0^4 (b) φ_1^4 (c) φ_2^4

(d) φ_3^4 (e) φ_4^4 (f) φ_5^4

FIGURE 3.2. Scaling functions with $j = 4$ and $N = 2$. Note that φ_4^4 and φ_5^4 are the transition of the same function.

$2N$ points are symmetrically centered around y_k^j (i.e., $\Upsilon_k^j = \{l : -N + k + 1 \leq l \leq N + k\}$), but they take different values when y_k^j is near or at the boundaries of the interval. On the other hand, for nonuniform nested grids (such as Chebyshev-Gauss-Lobatto points), such a scheme yields weights depending on both k and j.

We tabulate in Table 3.1 some sample weights for $N = 2$, where N_{left} and N_{right} are the number of points on the left and right of y_k^j.

TABLE 3.1. Weights for $N = 2$

N_{left}	N_{right}	$w_{k,k-3}^j$	$w_{k,k-2}^j$	$w_{k,k-1}^j$	$w_{k,k}^j$	$w_{k,k+1}^j$	$w_{k,k+2}^j$	$w_{k,k+3}^j$	$w_{k,k+4}^j$
0	4					2.1875	−2.1875	1.3125	−0.3125
1	3				0.3125	0.9375	−0.3125	0.0625	
2	2			−0.0625	0.5625	0.5625	−0.0625		
3	1		0.0625	−0.3125	0.9375	0.3125			
4	0	−0.3125	1.3125	−2.1875	2.1875				

3.2.2. *Wavelet transforms.* Let $\{\mathbf{V}^j, \mathbf{W}^j\}$ be a multi-resolution analysis generated by the foregoing polynomial interpolation scaling functions and wavelets $\{\varphi_k^j, \psi_k^j\}$. For any $f^{j+1}(x) \in \mathbf{V}^{j+1}$, we have the decomposition

$$f^{j+1}(x) = f^j(x) + d^j(x), \quad f^j(x) \in \mathbf{V}^j, \quad d^j(x) \in \mathbf{W}^j. \tag{3.14}$$

Equivalently, it can be expressed in terms of the basis functions:

$$\sum_{k\in\mathcal{G}^{j+1}} c_k^{j+1}\varphi_k^{j+1}(x) = \sum_{l\in\mathcal{G}^j} c_l^j\varphi_l^j(x) + \sum_{m\in\mathcal{F}^j} d_m^j\psi_m^j(x), \tag{3.15}$$

where $\{c_k^{j+1}\}, \{c_l^j\}$ and $\{d_m^j\}$ are the expansion coefficients of f^{j+1}, f^j and d^j, respectively. Thanks to the interpolating property of the scaling functions (cf. (3.10)), we have

$$c_k^{j+1} = f^{j+1}(x_k^{j+1}), \quad c_k^j = f^j(x_k^j).$$

Hence, substituting $x = x_{2k}^{j+1}, x_{2k-1}^{j+1}$ into the equation (3.15), and using the two-scale relation (3.12) and the interpolation property (3.10), we derive the forward wavelet transform (decomposition):

$$d_k^j = \frac{1}{2}\Big(c_{2k+1}^{j+1} - \sum_{l\in\mathcal{G}^j} w_{k,l}^j c_{2l}^{j+1}\Big); \quad c_k^j = c_{2k}^{j+1}, \tag{3.16}$$

where $w_{k,l}^j = 0$ if $l \notin \Upsilon_k^j$ as before.

On the other hand, given f^j and d^j in (3.14), we can recover the expansion coefficients $\{c_k^{j+1}\}$ of f^{j+1} by the inverse wavelet transform (reconstruction):

$$c_{2k}^{j+1} = c_k^j; \quad c_{2k+1}^{j+1} = 2d_k^j + \sum_{l\in\mathcal{G}^j} w_{k,l}^j c_l^j. \tag{3.17}$$

It is worthwhile to point out that the transform can be carried out efficiently with around 2^jN operations.

Some variances of the subdivision polynomial interpolation have been considered by several authors. For example, Jain and Zhou [28] proposed to use the points not only on the coarse level but also on the same level to compute the interpolation, and Chan and Zhou [12,13] suggested the use of ENO-wavelets to avoid artificial numerical oscillations and nonphysical interpolation of the data. Such treatments can enhance the compression capability of the wavelets, but might complicate the implementation of grid adaption in multi-dimensions.

Sweldens [49] constructed some second-generation wavelets by applying a lifting technique to the existing scaling functions and wavelets, which allows to impose some desired properties such as the vanishing moments to the underlying wavelets. The basic idea is to modify the wavelets in (3.13) by adding a linear combination of the scaling functions at the same level:

$$\psi_k^{j,\mathrm{new}}(x) = \psi_k^j(x) - \sum_{l\in\tilde{\Upsilon}_k^j} \tilde{w}_{l,k}^j\varphi_l^j(x), \tag{3.18}$$

where $\{\tilde{w}_{l,k}^j\}$ are the lifting coefficients and $\tilde{\Upsilon}_k^j$ is an index set of cardinality $2\tilde{N}$ (as Υ_k^j in (3.8)), to be determined by the specified properties. Typically, we require that the new wavelets have $2\tilde{N}$ vanishing moments, namely,

$$\int_0^1 x^m\psi_k^{j,\mathrm{new}}(x)dx = 0, \quad 0 \le m \le 2\tilde{N}-1. \tag{3.19}$$

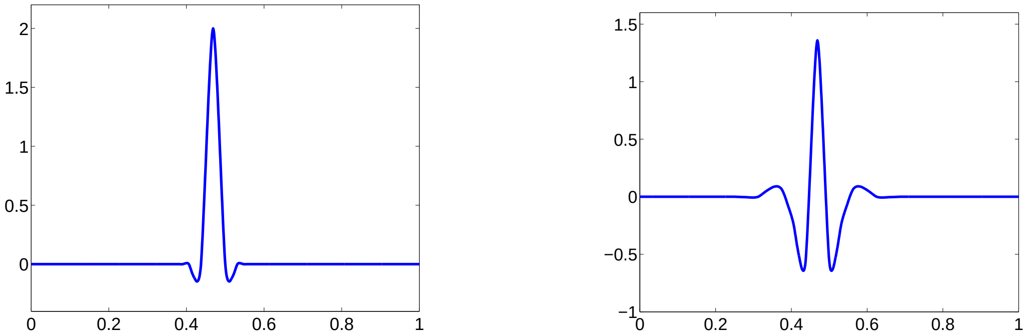
Hence, $\{\tilde{w}_{l,k}^j\}$ can be determined by the linear system:

$$2\mathcal{M}_{2k+1,m}^{j+1} - \sum_{l\in\tilde{\Upsilon}_k^j} \tilde{w}_{l,k}^j\mathcal{M}_{l,m}^j = 0, \ \ 0 \le m \le 2\tilde{N}-1, \tag{3.20}$$

where

$$\mathcal{M}_{k,m}^j := \int_0^1 x^m\varphi_k^j(x)dx.$$

We plot in Figure 3.3 a sample of the original wavelet and the lifted version with $N = \tilde{N} = 2$.



(a) Interpolating wavelet

(b) Lifted interpolating wavelet

FIGURE 3.3. $\psi_7^4(x)$ versus $\psi_7^{4,\mathrm{new}}(x)$.

The wavelet transforms involving $\{\varphi_k^{j,\mathrm{new}} = \varphi_k^j, \psi_k^{j,\mathrm{new}}\}$ can be derived in the same fashion as for (3.16) and (3.17). Basically, let $\{\tilde{c}_k^j, \tilde{c}_k^{j+1}, \tilde{d}_k^j\}$ be the expansion coefficients of $\{f^{j+1}, f^j, d^j\}$ in (3.14)-(3.15) in terms of the new basis. As the counterparts of (3.16) and (3.17), the wavelet decomposition is carried out by

$$\tilde{d}_k^j = \frac{1}{2}\Big(\tilde{c}_{2k+1}^{j+1} - \sum_{l\in\mathcal{G}^j} w_{k,l}^j \tilde{c}_{2l}^{j+1}\Big); \quad \tilde{c}_k^j = \tilde{c}_{2k}^{j+1} + \sum_{l\in\mathcal{F}^j} \tilde{w}_{k,l}^j \tilde{d}_l^j, \tag{3.21}$$

and the reconstruction can be implemented by

$$\tilde{c}_{2k}^{j+1} = \tilde{c}_k^j - \sum_{l\in\mathcal{F}^j} \tilde{w}_{k,l}^j \tilde{d}_l^j; \quad \tilde{c}_{2k+1}^{j+1} = 2\tilde{d}_k^j + \sum_{l\in\mathcal{G}^j} w_{k,l}^j \tilde{c}_{2l}^{j+1}, \tag{3.22}$$

where $\tilde{w}_{k,l}^j = 0$, if $k \notin \tilde{\Upsilon}_l^j$.

