Summer 2010

Multiresolution analysis for adaptive refinement of multiphase flow computations

Neal Phillip Grieb
University of Iowa

Recommended Citation
Grieb, Neal Phillip. "Multiresolution analysis for adaptive refinement of multiphase flow computations." MS (Master of Science) thesis, University of Iowa, 2010.
https://doi.org/10.17077/etd.fjqe5lib

Follow this and additional works at: https://ir.uiowa.edu/etd
Part of the Mechanical Engineering Commons

MULTIRESOLUTION ANALYSIS FOR ADAPTIVE REFINEMENT OF MULTIPHASE FLOW COMPUTATIONS

by

Neal Phillip Grieb

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

July 2010

Thesis Supervisor: Professor H.S. Udaykumar

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Neal Phillip Grieb

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Mechanical Engineering at the July 2010 graduation.

Thesis Committee:

H.S. Udaykumar, Thesis Supervisor

James Buchholz

Christoph Beckermann

Albert Ratner

ABSTRACT

Flows around immersed boundaries exhibit many complex, well defined and active dynamical structures. In fact, features such as shock waves, strong vorticity concentrations in shear layers, wakes, or boundary layer regions are critical elements in representing the dynamics of a flow field. In order to capture the correct kinematic and dynamic quantities associated with the fluid flows, one must be able to efficiently refine the computational mesh around areas containing high gradients of pressure, density, velocity, or other suitable flowfield variables that characterize distinct structures. Although there are techniques which utilize simple gradient-based Local Mesh Refinement (LMR) to adapt resolution selectively to capture structures in the flow, such methods lack the ability to refine structures based on the relative strengths and scales of structures that are presented in the flow. The inability to adequately define the strength and scale of structures typically results in the mesh being over-refined in regions of little consequence to the physical definition of the problem, under-refined in certain regions resulting in the loss of important features, or even the emergence of false features due to perturbations in the flowfield caused by unnecessary mesh refinement. On the other hand, significant user judgment is required to develop a “good enough” mesh for a given flow problem, so that important structures in the flowfield can be resolved. In order to overcome this problem, multiresolution techniques based on the wavelet transform are explored for feature identification and refinement. Properties and current uses of these functional transforms in fluid flow computations will be briefly discussed. A Multiresolution Transform (MRT) scheme is chosen for identifying coherent structures because of its ability to capture the scale and relative intensity of a structure, and its easy application on non-uniform meshes.

The procedure used for implementation of the MRT on an octree/quadtree LMR mesh is discussed in detail, and techniques used for the identification and capture of jump discontinuities and scale information are also presented. High speed compressible flow simulations are presented for a number of cases using the described MRT LMR scheme. MRT based mesh refinement performance is analyzed and further suggestions are made for refinement parameters based on resulting refinement.

The key contribution of this thesis is the identification of methods that lead to a robust, general (i.e. not requiring user-defined parameters) methodology to identify structures in compressible flows (shocks, slip lines, vertical patterns) and to direct refinement to adequately refine these structures. The ENO-MRT LMR scheme is shown to be a robust, automatic, and relatively inexpensive way of gaining accurate refinement of the major features contained in the flowfield.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The interaction of shock waves with embedded objects is important in several applications such as high speed flows with droplets, bubbles and particles, high velocity impact and penetration, and detonation diffraction. The results of such shock-interface interaction processes are total or partial reflection, refraction, and transmission of shock waves. These interactions lead to complex, unsteady, evolving flow patterns that represent highly localized regions of high “activity” in the flow region.

Similarly in lower speed incompressible flows, interaction of objects in fluid environments, such as air foils, car bodies, buildings, and other structures is important to automotive and aeronautical research and development, city development, offshore oil production, wind turbine research and development, and numerous other everyday engineering applications. In these lower speed flows, objects also introduce small scale phenomena such as boundary layers, vortex shedding, and turbulent wake patterns into the flowfield.