We point out that the choice of $\tilde{N}$ could be different from the parameter N in (3.8), and the lifting coefficients $\{\tilde{w}_{k,l}^j\}$ are independent of the level j. We see that with little additional cost for solving (3.20), the transforms (3.21) and (3.22) can be performed as efficiently as (3.16) and (3.17). Thanks to (3.19), the lifted wavelets provide a good low-pass filter with a better scale separation. Some other advantages were addressed in e.g., Sweldens [49, 50]. Vasilyev et al. [31, 53] adopted the lifted basis in adaptive collocation methods for simulations of fluid dynamics. However, we realize that the constant-pass filter interpolating wavelets lead to much sparser grids and solution representations for the segmentation problems, where the piecewise constant structures are mostly favored. To illustrate this, we present some numerical comparisons of the compression capability of the interpolating wavelets and lifted counterparts. Conventionally, given a threshold ϵ, we analyze the wavelet coefficients and rewrite the approximation (3.7) as

$$f^J(x) = f_{\geq}^J(x) + f_{<}^J(x), \tag{3.23}$$

where

$$f_{\geq}^J(x) = \sum_{k\in\mathcal{G}^{J_0}} c_k^{J_0}\varphi_k^{J_0}(x) + \sum_{j=J_0}^{J-1} \sum_{l\in\mathcal{F}^j,\ |d_l^j|\geq\epsilon} d_l^j \psi_l^j(x), \tag{3.24}$$

and

$$f_{<}^J(x) = \sum_{j=J_0}^{J-1} \sum_{l\in\mathcal{F}^j,\ |d_l^j|<\epsilon} d_l^j \psi_l^j(x). \tag{3.25}$$

We reconstruct f^J by $f^J_{\geq}$, and it is expected that (cf. Donoho [21]):

$$\big|f^J(x) - f^J_{\geq}(x)\big| = O(\epsilon). \tag{3.26}$$

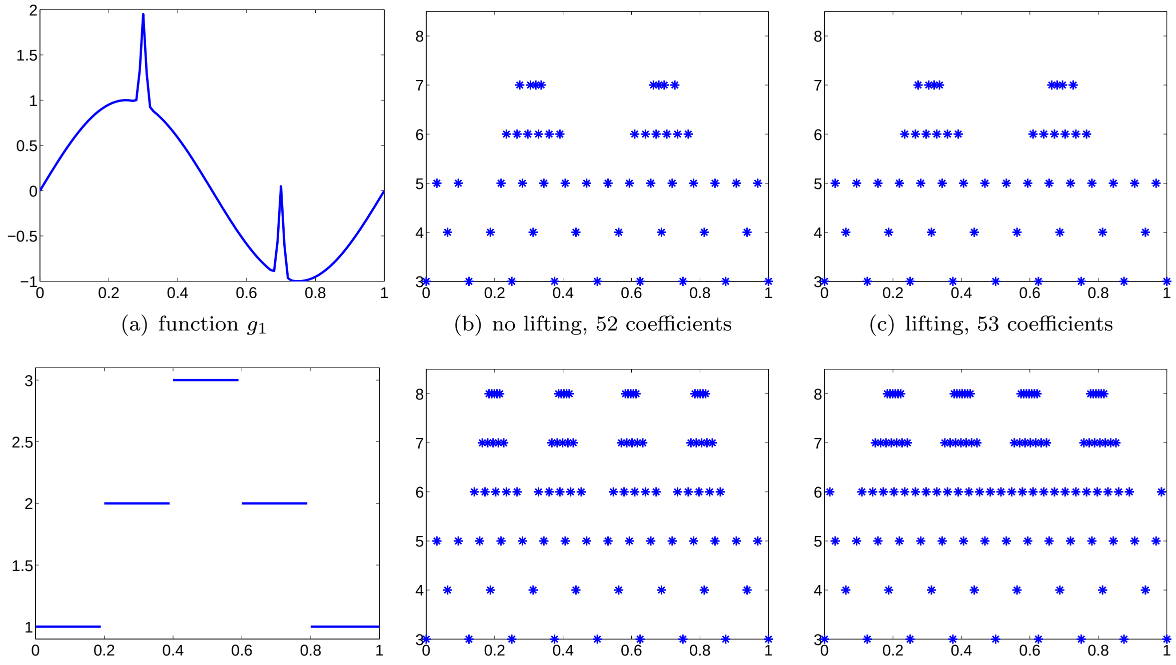
We test the compression and reconstruction of the smooth function with large variation:

$$g_1(x) = \sin(2\pi x) + \exp\big(-1000(x-0.3)^2\big) + \exp\big(-1000(x-0.7)^2\big),$$

and the piecewise-constant function

$$g_2(x) = \begin{cases} 1, & 0 \leq x < 0.2, \\ 2, & 0.2 \leq x < 0.4, \\ 3, & 0.4 \leq x < 0.6, \\ 2, & 0.6 \leq x < 0.8, \\ 1, & 0.8 \leq x \leq 1. \end{cases}$$

We plot in Figure 3.4 the distributions of the significant wavelet coefficients $\{|d^j_k| \geq \epsilon\}$ with $J_0 = 3, J = 8, N = \tilde{N} = 3$ and $\epsilon = 10^{-3}$. We see that for smooth functions with large variations, two types of wavelets essentially have similar performance, but the piece-wise constant structures, the interpolating wavelets gives sparser representations and therefore more efficient grid adaptation. The restoration in both cases confirms (3.26).



(a) function g_1

(b) no lifting, 52 coefficients

(c) lifting, 53 coefficients

(d) function g_2

(e) no lifting, 113 coefficients

(f) lifting, 135 coefficients

FIGURE 3.4. (a) and (d): test functions; (b), (c), (e) and (f): Distribution of significant wavelets coefficients for $\epsilon > 10^{-3}$; The L_∞ errors: (b) $2.244 * 10^{-3}$, (c) $1.523 * 10^{-3}$, (e) 0, (f) $1.216 * 10^{-3}$.

3.3. **Multi-dimensional wavelets and transforms.** Multidimensional wavelets can be introduced by using tensor products of one-dimensional wavelets [18, 31] or constructing non-separable multi-dimensional wavelets [17,46]. Since the underlying image domain Ω is usually rectangular, we consider tensor-product wavelets and restrict the description to two dimensional case.

Let $\{\mathcal{G}_x^j, \mathcal{F}_x^j\}_{j=J_0}^J$ and $\{\mathcal{G}_y^j, \mathcal{F}_y^j\}_{j=J_0}^J$ be the sets of nested grids in x- and y-directions, respectively, as defined in (3.1)-(3.2), and let $\{\varphi_{x,k}^j, \psi_{x,k}^j\}$ and $\{\varphi_{y,k}^j, \psi_{y,k}^j\}$ be the associated interpolating scaling functions and wavelets. Define $\mathcal{G}_{xy}^j = \mathcal{G}_x^j \otimes \mathcal{G}_y^j$, and denote the three separated subsets of $\mathcal{G}_{xy}^{j+1} \setminus \mathcal{G}_{xy}^j$ by

$$\mathcal{F}_{xy}^{j,1} = \mathcal{G}_x^j \otimes \mathcal{F}_y^j, \qquad \mathcal{F}_{xy}^{j,2} = \mathcal{F}_x^j \otimes \mathcal{G}_y^j, \qquad \mathcal{F}_{xy}^{j,3} = \mathcal{F}_x^j \otimes \mathcal{F}_y^j, \tag{3.27}$$

see Figure 3.1 (right) for an illustration. The two-dimensional scaling function is given by

$$\varphi_{i,k}^j(x,y) = \varphi_{x,i}^j(x)\varphi_{y,k}^j(y), \quad \forall (i,k) \in \mathcal{G}_{xy}^j,$$

and the wavelets are defined as

$$\forall (i,k) \in \mathcal{F}_{xy}^{j,\mu}, \quad \psi_{i,k}^{j,\mu} = \begin{cases} \psi_{x,i}^j(x)\varphi_{y,k}^j(y), & \mu = 1, \\ \varphi_{x,i}^j(x)\psi_{y,k}^j(y), & \mu = 2, \\ \psi_{x,i}^j(x)\psi_{y,k}^j(y), & \mu = 3. \end{cases} \tag{3.28}$$

Similarly, the lifted two-dimensional wavelets can be obtained by replacing ψ^j by $\psi^{j,\text{new}}$ (cf. (3.18)). The tensor-based wavelet transforms can be performed in a dimension-wise fashion, and the decomposition from level-J to level-J_0 is carried out as follows:

(i) set $j = J - 1$;
(ii) apply the one-dimensional transform (3.16) (or (3.21)) to sweep all the rows of $\mathcal{G}_{xy}^{j+1}$, i.e., in $x-$direction;
(iii) apply the transform (3.16) (or (3.21)) to the updated data and sweep all the columns in $\mathcal{G}_{xy}^{j+1}$;
(iv) set $j = j - 1$, and goto step (ii) if $j \geq J_0$.

The reconstruction process is manipulated conversely by applying (3.17) (or (3.22)) in $y-$direction then $x-$direction.

Now, we test the performance of the wavelets in image compression, and consider three typical images of size 256 × 256 in the first row of Figure 3.5. Here the compression rate is defined as

$$\text{Compression rate} = \frac{\text{Number of used wavelets (pixels)}}{\text{Size of image}} \times 100\%. \tag{3.29}$$