In both compressible and incompressible flows, the key issue for computational techniques is to produce accurate predictive results; this hinges partly on the ability of the computational grid to adequately resolve the important structures in the flowfield. *This is the central problem addressed by this thesis, i.e. how to devise a robust, efficient and automatic technique to identify important structures in a flowfield and to direct computational resources (or improved resolution) to such regions.*

Typically, in complex flows, features can arise at different scales, ranging from discontinuities such as shocks, thin boundary layers and vortical patterns of substantial thickness (typically of the same length scale as the embedded object). Therefore, a scheme that seeks to identify and capture structures must be able to distinguish the difference in scale as well as the strength of the structure and to direct the appropriate amount of refinement to each structure. In addition, it should be able to determine regions in the flowfield which are of little of dynamic consequence and indicate that such inactive regions be treated with rather coarse refinement. These desired features of a refinement triggering scheme apply in particular to Cartesian grid methods, where the base grid typically assumes a regular rectangular (in 2D) or cuboidal (in 3D) topology. In this setting, refinement can be all the more tricky since the base grid can be arbitrarily oriented with respect to both embedded geometries as well as the flow features that need to be resolved.

While the Cartesian grid based approach significantly alleviates the complexity associated with mesh management (which is a key issue for boundary fitted moving mesh methods for instance), this method lacks flexibility in efficient and automatic grid point placement in regions of the flowfield with rich and active structures. Typically, the regions with active structures are spatially localized in a narrow regime occupied by features such as shock waves, rarefaction waves, material interfaces, boundary layers, and vorticity concentrations. Moreover, the mesh resolution required to resolve these structures is dictated by the thickness and strength of the dominant structure (such as the width of the primary shock wave, thickness of the boundary layer etc.). Employing a uniform mesh (as dictated by the width of the dominant structure) is not pragmatic as it would result in significant computational overhead. Alternatively, one could supplement the Cartesian grid approach with automatic mesh adaptation techniques. Quadtree- (in 2D) and Octree- (in 3D) based hierarchical mesh structures are commonly employed in a Cartesian grid setting to effect local solution adaptive mesh refinement, as pictured in Figure 1.1. The key idea is to start with a rather coarse base mesh, let the flow solution evolve, examine the solution for fine-scale structures, and progressively refine the mesh to follow the fine structures. In this setting, the challenge is to trigger refinement in an optimal way. The basic requirements for developing good refinement triggers can be listed as follows:

1. The refinement criterion should be sensitive enough that the important structures are resolved, but not so sensitive that small amplitude variations (especially numerical noise) can trigger refinement.
2. The refinement trigger should be problem-independent and robust and should be automated without user-defined switches.
3. The refinement criteria should set the gradation of the mesh in such as way that structures at various scales should be resolved at their respective scales.

In the following, we seek to develop such a refinement strategy within the context of octree-based local mesh refinement.

1.1 Current Refinement Methods

In [1] an octree based Local Mesh Refinement (LMR) technique was developed for the simulation of compressible multiphase flow problems. In contrast to an AMR approach [2], the LMR technique identifies cells occupied by active structures and dominant gradients, and recursively divides the cells until the desired level of refinement is achieved. The gradient-based refinement criterion works as follows:

1. Gradients of the density, entropy, and pressure are calculated using centered and off-centered (forward and backward) differences. In areas

containing smooth features, the centered and off-centered differences should nearly be equal.

2. The user defines gradient error bounds (between centered and off-centered difference values), δ_1 and δ_2, as in [3], in order to restrict the refinement in the domain.
3. If the gradient error of the current cell falls within the bounds it is flagged for refinement, and is refined to the highest level.

However, a significant drawback to this approach is that the preset user-defined tolerance limits required, have a large range of possible “good” values [3], and all flow features meeting this tolerance limit are immediately refined to the finest level. This leads to trial-and-error searches for the proper criteria for refinement for a given problem. If an optimal error bound is not specified, then the mesh employed for the calculation may result in over-refinement (i.e. inefficient computation) or under-refinement (i.e. reduced accuracy and loss of features).

Figure 1. 1 depicts a simulation of a bullet traveling through air at Mach 1.22. The top portion depicts the mesh used to capture these flow features, and the bottom portion contains a numerical Schlieren image overlaid on the mesh. In this figure the primary shock systems, such as the incident and the reflected shock systems, along with the features such as the Prandtl-Myer expansion wave, slip-lines, and vortices have been refined to the finest level (four levels of refinement were used in the calculations). Although the mesh refinement is confined to regions with rich structures, weaker features such as the expansion wave and slip lines need not be refined to the finest level (areas highlighted in the top mesh portion). The resulting mesh topology may be acceptable in two-dimensions, but may result in significant overhead in computational resources in

three-dimensions. Thus to maintain an optimal computational effort, one could grade the levels of mesh refinement by maintaining dominant structures such as the primary shock systems at the finest level, and weaker features at a coarser level.

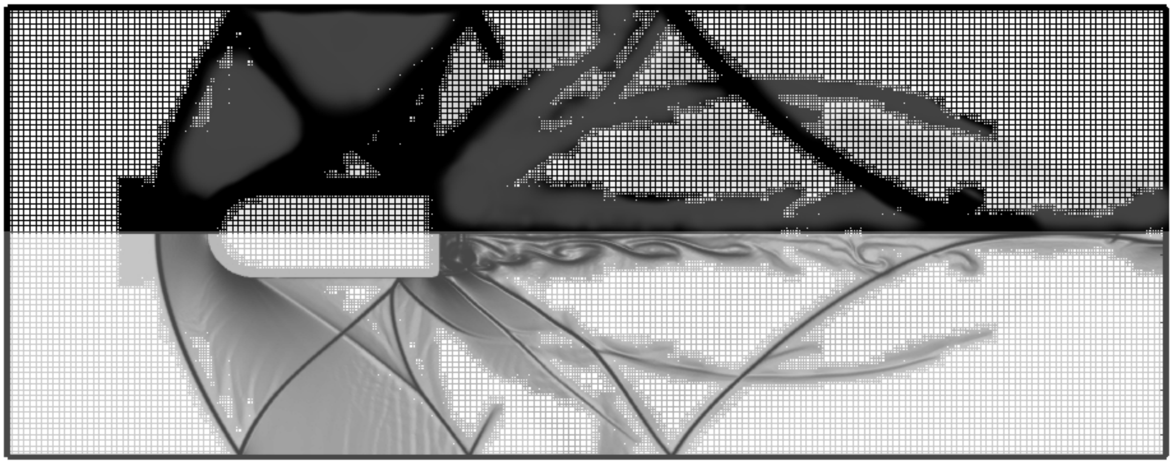


Figure 1. 1 Excessive Mesh Refinement Using the Current Gradient Based Refinement Scheme.

1.2 Objective/Proposal

It is clear that the smoothness of the flow field solution should dictate where and how much refinement is necessary. Thus, it would be advantageous for the gradient-based refinement approach to be replaced by another means of finding a global or local measure of the smoothness of the flowfield solution.

One scheme for determining the smoothness of a function is through a functional approximation method. Functional approximation methods involve an approximation of the function of interest through decomposition into a series of “well behaved” functions (for example Fourier components) such that a given function *f(x)*, may be represented as:

$$f(x) \approx \sum_{i=1}^{\infty} a_i \varphi(x)_i \qquad \text{Eqn 1. 1}$$

Using Fourier decomposition for example, it is possible to decompose a spatially/temporally smooth function into its contribution from various scales (in the wavenumber or frequency domain). However, as explained later, Fourier transforms lead to global descriptions, while what we are seeking is local information on scales and structure intensities. Thus, local functional multi-scale representations are necessary, which is a facility provided by wavelet transforms.

In the next few chapters a functional approximation approach for the purposes of locating the gradients and scale of the features in the flow field will be developed. An overview of the Fourier, wavelet, and multiresolution transforms are presented, including basic properties and definitions, implementation, common uses in engineering, and the practicality of use with the current objective. A cell-based Multiresolution Transform (MRT) scheme, as defined by Amat et al. in [4], is adopted in order to eliminate unnecessary refinement of the computational grid. Multiresolution analysis of the flow field is superior to the use of gradients because both the strength and size of the structure can be captured, thus allowing for a relatively weighted refinement of the mesh.

The use of the MRT for refinement of flowfield structures contained within an unstructured computational mesh has been implemented in a few works including Chiavassa et al. [5], Cohen et al. [6], and Bihari et al. [7]. The majority of publications containing 2D MRT use for mesh refinement however do not provide a detailed set of instructions for the creation and implementation of the MRT in the unstructured grid framework. Most current publications also do not provide any quantitative analysis of the obtained detail coefficients or method for developing a robust detail coefficient truncation parameter. In fact in [5] and [7] a seemingly arbitrary static thresholding parameter is used to truncate the detail coefficients before the remaining detail coefficients are used to flag cells for refinement. The authors of [6] mention their previous works in which a robust truncation method is developed, but do not provide any further information regarding the method used for determining the threshold used.