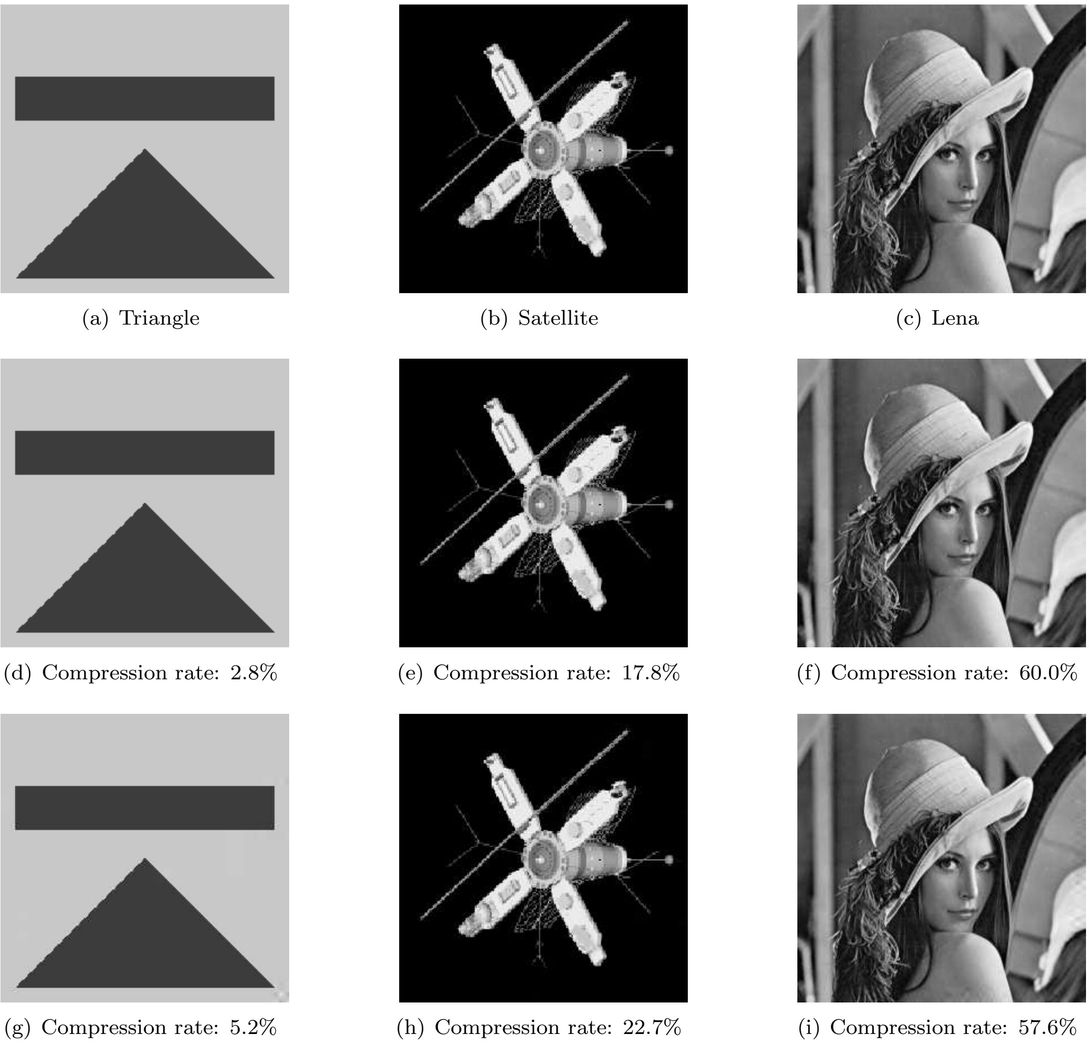
In Table 3.2, we tabulate the compression rate of two versions of wavelets with $J = 8, J_0 = 4, \tilde{N} = N$, and various threshold ϵ, where for each case, the rates of the lifted wavelets are listed in the latter column. We see that for images with more localized structures, the use of interpolating wavelets leads to a much lower compression rate, but for images with more detailed features, the saving is relatively less. This motivates us to employ the interpolating wavelets in the adaptive algorithm, which is anticipated to give more sparser representation for image segmentation.

4. Adaptive wavelet collocation algorithms

We are in a position to describe the adaptive wavelet algorithms for image segmentation and restoration models (2.18) in Section 2. For clarity of presentation, we assume that u_0 is a given image of size $(2^J + 1)^2$ (i.e., the level with finest grids), and let level J_0 be the prescribed coarsest level with $(2^{J_0} + 1)^2$ pixels. Although the algorithms can refine the grids to a level $> J$, in this application, we only use wavelets to select the significant grid points. Further, let $\phi^0 = \phi(\boldsymbol{x}, 0)$ and ϕ^m be the

TABLE 3.2. A comparison of compression rate

	Triangle				Satellite				Lena			
ϵ	$N=1$		$N=2$		$N=1$		$N=2$		$N=1$		$N=2$	
0.9	2.80	5.20	7.20	8.60	17.8	22.7	22.8	25.2	60.0	57.6	61.4	58.2
0.3	2.80	5.30	7.20	9.50	19.6	25.5	25.7	30.1	83.5	84.4	85.9	84.3
0.09	2.80	5.70	8.10	11.0	21.5	28.2	28.1	34.1	94.5	95.4	96.1	95.3
0.03	2.80	5.90	8.10	12.0	22.1	30.7	29.4	36.9	96.0	98.7	99.0	98.5



(a) Triangle

(b) Satellite

(c) Lena

(d) Compression rate: 2.8%

(e) Compression rate: 17.8%

(f) Compression rate: 60.0%

(g) Compression rate: 5.2%

(h) Compression rate: 22.7%

(i) Compression rate: 57.6%

FIGURE 3.5. Original images: top; recovered images after thresholding: middle: no lifting with $N = 1, \epsilon = 0.9$, bottom: lifting with $N = \tilde{N} = 1, \epsilon = 0.9$. Errors: (d) 0.0; (e) 0.35; (f) 1.23; (g) 0.14; (h) 0.43; (i) 0.92;

adaptive wavelet approximation of ϕ at $t_m = m\tau$ for $1 \leq m \leq M$, where τ is the time step and M is the terminated step.

We outline the adaptive wavelet algorithm as follows.

Step 0. Perform wavelet decomposition of ϕ^0 from level J to J_0, and analyze the wavelet coefficients by thresholding, which leads to the grid $\mathcal{M}_{\geq}^{0,J}$.

For $m =$ $0, 1, \cdots, M$,

Step 1. Update the computational grid $\mathcal{M}_{\geq}^{m,J}$ by adding the points whose corresponding wavelet coefficients may become significant at t_{m+1}. We denote the new grid by $\widetilde{\mathcal{M}}_{\geq}^{m,J}$.

Step 2. Solve the equation (2.18) on $\widetilde{\mathcal{M}}_{\geq}^{m,J}$ by finite difference and obtain ϕ^{m+1} with physical values on $\widetilde{\mathcal{M}}_{\geq}^{m,J}$.

Step 3. Analyze ϕ^{m+1} by computing the wavelet coefficients associated with the points in $\widetilde{\mathcal{M}}_{\geq}^{m,J}$, and remove the insignificant ones through thresholding, which yields the adapted grid $\mathcal{M}_{\geq}^{m+1,J}$. Go to **Step 1**.

Next, we present more detailed description of Steps 1-3.

Step 1.

For an evolution PDE, the solution may develop and/or dismiss local structures rapidly with time. In order to adapt to such features and to avoid numerical instability, it is necessary to expand the grid $\mathcal{M}_{\geq}^{J}$ (resulted from wavelet thresholding at the current time step) to include some points that might become significant in the next time step (cf. [25, 28, 34, 53]). Typically, the candidates are the points corresponding to neighboring wavelets (i.e., physical points). For instance, for $y_k^j \in \mathcal{M}_{\geq}^{J}$, we may add the points in

$$\Big\{ y_{k'}^{j'} \in \mathcal{F}^{j'} : j - L_1 \leq j' \leq j + L_2,\ |2^{j'-j}(k+1) - 1 - k'| \leq L_3 \Big\}, \quad L_1, L_2, L_3 \geq 0, \tag{4.1}$$

where L_1 (resp. L_2) is the number of coarser (resp. finer) levels, and L_3 is the number of points to be added at each level. The values of L_1, L_2 and L_3 depend on the evolution of the underlying solution. In this paper, we choose $L_1 = L_3 = 1$ and $L_2 = 0$, which admits a good trade-off between grid adaptation and efficiency of the implementation.

Step 2.

We now describe the finite difference to update ϕ^{m+1} on the expanded grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$ by solving the equation (2.18) with input ϕ^m on the adapted gird $\mathcal{M}_{\geq}^{m,J}$. In practice, the explicit Euler scheme is commonly used in time to evolve the nonlinear TV-equation, and the full scheme with collocation in space is formulated as

$$\phi^{m+1} = \phi^m + \tau\beta \mathrm{div}\Big(\frac{\nabla\phi^m}{|\nabla\phi^m|_\delta}\Big) + \tau F(\phi^m), \quad \forall\, \boldsymbol{x} \in \widetilde{\mathcal{M}}_{\geq}^{m,J}, \tag{4.2}$$

where

$$|\nabla\phi^m|_\delta := \sqrt{(\phi_x^m)^2 + (\phi_x^m)^2 + \delta^2} \tag{4.3}$$

is a regularization of the TV-term to avoid division by zero. We may also adopt some explicit multi-step schemes such as Runge-Kutta methods. Essentially, at each step, one needs to evaluate numerical derivatives on the incomplete grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$.

On the other hand, to relax the time-step constraint induced by an explicit method, we may utilize a semi-implicit time-stepping scheme on the linearized equation, e.g.,

$$\phi^{m+1} - \tau\beta \mathrm{div}\Big(\frac{\nabla\phi^{m+1}}{|\nabla\phi^m|_\delta}\Big) = \phi^m + \tau F(\phi^m), \qquad \forall\, \boldsymbol{x} \in \widetilde{\mathcal{M}}_{\geq}^{m,J}. \tag{4.4}$$

We also point out that the additive operator splitting (AOS) scheme has been widely applied in image processing [56], and historically, it was first developed for Navier-Stokes equations in [37, 38]. We see that such solvers need to resolve a linear system on $\widetilde{\mathcal{M}}_{\geq}^{m,J}$ at time step, which might be solved by a direct or an iterative method.

At this point, two important issues need to be addressed:

i) How to recover the values of ϕ^m (merely has physical values on $\mathcal{M}_{\geq}^{m,J}$) on $\widetilde{\mathcal{M}}_{\geq}^{m,J}$?

ii) How to compute the spatial derivatives on the incomplete grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$?

The first question will be answered in the description of **Step 3** below, so we first assume that ϕ^m is well-defined on the expanded grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$, and discuss the second issue by considering the finite difference approximation of the TV-term in (4.2). We present two approaches.