It is therefore the goal of this thesis to fully explain the strategies employed in the creation of an operational cell-based MRT scheme that includes both a detailed analysis of the resulting detail coefficients and a number of possible truncation schemes to ensure robust feature detection for both compressible and incompressible flow computations. The implementation of an Essentially Nonoscillatory Multiresolution Transform (ENO-MRT) refinement method for a fixed octree/quadtree Cartesian grid will be discussed in depth and compared to current gradient based refinement techniques. Suggestions for associated transform parameters will be made, and possible future works in this area will be discussed.

CHAPTER 2

FUNCTIONAL APPROXIMATION

Functional approximation methods can be very efficient and powerful methods for finding general information about a function $f(x)$. Through a simple rearrangement of the general equation given in Eqn 1.1, the functional approximation method can be used to express functions in a chosen basis set $\varphi(x)$. Use of functional approximation methods in such a manner is then termed a functional transform. Functional transformation methods give insight into the behavior of a function because they represent how well the function $f(x)$ can be mapped to the known behavior of the basis functions φ. A custom selection of the basis function, φ, can be made in order to gain different information about the function to be approximated. Functional approximation methods also enable a method of compression or filtration through truncation of terms with coefficients of “little consequence” to the overall solution. This procedure is termed truncation or denoising.

All of these properties could be very useful in not only finding the areas in the flowfield of high gradient, but also in limiting the amount of noise introduced into the mesh refinement. The next three subsections of this chapter step through the natural progression of functional approximation techniques before the suitable choice of basis functions is reached. A brief synopsis of the discrete Fourier transform and discrete wavelet transform will be made in order to show the properties of their basis functions. The multiresolution transform will then be introduced and the reasons for its use with the current problem set are described in detail.

2.1 The Fourier Transform

The Fourier transform is a common and extremely powerful functional approximation method. It was originally used by Fourier to solve the heat conduction PDE, but since has been used in many other areas of science and engineering.

A Fourier transform is a functional transform which transforms a function from physical (temporal) space to the wavenumber (frequency) domain. The Fourier transform allows for the major frequency components of a function to be determined, which is meaningful in areas of solid body dynamics and vibrations, signal processing, optics, and solutions to the heat and wave equations.

2.1.1 Definition

A Fourier transform of a function g(x) is defined in [8] as:

$$G(\omega) = \int_{-\infty}^{\infty} g(x) e^{-ix\omega} dx, \text{where } \omega = 2\pi n \qquad \text{Eqn 2. 1}$$

The above definition is of course for the case of a continuous function (analog signal). It is clear that for a finite set of points, obtained in numerical computations, time-sampling of data, or digital time-series recordings the Discrete Fourier Transform (DFT) is needed. The mathematical definition of the DFT is shown in Eqn 2. 2.

$$F(\omega) = \sum_{n=0}^{N-1} f(n) e^{\frac{-i2\pi kn}{N}}, k = 1,2,\ldots N-1 \qquad \text{Eqn 2. 2}$$

This discrete time representation will be the one of primary interest to us as the examples given, as well as most engineering applications use this method.

2.1.2 Properties

The discrete Fourier transform has many defining properties. The two most important to it, as for most other functional approximation methods, are the completeness and orthogonality of the basis functions. Completeness of the basis functions implies that the space function of interest, *f(x)*, is completely transformed, or in other words, all the information contained in *f(x)* is transformed. Completeness, therefore, also implies that the transform may be inverted without any loss of data. Orthogonality of the basis functions used in functional approximation methods is, for the most part, a necessity because it allows for the data contained in the original function to be represented in the most compact way possible (i.e. there is no redundant information). The orthogonality of the Fourier transforms basis functions allows each expansion coefficient $F(\omega)$ (or $a_{n,}$ in Eqn. 1. 1), to be calculated by inner product. Some other major mathematical properties associated with the Fourier transform include linearity, convolution, periodicity, globality, and inversion, and are summed up in Table 2. 1.

Table 2. 1 Important Properties of the Fourier Transform.

Property	**Mathematical Description**	
Orthogonality	$\int_{-\infty}^{\infty} e^{-inx} e^{-imx} = \begin{Bmatrix} m = n, C \\ m \neq n,\ 0 \end{Bmatrix}$	Eqn 2. 3
Linearity	$g(x) + f(x) \Leftrightarrow G(\omega) + F(\omega)$	Eqn 2. 4
Convolution	$f(x) * g(x) = F(\omega) G(\omega)$	Eqn 2. 5
Global	$\lvert a_n \rvert \propto \frac{1}{f^{p}}$ (f^{p} global)	Eqn 2. 6
Periodicity	*f(x)* is assumed periodic $\forall \in x$	Eqn 2. 7
Inversion	$g(x) = \int_{-\infty}^{\infty} G(\omega) e^{ix\omega} d\omega$	Eqn 2. 8

The properties of the DFT are the same as those of the continuous Fourier transform, however, since the number of terms that represent the signal have decreased to N terms the approximation energy is given by Parseval equality as:

$$\sum_{m=1}^{N} \left| f_n \right|^2 = \frac{1}{N} \sum_{n=-N/1}^{N/2-1} \left| F_n \right|^2 \qquad \text{Eqn 2. 9}$$

2.1.3 Common Uses in Engineering

The ability of the Fourier transform to obtain useful information about fluctuating phenomena has led to many uses in science and engineering. A few of the most common uses for the Fourier analysis include obtaining solutions to the heat and wave equations, denoising of audio and electrical signals, clarification of images which utilize numerous sound and light waves, etc. For a more complete discussion of the history and some of the uses of the Fourier transform refer to [9].

2.1.4 Uses in CFD

The Fourier transform is used in CFD for the study of turbulent flows, through a transformation of the Navier-Stokes equations into Fourier space. The class of spectral methods employs this strategy. The Fourier transform allows for the selection of a range of vortical elements with frequency, f, to be easily filtered for analysis. The computational costs of solving the Fourier Navier-Stokes equations are large because the nonlinear advection term in the Navier-Stokes equation cannot be directly transformed due to the linearity property of the Fourier transform (refer to Eqn 2. 4). In order to circumnavigate this problem the nonlinear term is solved in regular space and the DFT of the resulting numerical data is used to complete the equation. Solving the nonlinear term in regular space requires constant translation between Fourier space and regular space and this adds considerably to the computational costs of the method. On top of the already computationally hefty calculation if a study of higher frequency vorticity is desired one must also reduce the grid size (everywhere) until that frequency can be resolved. Another issue with employing Fourier basis functions is the global nature of Fourier modes, which places restrictions on the types of boundary conditions that can be applied. Typically periodic domains or homogeneous boundary conditions are desired. Refer to [8] for a more in depth description of spectral CFD codes.

2.1.5 Examples

For further understanding of the Fourier transform as it applies to signal processing, and in order to point out some of its pitfalls, simple one dimensional transforms will be preformed and errors or problems associated with approximations will be presented and discussed.

If *f(x)* is the function of interest and is defined by:

$$f(x) = \left\{ \begin{matrix} 2sin(6x) + 0.5sin(60x), \ 0 \leq x < \pi \\ 2sin(6x), \ x \geq \pi \end{matrix} \right\} \qquad \text{Eqn 2. 10}$$

The function is graphed below in Figure 2. 1(a). The DFT representation of this function is shown below in Figure 2. 1(b). Notice that the two major frequencies of the function, 6 Hz and 60 Hz, have been captured by the Fourier transform, and that the relative energy or amplitude of the lower frequency waveform has been correctly captured. It is important to note that from the DFT we cannot distinguish that the higher frequency waveform does not continue throughout the entire waveform, but only that it exists, and that it has a relative strength of roughly 0.25 (which is in fact incorrect). These discrepancies are due to the periodicity assumed by the Fourier transform, the fact that only half the waveform contains higher frequencies explains why only half of its signal strength was captured. The higher frequency can also be filtered out of the function by simply setting components in the band of 40-80 Hz to zero (also known as a band stop filter), thus allowing for filtering of noise and/or compression.