Firstly, for any point $\boldsymbol{x} = (x_i, y_j) \in \widetilde{\mathcal{M}}_{\geq}^{m,J}$, let $\phi_{i,j}^m$ be the finite difference approximation, and let h be the (regular) mesh size. The central difference discretization of the TV-term reads

$$\frac{1}{h^2}\Big(C^m_{i+\frac{1}{2},j}(\phi^m_{i+1,j}-\phi^m_{i,j}) - C^m_{i-\frac{1}{2},j}(\phi^m_{i,j}-\phi^m_{i-1,j}) + D^m_{i,j+\frac{1}{2}}(\phi^m_{i,j+1}-\phi^m_{i,j}) - D^m_{i,j-\frac{1}{2}}(\phi^m_{i,j}-\phi^m_{i,j-1})\Big), \quad (4.5)$$

where the coefficients

$$C^m_{i+\frac{1}{2},j} = \left((\Delta^x_+\phi^m_{i,j})^2 + \frac{(\Delta^y_+\phi^m_{i+\frac{1}{2},j})^2}{2} + \frac{(\Delta^y_-\phi^m_{i+\frac{1}{2},j})^2}{2} + \delta^2 \right)^{-\frac{1}{2}}, \quad (4.6)$$

and

$$D^m_{i,j+\frac{1}{2}} = \left((\Delta^y_+\phi^m_{i,j})^2 + \frac{(\Delta^x_+\phi^m_{i,j+\frac{1}{2}})^2}{2} + \frac{(\Delta^x_-\phi^m_{i,j+\frac{1}{2}})^2}{2} + \delta^2 \right)^{-\frac{1}{2}}. \quad (4.7)$$

Here, $\Delta^x_+, \Delta^x_-, \Delta^y_+$ and Δ^y_- are forward and backward differences in x-direction and y-direction, respectively, and

$$\phi^m_{i+\frac{1}{2},j} = \frac{1}{2}\phi^m_{i,j} + \frac{1}{2}\phi^m_{i+1,j}, \qquad \phi^m_{i,j+\frac{1}{2}} = \frac{1}{2}\phi^m_{i,j} + \frac{1}{2}\phi^m_{i,j+1}. \quad (4.8)$$

We see that (4.5) involves eight neighboring points, which might not sit in $\widetilde{\mathcal{M}}_{\geq}^{m,J}$. In this case, we interpolate the physical values of the missing points from a coarser level as described in Algorithm 1 of **Step 3** below. However, this technique is not appropriate for the AOS scheme, as it will destroy the tridiagonal property of the system. In this case, it is preferable to use finite differences on nonuniform grids but with a considerable complication in implementation.

Again, we consider the finite difference discretization of the TV-term at $\boldsymbol{x} = (\xi_p, \eta_q) \in \widetilde{\mathcal{M}}_{\geq}^{m,J}$. Denote by $(\xi_{p-1}, \eta_q) \in \widetilde{\mathcal{M}}_{\geq}^{m,J}$ (resp. (ξ_{p+1}, η_q)) the left (resp. right) neighboring point in x-direction, and likewise for (ξ_p, η_{q-1}) and (ξ_p, η_{q+1}) in y-direction. Let $h_{x,p} = \xi_{p+1} - \xi_p$ and $h_{y,q} = \eta_{q+1} - \eta_q$. The finite difference approximation is

$$\begin{aligned}\operatorname{div}\Big(\frac{\nabla\phi^{m+1}}{|\nabla\phi^m|_\delta}\Big)\Big|_{(\xi_p,\eta_q)} &\sim \frac{2}{h_{x,p}+h_{x,p-1}}\Big(C^m_{p+\frac{1}{2},q}\frac{\phi^{m+1}_{p+1,q}-\phi^{m+1}_{p,q}}{h_{x,p}} - C^m_{p-\frac{1}{2},q}\frac{\phi^{m+1}_{p,q}-\phi^{m+1}_{p-1,q}}{h_{x,p-1}}\Big)\\ &+ \frac{2}{h_{y,q}+h_{y,q-1}}\Big(D^m_{p,q+\frac{1}{2}}\frac{\phi^{m+1}_{p,q+1}-\phi^{m+1}_{p,q}}{h_{y,q}} - D^m_{p,q-\frac{1}{2}}\frac{\phi^{m+1}_{p,q}-\phi^{m+1}_{p,q-1}}{h_{y,q-1}}\Big),\end{aligned} \quad (4.9)$$

where the coefficients $C^m_{p+\frac{1}{2},q}$ and $D^m_{p,q+\frac{1}{2}}$ can be computed by (4.6) and (4.7), or by finite difference on nonuniform grids in a similar fashion (note: in this case, an interpolation from $\mathcal{M}_{\geq}^{m,J}$ to $\widetilde{\mathcal{M}}_{\geq}^{m,J}$ is also needed).

Step 3.

The purpose of this step is to analyze the solution ϕ^{m+1} obtained from **Step 2** to generate the new grid $\mathcal{M}_{\geq}^{m+1,J}$ for the next time step. This procedure involves an interpolation which also makes the difference (4.5)-(4.7) feasible.

Let us focus on the interpolation process first. For clarity of presentation, we mainly describe the one-dimensional implementation, and the idea can be extended to multiple dimensions straightforwardly. Hereafter, the notation is the same as before (cf. (3.1), (3.2), (3.8) and (3.16)), and recall that the coefficients $\{c_k^j\}$ associated with the scaling functions are physical values of ϕ^{m+1} at $\{x_k^j\} \subset \mathcal{G}^j$. Our objective is to calculate $\phi^{m+1}(x_l^j)$ for $x_l^j \notin \widetilde{\mathcal{M}}_{\geq}^{m,J}$. In view of the dyadic structure of the grids and the fact $\mathcal{G}^{J_0} \subset \widetilde{\mathcal{M}}_{\geq}^{m+1,J}$, the point $x_l^j \in \mathcal{F}^{j'}$ and corresponds to $y_m^{j'}$, i.e., $x_l^j = y_m^{j'} = x_{2m+1}^{j'+1}$ (cf. (3.3)). Thus, we have $\phi^{m+1}(x_l^j) = \phi^{m+1}(x_{2m+1}^{j'+1}) = c_{2m+1}^{j'+1}$. Notice that $|d_m^{j'}| \leq \epsilon$, so we compute $\phi^{m+1}(x_l^j)$ by (3.17):

$$c_{2m+1}^{j'+1} = 2d_m^{j'} + \sum_{n \in \Upsilon_m^{j'}} w_{m,n}^{j'} c_n^{j'+1} \sim \sum_{n \in \Upsilon_m^{j'}} w_{m,n}^{j'} c_n^{j'} = \sum_{n \in \Upsilon_m^{j'}} w_{m,n}^{j'} \phi^{m+1}(x_n^{j'}), \tag{4.10}$$

where if $x_n^{j'} \notin \widetilde{\mathcal{M}}_{\geq}^{m,J}$, then we apply the above procedure downwards recursively to recover $\phi^{m+1}(x_n^{j'})$. The recursion would be terminated at the coarsest level-J_0, where all points $x_l^{J_0} \in \widetilde{\mathcal{M}}_{\geq}^{m,J}$. We summarize the above in **Algorithm 1**.

```
Algorithm 1.
  Input the grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$, the physical values $\phi^{m+1}(\widetilde{\mathcal{M}}_{\geq}^{m,J})$ and the point $x_l^j$. Output $\phi^{m+1}(x_l^j)$.
   (i)  If $x_l^j \in \widetilde{\mathcal{M}}_{\geq}^{m,J}$, return $\phi^{m+1}(x_l^j)$.
  (ii)  If $x_l^j \notin \widetilde{\mathcal{M}}_{\geq}^{m,J}$, find $j', m$ such that $y_m^{j'} = x_l^j$.
 (iii)  For $\forall n \in \Upsilon_m^{j'}$, apply this algorithm recursively to compute $\phi^{m+1}(x_n^{j'})$.
  (iv)  Compute
            $\phi^{m+1}(x_l^j) \sim \sum_{n \in \Upsilon_m^{j'}} w_{m,n}^{j'} \phi^{m+1}(x_n^{j'})$.
```

Now, we are ready to carry out **Step 3**, that is, to generate a new grid $\mathcal{M}_{\geq}^{m+1,J}$ by applying wavelet thresholding analysis to ϕ^{m+1} obtained from **Step 2**. In other words, we have to find the wavelet decomposition of ϕ^{m+1} on the incomplete grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$. For clarity, we still focus on one-dimensional implementation. By (3.16), the wavelet coefficient d_k^j is computed from $c_{2k+1}^{j+1} = \phi^{m+1}(x_{2k+1}^{j+1})$, and the closest $2N$ physical values $c_{2l}^{j+1} = \phi^{m+1}(x_{2l}^{j+1})$ for all $l \in \Upsilon_k^j$. If $x_{2l}^{j+1} \notin \widetilde{\mathcal{M}}_{\geq}^{m,J}$, we call **Algorithm 1** to interpolate its value $\phi^{m+1}(x_{2l}^{j+1})$. The algorithm for generating the new grid $\mathcal{M}_{\geq}^{m+1,J}$ is summarized in **Algorithm 2**.