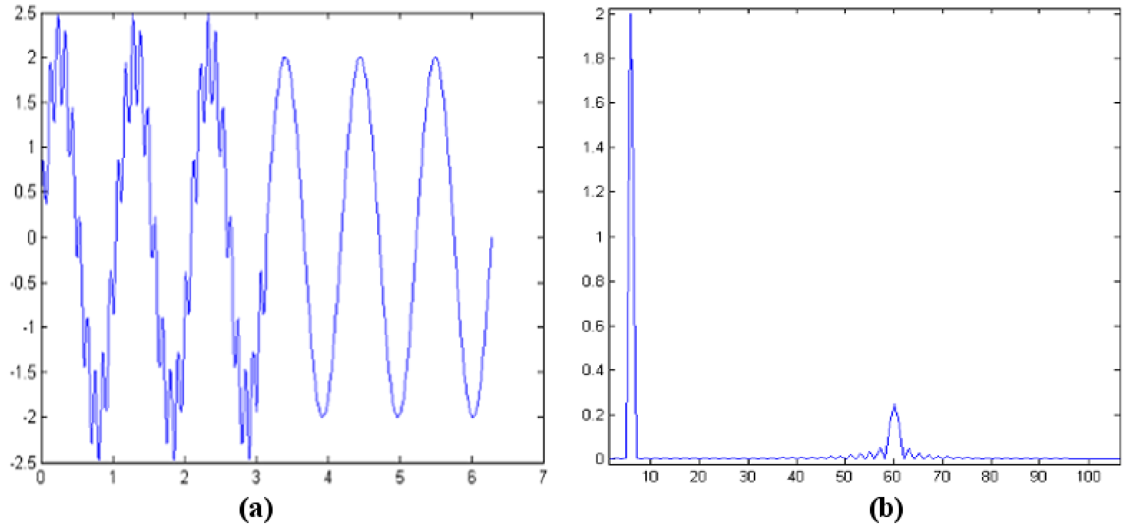
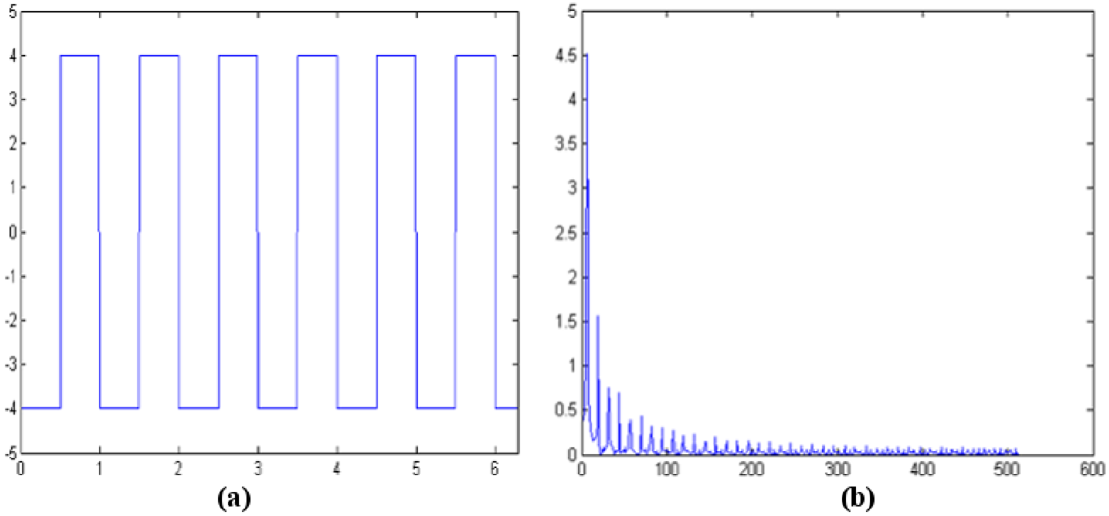


Figure 2. 1 Fourier Transform of a Multi Frequency Sine Function: (a) Function described in Eqn 2. 10, (b) Fourier transform $F(\omega)$.

Another problem associated with the Fourier transform is that jump discontinuities in the function being transformed cannot be approximated by the globally smooth basis functions that are intrinsic to DFTs, even when the number of terms in the series approaches infinity. Gibbs-like oscillations occur around the points of discontinuity and appear as multiple high frequencies in the DFT as can be seen in Figure 2. 2. In order to proscribe error only in the vicinity of the discontinuity a large number of terms would need to be used for the approximation, which is computationally undesirable. Therefore, the inability of the conventional Fourier transforms to correctly represent local behavior renders it unsuitable to indentifying local structures and ascribing appropriate scales to the local structures.



(a) **(b)**

Figure 2. 2 The DFT of a Square Wave: (a) Function *f(x)*, (b) DFT of *f(x)*.