Alternatively, we may directly perform the decomposition of ϕ^{m+1} on the incomplete grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$ by solving the linear system

$$\sum_{l \in \mathcal{G}^{J_0}} c_l^{J_0} \varphi_l^{J_0}(x_{k'}^{j'}) + \sum_{j,k,y_k^j \in \widetilde{\mathcal{M}}_{\geq}^{m,J}} d_k^j \psi_k^j(x_{k'}^{j'}) = \phi^{m+1}(x_{k'}^{j'}), \quad \forall\, x_{k'}^{j'} \in \widetilde{\mathcal{M}}_{\geq}^{m,J}. \tag{4.11}$$

This takes the advantage of the interpolating wavelets:

$$\varphi_l^{J_0}(x_{k'}^{J_0}) = \delta_{J_0,k'}, \quad \psi_k^j(x_{k'}^{j'}) = 2\varphi_{2k+1}^{j+1}(x_{k'}^{j'}) = 2\delta_{2k+1,k'}\delta_{j+1,j'}, \quad j' \leq j+1,$$

which can be solved from the coarsest level-J_0 up to the finest level-$J-1$ by backward substitution. More precisely, we first take $x_{k'}^{j'} = x_l^{J_0}$ for $l \in \mathcal{G}^{J_0}$ in (4.11). This leads to $c_l^{J_0} = \phi^{m+1}(x_l^{J_0})$, so by (4.11),

$$2d_k^{J_0} = \phi^{m+1}(x_{2k+1}^{J_0+1}) - \sum_{l \in \mathcal{G}^{J_0}} c_l^{J_0} \varphi_l^{J_0}(x_{2k+1}^{J_0+1}) \quad \text{for} \quad y_k^{J_0} \in \widetilde{\mathcal{M}}_{\geq}^{m,J}.$$

Setting $j' = J_0 + 1$, we can find $\{d_k^{J_0+1}\}$ by substituting $\{c_l^{J_0}\}$ and $\{d_k^{J_0}\}$ into (4.11). Repeating this procedure gives all the wavelets coefficients on $\widetilde{\mathcal{M}}_{\geq}^{m,J}$. It is worthwhile to point out that this algorithm is easier to implement and extend to higher dimension, but it requires to compute the scaling function ϕ at the points of all levels.

Algorithm 2.

```
Input the grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$ and the physical values $\phi^{m+1}(\widetilde{\mathcal{M}}_{\geq}^{m,J})$. Output the new grid $\mathcal{M}_{\geq}^{m+1,J}$ .
Set $c_k^{J_0} = \phi^{m+1}(x_k^{J_0})$, for all $k \in \mathcal{G}^{J_0}$;
For $j = J-1, \cdots, J_0$,
    For all $k \in \mathcal{F}^j$,
        If $y_k^j \notin \widetilde{\mathcal{M}}_{\geq}^{m,J}$, then $d_k^j = 0$;
        If $y_k^j \in \widetilde{\mathcal{M}}_{\geq}^{m,J}$,
            For all $l \in \Upsilon_k^j$, compute $\phi^{m+1}(x_{2l}^{j+1})$ by Algorithm 1;
            Compute $d_k^j$ by (3.16);
    End
End
Analyze the wavelet coefficients by thresholding and derive the new grid $\mathcal{M}_{\geq}^{m+1,J}$.
```

5. Numerical Results

We present in this section a number of numerical examples to validate the efficiency of the adaptive algorithms for the TV-Allen-Cahn image segmentation and ROF image denoising. We shall see that the proposed methods lead to nearly-optimal computational grids, which are adapted to the local structures of the image. Hence, the use of this technique can relax the stiffness of the systems, and can largely reduce the computational memory. Moreover, the methods are also not sensitive to the choice of thresholding parameters.

5.1. **Examples of image segmentation.** We start with the test of the adaptive methods for the (2.13). We use a first-order operator splitting scheme and employ an explicit time discretization to integrate the equation (2.15). As usual, we utilize the quantity signal-to-noise-ratio (SNR) to indicate the amount of noise in the observed data,

$$\mathbf{SNR} = \frac{\text{variance of data}}{\text{variance of noise}},$$

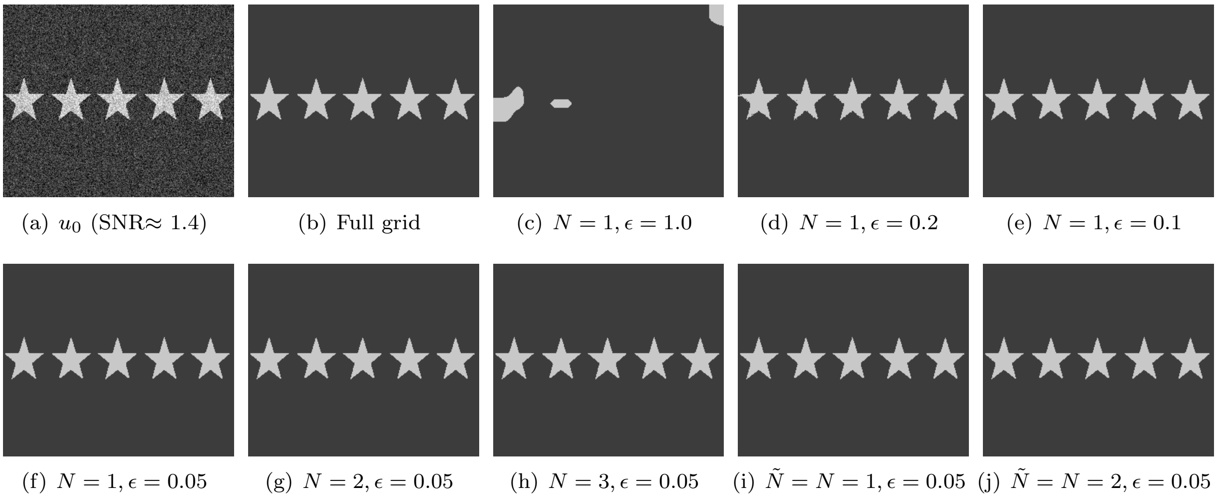
and adopt the compression rate defined in (3.29) to qualify the grid compression and adaptation. The stopping criteria is based on the l_2 error of the computational phase value vector $\mathbf{c}$ between two consecutive time steps less than 10^{-4}, which ensures the fidelity of the data in the minimization of cost functional.

We first consider a two-phase segmentation with a noise (with SNR 1.4) five-star image of size 300×250. In the following computations, the parameters are chosen as: (i) five-level wavelet transforms, i.e., $J - J_0 = 5$, (ii) the regularization parameter in the TV-term (cf. (4.3)): $\delta^2 = 1e - 16$, and (iii) $\beta = 6.48e + 7$ and $\gamma = 2e + 8$. Notice that since the parameter β in the leading TV-term of (2.15) is big and δ^2 is close to machine zero, the time step must be small and we take $\tau = 5e - 11$ (which is roughly $1e-3$ by scaling out β). We test the methods based on the interpolating wavelets with local supports $N = 1, 2$ and with various thresholding parameter ϵ. We also compare the performance against the full grid in terms of compression rate, number of iterations and quality of segmentation.

We plot in Figure 5.1 the given noise image u_0 and the segmented images obtained from the adaptive methods using two types of wavelets with different threshold ϵ and local support parameter $N/\tilde{N}$. In all cases, the initial phase function ϕ^0 is given in Figure 5.3(a).To illustrate the adaptation of the grids and compression of the wavelets, we depict in Figure 5.2 the grid distribution at various time steps, and tabulate in Table 5.1 the compression rates for various cases, where the last column is the number of steps needed to meet the stopping criteria. We also plot in Figure 5.3(b) the snapshot of the level set function at the final step, which converges to a piecewise constant function segmenting the given noise image. In Figure 5.3(c-d), we plot the numerical energy of the cost functional:

$$\int\limits_{\Omega} |u - u_0|^2 d\boldsymbol{x} + \beta \int\limits_{\Omega} |\nabla \phi| d\boldsymbol{x} + \frac{\gamma}{4} \int_{\Omega} (\phi^2 - 1)^2 d\boldsymbol{x}.$$

for regular full grid finite difference and the adaptive algorithms. In all cases, the energy decreases with time, and the proposed methods give slightly smaller numerical energy.


(a) u_0 (SNR≈ 1.4) (b) Full grid (c) $N = 1, \epsilon = 1.0$ (d) $N = 1, \epsilon = 0.2$ (e) $N = 1, \epsilon = 0.1$

(f) $N = 1, \epsilon = 0.05$ (g) $N = 2, \epsilon = 0.05$ (h) $N = 3, \epsilon = 0.05$ (i) $\tilde{N} = N = 1, \epsilon = 0.05$ (j) $\tilde{N} = N = 2, \epsilon = 0.05$

FIGURE 5.1. Segmentation for fivestar image. (a): Observed image; (b)-(j): Segmented images.

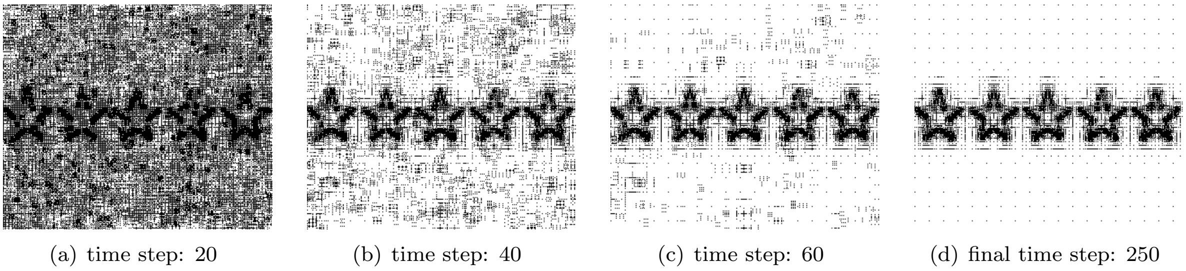
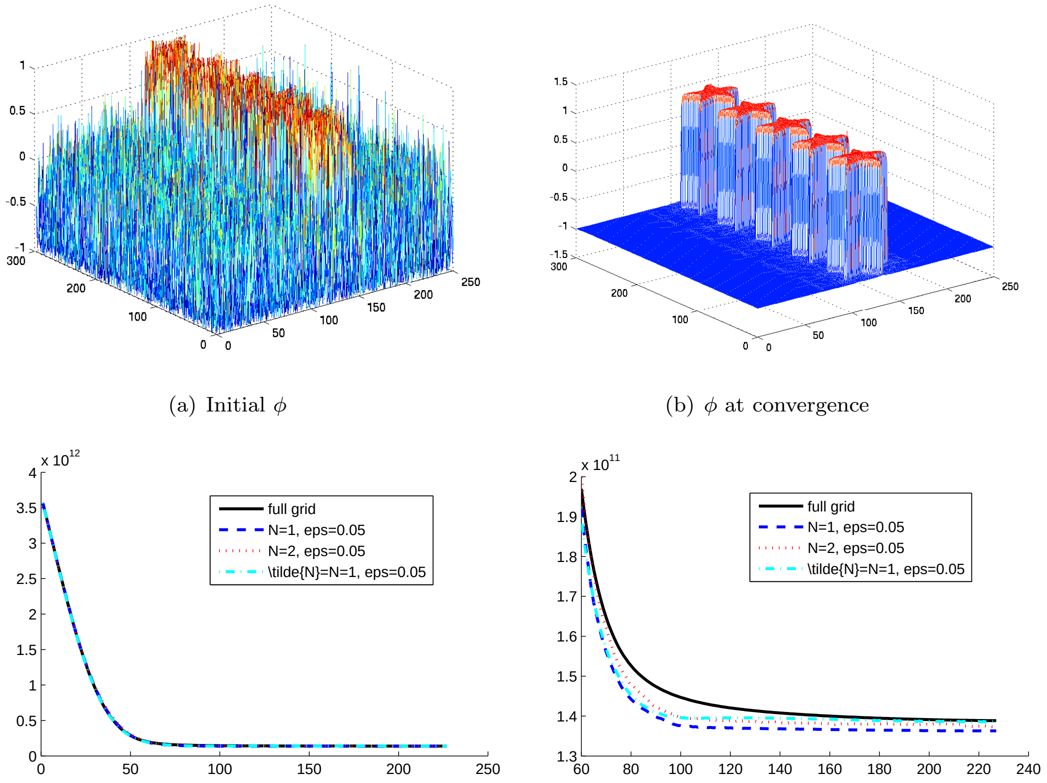

(a) time step: 20 (b) time step: 40 (c) time step: 60 (d) final time step: 250

FIGURE 5.2. Illustration of the computational grids at different time with $N = 1$ and $\epsilon = 0.1$.



(c) Decay of energy: 1-240 steps

(d) Decay of energy: 60-240 steps

FIGURE 5.3. Profiles of the initial and final phase functions with $N = 1$ and $\epsilon = 0.1$: (a)-(b), and the decay of numerical energy: (c)-(d).

TABLE 5.1. Compression rate of computational grid at different time step.

	20	30	40	50	60	70	80	final	steps
$N = 1, \epsilon = 0.2$	18.80	13.52	11.20	10.11	9.62	9.32	9.21	9.17	268
$N = 1, \epsilon = 0.1$	56.25	33.32	21.21	15.89	13.50	12.62	12.13	11.65	250
$N = 1, \epsilon = 0.05$	87.66	62.25	37.52	23.39	17.33	14.84	13.84	12.33	271
$N = 2, \epsilon = 0.05$	90.15	65.45	40.46	25.60	18.66	16.06	15.11	14.42	445
$N = 3, \epsilon = 0.05$	91.37	67.63	42.82	27.84	20.30	17.29	16.07	14.87	253
lifted $\tilde{N} = N = 1, \epsilon = 0.05$	99.48	89.14	59.51	35.77	24.89	21.04	19.77	18.77	446
lifted $\tilde{N} = N = 2, \epsilon = 0.05$	99.81	94.77	70.69	44.16	30.24	25.57	23.68	22.39	301

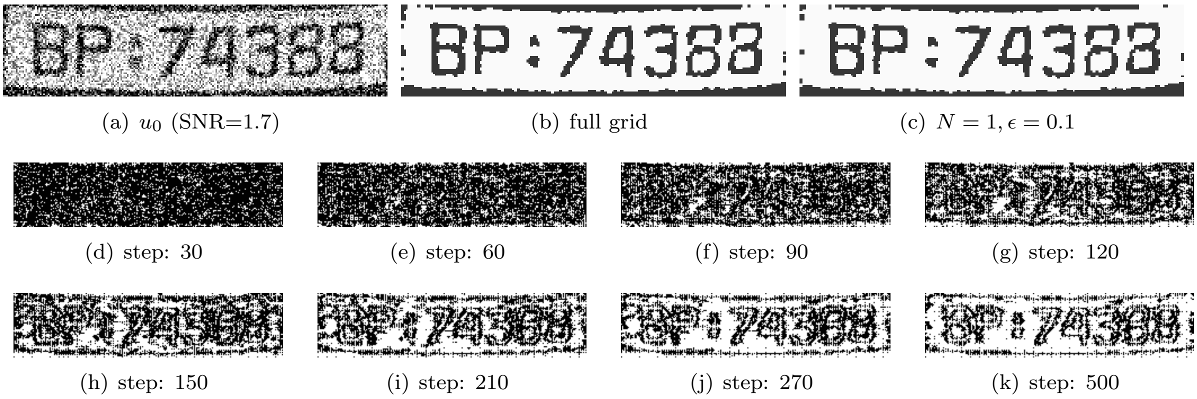
These results show that

(i) the adaptive algorithm for TV-Allen-Cahn segmentation is not so sensitive to the choice of the thresholding parameter ϵ, as satisfactory results can usually be obtained by setting $0.05 \leq \epsilon \leq 0.3$;

(ii) the use of wavelets with smaller supports leads to higher compression rate (i.e., sparser computational grids), and normally for image segmentation, the choice $N = 1$ gives nearly-optimal grids. Moreover, as with the tests in the previous section, the lifted wavelets oftentimes do not really improve the overall performance of the algorithms. Notice that the **Algorithms 1** and **2** in the previous section should be slightly modified for lifted wavelets. One might use perfect reconstruction technique proposed in [53] to add some points into computational grid $\widetilde{\mathcal{M}}_{\geq}^{m,J}$. This would include some extra grids so it may reduce compression rate of grid. However, the intrinsic vanishing moments of the lifted wavelets are low-pass filters, so these might be useful for imaging denoising as to be shown later on;

(iii) the proposed method usually outperforms the finite difference method on a uniform grids in terms of iterations particular for problems with locally structured solutions. Here, no comparison was made in terms of computational time, since our current implementation is not based on sparse data structure. However, the method should be competitive also in CPU time.

As a second example, we test the model and adaptive algorithm on a natural car-plate image of size 239 × 57. In the computation, we use three level wavelet transforms: $J - J_0 = 3$, and take $\delta^2 = 1e - 16$, $\beta = 4.08e6$, $\gamma = 1e7$, $\tau = 1e - 8$, $N = 1$ and $\epsilon = 0.1$. In Figure 5.4, we plot the images segmented by the TV-Allen-Cahn models using regular finite difference on uniform grids and the proposed adaptive method, and we also depict the dynamical distribution of the computational grids. As with the previous example, we find the use of the interpolating wavelets with the smallest support $N = 1$ produces a high compression rate and reduces about 50 iterations when compared with the regular finite difference method.


(a) u_0 (SNR=1.7) (b) full grid (c) $N = 1, \epsilon = 0.1$

(d) step: 30 (e) step: 60 (f) step: 90 (g) step: 120

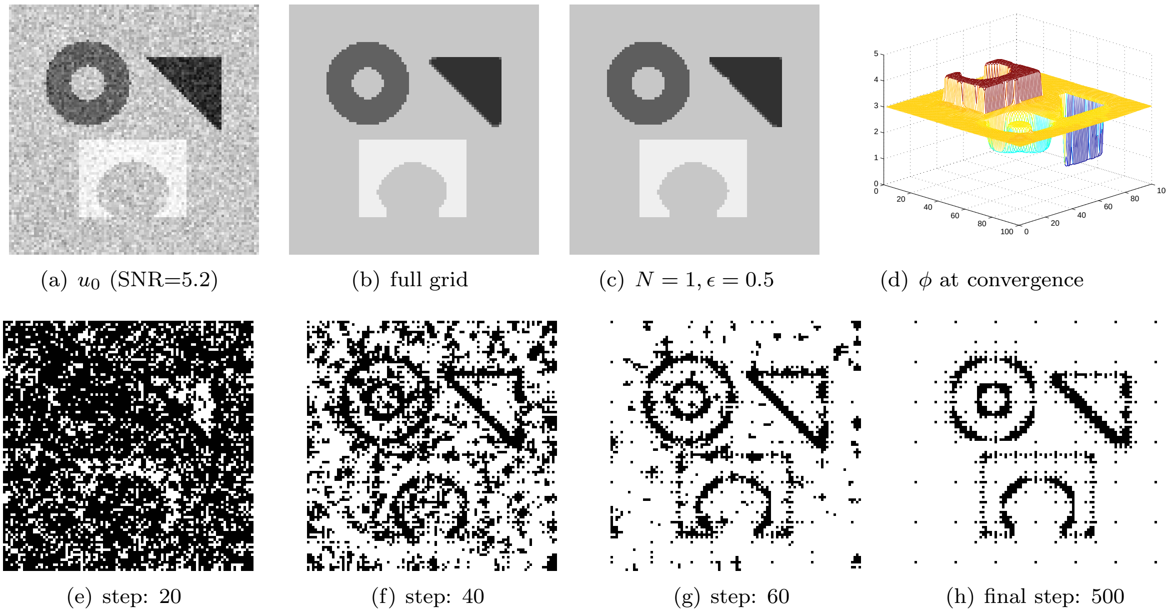
(h) step: 150 (i) step: 210 (j) step: 270 (k) step: 500

FIGURE 5.4. (a): given noise image; (b)-(c): segmented images; (d)-(k): distributions of the computational grids at different time steps, and the number of points used in the final step is about 27% of the given image size.

Next, we test a four-phase segmentation problem, where the image contains three objects as shown in Figure 5.5(a). In the computation, we take $J - J_0 = 4$, $\delta^2 = 1e - 16$, $\beta = 1.62e7$, $\gamma = 5e6$, $\tau = 6e - 10$, $N = 1$ and $\epsilon = 0.5$. In Figure 5.5(d), we plot the phase function ϕ at convergence, which approximates the piecewise constant phase values $\phi = 1 \vee 2 \vee 3 \vee 4$ as expected. The quality of segmentation compared with the full grids is demonstrated in Figures 5.5(b) and 5.5(c). Similar performance is observed as with the two-phase cases.

5.2. **Examples of image denoising.** Next, we test the algorithms on ROF denoising model (2.17). As usual, we use the peak-signal-to-noise (PSNR) as a criteria for the quality of restoration, which is expressed

$$\mathrm{PSNR} = 10\log_{10}\Big(\frac{\sum_{\mathrm{i,j}=1} 255^2}{\sum_{\mathrm{i,j}=1}(\phi_{\mathrm{i,j}} - \mathrm{u}_{0_{\mathrm{i,j}}})^2}\Big)$$



(a) u_0 (SNR=5.2) (b) full grid (c) $N = 1, \epsilon = 0.5$ (d) ϕ at convergence

(e) step: 20 (f) step: 40 (g) step: 60 (h) final step: 500

FIGURE 5.5. (a) given noise image; (b)-(c): segmented images; (d): the profile of the level set function at convergence; (e)-(h): distributions of the computational grids at different time steps, and the number of points used in the final step is about 38% of the given image size.

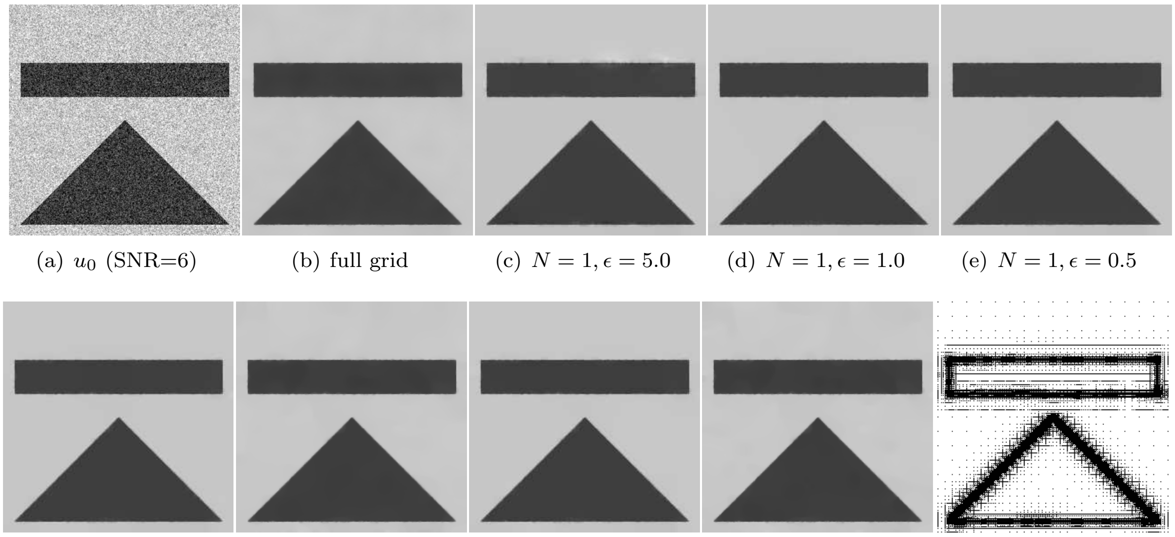
where $\{\phi_{i,j} - u_{0_{i,j}}\}$ are the difference of the pixel values between the restored and given image u_0. In this context, the stopping rule is based on the errors of the numerical TV-energy

$$E^k_{TV} = \int_\Omega \beta|\nabla u^k| + \frac{1}{2}(u^k - u_0)^2 d\boldsymbol{x},$$

between two subsequential steps such that $|E^{m+1}_{TV} - E^m_{TV}| < 0.5$. We take three typical images: rectangle-bar (256×256), satellite (512×512) and Lena (512×512).

For the first example, we choose the parameters $J - J_0 = 4$, $\delta^2 = 1e - 16$, $\beta = 244.76$, and $\tau = 4e - 4$. Firstly, we fix the support $N = 1$ to examine the influence of thresholding parameter ϵ. We demonstrate in Figure 5.6 the restored images obtained by regular finite difference method on full grid and the adaptive wavelet methods with various choices of wavelets. We observe that the proposed methods provide reliable restoration in all cases. Due to the vanishing moments, the lifted wavelets normally produces slightly better results with relatively higher PSNR values (cf. Table 5.2), but might lead to denser computational grids (cf. Table 5.3). Moreover, Table 5.3 also shows of the reduction of the number of iterations (compared with the regular finite difference method).

In the last two examples, we apply the methods with $J - J_0 = 4$, $\delta^2 = 1e{-}16$, $\beta = 24.48$ and $\tau = 4e - 4$. The original images and restored ones are plotted in Figure 5.7 and Figure 5.8. We see that the quality of restoration of different methods is almost the same, and similar advantages are observed as previous tests. It is anticipated that for images full of finer scales, e.g., Lena, the strength of adaptive algorithms is not as much as for those with simple structures. However, the use of lifted wavelets together with ROF models usually produce a better restoration than a single technique.



(a) u_0 (SNR=6) (b) full grid (c) $N = 1, \epsilon = 5.0$ (d) $N = 1, \epsilon = 1.0$ (e) $N = 1, \epsilon = 0.5$

(f) $N = 1, \epsilon = 0.2$ (g) $N = 2, \epsilon = 0.2$ (h) $\tilde{N} = N = 1, \epsilon = 0.2$ (i) $\tilde{N} = N = 2, \epsilon = 0.2$ (j) grid $N = 1, \epsilon = 0.2$

FIGURE 5.6. (a): noise image u_0; (b)-(i): restored images; (j): grid distribution with $N = 1, \epsilon = 0.2$, and the compression rate is 23.30%.

TABLE 5.2. Comparison of PSNR

	full grid	$N = 1$			$N = 2$			$N = 3$		
ϵ		0.5	0.2	0.1	0.5	0.2	0.1	0.5	0.2	0.1
PSNR	34.08	33.67	33.89	34.20	33.96	34.07	34.23	33.90	34.34	34.19
		$\tilde{N} = N = 1$			$\tilde{N} = N = 2$			$\tilde{N} = N = 3$		
ϵ		0.5	0.2	0.1	0.5	0.2	0.1	0.5	0.2	0.1
PSNR		34.50	34.19	34.30	34.08	34.28	34.14	34.11	34.14	34.36

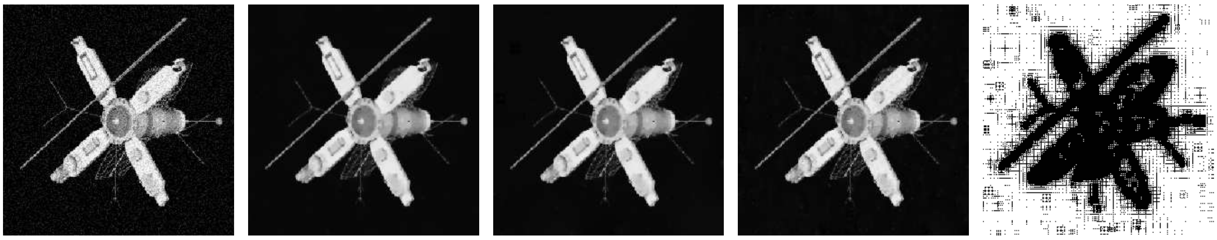
TABLE 5.3. Comparison of grid compression

ϵ	150	200	250	300	350	400	final	steps
full grid	100	100	100	100	100	100	100	493
$N = 1, \epsilon = 1.0$	97.52	76.19	42.16	24.31	20.34	19.30	19.10	418
$N = 1, \epsilon = 0.5$	99.69	89.66	61.30	34.22	24.17	21.73	20.67	475
$N = 1, \epsilon = 0.2$	99.98	97.27	80.46	54.83	34.25	25.42	23.30	482
$N = 1, \epsilon = 0.1$	100	99.14	89.47	68.80	47.12	31.70	26.33	460
lifted $\tilde{N} = N = 1, \epsilon = 0.5$	100	99.34	84.41	53.33	41.10	37.83	37.57	410
lifted $\tilde{N} = N = 1, \epsilon = 0.2$	100	99.95	96.41	76.17	52.40	42.36	40.14	464
lifted $\tilde{N} = N = 2, \epsilon = 0.5$	100	99.54	89.08	66.28	54.69	49.88	48.10	488
lifted $\tilde{N} = N = 2, \epsilon = 0.2$	100	99.98	97.33	83.15	66.55	56.81	53.17	464

REFERENCES

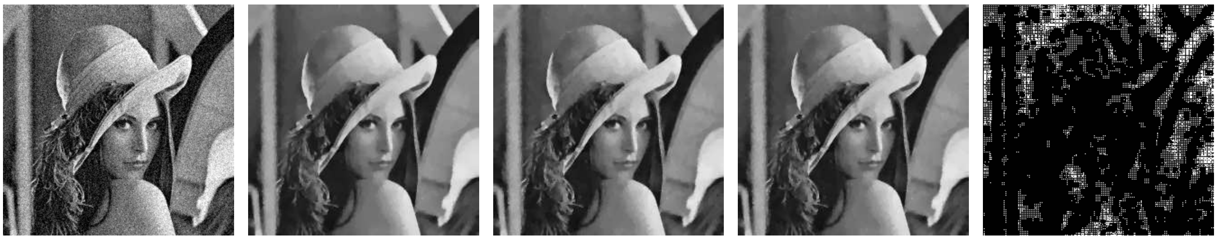
[1] S.M. Allen and J.W. Cahn. A microscope theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metall.*, 27(6):1085–1095, 1979.

[2] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi. Adaptive solution of partial differential equations in multiwavelet bases. *Journal of Computational Physics*, 182(1):149–190, 2002.



(a) u_0 (SNR=26.37) (b) full grid (c) $N = 1, \epsilon = 1.0$ (d) $\tilde{N} = N = 1, \epsilon = 1.0$ (e) grid $N = 1, \epsilon = 1.0$

FIGURE 5.7. (a): noise image u_0; (b)-(d): restored images with PSNR: 27.35, 27.24, 27.71, respectively; (e) grid distribution with $N = 1, \epsilon = 1.0$, and the compression rate is 40%.



(a) u_0 (SNR=8.93) (b) full grid (c) $N = 1, \epsilon = 0.5$ (d) $\tilde{N} = N = 1, \epsilon = 0.5$ (e) grid $N = 1, \epsilon = 0.5$

FIGURE 5.8. (a): noise image u_0; (b)-(d): restored images with PSNR: 28.03, 28.40, 28.40, respectively; (e) grid distribution with $N = 1, \epsilon = 1.0$, and the compression rate is 89%.

[3] E. Bae and X.C. Tai. Graph cuts for the multiphase Mumford-Shah model using piecewise constant level set methods. *UCLA CAM Report 08*, 36, 2008.

[4] E. Bae and X.C. Tai. Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. *Scale Space and Variational Methods in Computer Vision*, pages 1–13, 2009.

[5] W. Bangerth and R. Rannacher. *Adaptive finite element methods for differential equations.* Birkhäuser, 2003.

[6] M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984.

[7] G. Beylkin and J.M. Keiser. On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. *Journal of Computational Physics*, 132(2):233–259, 1997.

[8] W. Cai and J. Wang. Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear PDEs. *SIAM Journal on Numerical Analysis*, 33(3):937–970, 1996.

[9] W. Cai and W. Zhang. An adaptive spline wavelet ADI (SW-ADI) method for two-dimensional reaction-diffusion equations. *Journal of Computational Physics*, 139(1):92–126, 1998.

[10] H.D. Ceniceros and T.Y. Hou. An efficient dynamically adaptive mesh for potentially singular solutions. *Journal of Computational Physics*, 172(2):609–639, 2001.

[11] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[12] T.F. Chan and H.M. Zhou. Total variation improved wavelet thresholding in image compression. In *Proc. Seventh International Conference on Image Processing*, volume 2, pages 391–394. Citeseer, 2000.

[13] T.F. Chan and H.M. Zhou. ENO-wavelet transforms for piecewise smooth functions. *SIAM Journal on Numerical Analysis*, 40(4):1369–1404, 2003.

[14] T.F. Chan and H.M. Zhou. Total variation wavelet thresholding. *Journal of Scientific Computing*, 32(2):315–341, 2007.

[15] Tony F. Chan, Ke Chen, and Xue-Cheng Tai. Nonlinear multilevel schemes for solving the total variation image minimization problem. In *Image processing based on partial differential equations*, Math. Vis., pages 265–288. Springer, Berlin, 2007.

[16] Oddvar Christiansen and Xue-Cheng Tai. Fast implementation of piecewise constant level set methods. In *Image processing based on partial differential equations*, Math. Vis., pages 289–308. Springer, Berlin, 2007.

[17] A. Cohen and I. Daubechies. Non-separable bidimensional wavelet bases. *Rev. Mat. Iberoamericana*, 9(1):51–137, 1993.

[18] I. Daubechies. *Ten lectures on wavelets*. Philadelphia, SIAM, 1992.

[19] L. Demkowicz, J.T. Oden, W. Rachowicz, and O. Hardy. Toward a universal hp adaptive finite element strategy. i: Constrained approximation and data structure. *Comput. Methods Appl. Mech. Engrg*, 77(1-2):79–112, 1989.

[20] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5(1):49–68, 1989.

[21] D.L. Donoho. Interpolating wavelet transforms. *Preprint, Department of Statistics, Stanford University*, 1992.

[22] S. Dubuc. Interpolation through an iterative scheme. *Journal of Mathematical Analysis and Applications*, 114(1):185–204, 1986.

[23] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, and L.Q. Chen. Spectral implementation of an adaptive moving mesh method for phase-field equations. *Journal of Computational Physics*, 220(1):498–510, 2006.

[24] A. Harten. Adaptive multiresolution schemes for shock computations. *Journal of Computational Physics*, 115(2):319–338, 1994.

[25] M. Holmström. Solving hyperbolic PDEs using interpolating wavelets. *SIAM Journal on Scientific Computing*, 21(2):405–420, 1999.

[26] M. Holmström and J. Waldén. Adaptive wavelet methods for hyperbolic PDEs. *Journal of Scientific Computing*, 13(1):19–49, 1998.

[27] W. Huang, Y. Ren, and R.D. Russell. Moving mesh methods based on moving mesh partial differential equations. *Journal of Computational Physics*, 113(2):279–290, 1994.

[28] S. Jain, P. Tsiotras, and H.M. Zhou. A hierarchical multiresolution adaptive mesh refinement for the solution of evolution PDEs. *SIAM Journal on Scientific Computing*, 31(2):1221, 2008.

[29] L. Jameson. A wavelet-optimized, very high order adaptive grid and order numerical method. *SIAM Journal on Scientific Computing*, 19(6):1980–2013, 1998.

[30] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[31] N.K.R. Kevlahan and O.V. Vasilyev. An adaptive wavelet collocation method for fluid-structure interaction at high Reynolds numbers. *SIAM Journal on Scientific Computing*, 26(6):1894–1915, 2005.

[32] H. Li and X.C. Tai. Piecewise constant level set methods for multiphase motion. *International Journal of Numerical Analysis and Modeling*, 4:274–293, 2007.

[33] R. Li, T. Tang, and P. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *Journal of Computational Physics*, 170(2):562–588, 2001.

[34] J. Liandrat and P. Tchamitchian. Resolution of the 1D regularized Burgers equation using a spatial wavelet approximation algorithm and numerical results, ICASE report 90-83, 1990.

[35] J. Lie, M. Lysaker, and X.C. Tai. A binary level set model and some applications to Mumford-Shah image segmentation. *IEEE Transactions on Image Processing*, 15(5):1171–1181, 2006.

[36] J. Lie, M. Lysaker, and X.C. Tai. A variant of the level set method and applications to image segmentation. *Mathematics of computation*, 75(255):1155–1174, 2006.

[37] T. Lu, P. Neittaanmaki, and X.C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Applied Mathematics Letters*, 4(2):25–29, 1991.

[38] T. Lu, P. Neittaanmaki, and X.C. Tai. A parallel splitting up method for partial diferential equations and its application to Navier-Stokes equations. *RAIRO Mathematical Modelling and Numerical Analysis*, 26(6):673–708, 1992.

[39] Z. Luo, L.Y. Tong, J.Z Luo, P. Wei, and M.Y. Wang. Design of piezoelectric actuators using a multiphase level set method of piecewise constants. *Journal of Computational Physics*, 228:2643–2659, 2009.

[40] G.I. Marchuk. Splitting and alternating direction methods, volume I of handbook of numerical analysis (P.G. Ciarlet, J.L. Lions eds), 1990.

[41] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math*, 42(5):577–685, 1989.

[42] J.T. Oden, L. Demkowicz, W. Rachowicz, and T.A. Westermann. Toward a universal hp adaptive finite element strategy. ii: A posteriori error estimation. *Comput. Methods Appl. Mech. Engrg*, 77(1-2):113–180, 1989.

[43] S. Osher and R.P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer, 2003.

[44] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[45] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[46] P. Schröder and W. Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 161–172. ACM New York, NY, USA, 1995.

[47] J. Shen and X.F. Yang. An efficient moving mesh spectral method for the phase-field model of two-phase flows. *Journal of Computational Physics*, 228(8):2978–2992, 2009.

[48] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, pages 506–517, 1968.

[49] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186–200, 1996.

[50] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, 1998.

[51] W. Sweldens and P. Schroder. Building your own wavelets at home. *Lecture Notes in Earth Sciences*, pages 72–130, 2000.

[52] X.C. Tai, O. Christiansen, P. Lin, and I. Skjælaaen. Image segmentation using some piecewise constant level set methods with MBO type of projection. *International Journal of Computer Vision*, 73(1):61–76, 2007.

[53] O.V. Vasilyev and C. Bowman. Second-generation wavelet collocation method for the solution of partial differential equations. *Journal of Computational Physics*, 165(2):660–693, 2000.

[54] O.V. Vasilyev and S. Paolucci. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *Journal of Computational Physics*, 125(2):498–512, 1996.

[55] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[56] J. Weickert, B. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, 1998.

[57] J.C. Xu and W.C. Shann. Galerkin-wavelet methods for two-point boundary value problems. *Numerische Mathematik*, 63(1):123–144, 1992.