2.1.6 Application to the Current Objective

As can be seen from Figure 2. 1 and Figure 2. 2 the reason that the Fourier transform cannot be applied to the task at hand (which may involve highly localized structures and discontinuities such as shocks) is due to its use of global and smooth basis functions. The fact that the basis functions are global means that the mapped transform of a function, *f(x)*, contains no (useable) data concerning the location of the frequencies and amplitudes contained in *f(x)*. The smoothness of the basis functions inhibits the Fourier transform from properly representing jump conditions as well. Because of these characteristics, the Fourier transform, although important in the derivation of the wavelet transform and multiresolution transform, cannot be directly applied for our intended use. While windowed Fourier transforms [10] partially alleviate the problems associated with global basis functions, there are problems associated with determining the location and size of the windows. Fortunately, there are stronger alternatives, as discussed below.

2.2 The Wavelet Transform

A wavelet transform maps a desired function from real space onto wavelet space. Wavelet transforms stem from the mathematical foundation of Fourier transforms; however, a wavelet transform is able to represent both the scale and space/time relationship of a function. This is done by selecting a pair of compactly supported basis functions (i.e. the basis functions do not contain any significant value outside of a bounded window), a wavelet function and its orthogonal scaling function, which are then translated and dilated across the function in order to preserve the real spatial mapping. Translation of the finite-bound wavelet allows for the description of functions of finite extent on the real axis, while dilation allows for multi-scale representation. These attractive features of wavelets are described below.

2.2.1 Definition

The Continuous Wavelet Transform (CWT) is defined as [11]:

$$w_{\psi}[f](a,b) = \frac{1}{\sqrt{a}} \int_{x \in R} f(x) \overline{\psi\left(\frac{x-b}{a}\right)} dx \qquad \text{Eqn 2. 11}$$

The parameter a (>0) is a scaling parameter, while b ($b \in \Re$) is a translation parameter of the wavelet function ψ ($\psi \in L_2(\Re)$). The wavelet function has compact support (of bound a) about the point x=b. Since the basis function no longer spans all elements of interest (due to its compact support) it must be translated in order to cover the span of the function to be represented. Perhaps the simplest example of a wavelet is the hat function also known as the Haar wavelet (shown below).

If the wavelet is admissible, i.e. if it preserves the signals energy, the inverse transform may be computed as shown in [11]:

$$s(x) = \frac{1}{C_\psi} \int_{\in \Re} w_\psi [s](a,b)\psi\left(\frac{x-b}{a}\right) db \frac{da}{a^2\sqrt{a}} \qquad \text{Eqn 2. 12}$$

In the discrete case the range of the parameters a and b must be truncated to a finite number of terms, and therefore not all of the signal energy can be accounted for by the wavelet function alone. The Discrete Wavelet Transform (DWT) is therefore coupled with another basis function (usually one that is orthonormal to ψ), known as the scaling function φ. The DWT from the function space to the wavelet space is therefore performed by two summation operations that take place for the wavelet and scaling functions. Subsets of the data are transformed through translation and dilation of the wavelet and scaling coefficients. This can be expressed, as shown in [11, 12], by:

$$f(x) = \sum_{k=0}^{2^{j_0}} c_{j_0}^k \phi_{j_0}^k + \sum_{l=j_0}^{j-1} \sum_{k=0}^{2^l-1} d_l^k \psi_l^k \qquad \text{Eqn 2. 13}$$

Further discretization of the scaling and wavelet functions φ and ψ, through use of the z-transform, results in the discrete time DWT's wavelet and scaling filter coefficients: α and β. The set of discrete filter coefficients, α and β, represent a more compact version of the orthogonal basis pair φ and ψ. (The filter length is limited by the number of vanishing moments required, as explained below). The process can then be further simplified by replacing the functions φ and ψ with α and β and performing two summation operations. For more detail on the derivation of the filter coefficients refer to Chapter 7 of [13] or [11].

When performing the transform, the scaling filter coefficients α_j are multiplied by the current dyadic subset of the function values (i.e. either the even or odd numbered discrete sample points of the function being transformed are used for translation of the basis functions), leading to an up-scaled and down-sampled representation of the function, *f(x)*, usually denoted as, *s*. and then stored in a the following manner:

$$s\left(x_{2j}^{k+1}\right)=\sum_{j=1}^{N}\alpha_j f\left(x_{2j+1}^{k}\right) \qquad \text{Eqn 2. 14}$$

The wavelet filter coefficients β_i are utilized in the same manner and the local odd function values are replaced with the so called wavelet or detail coefficients often labeled, *d*.

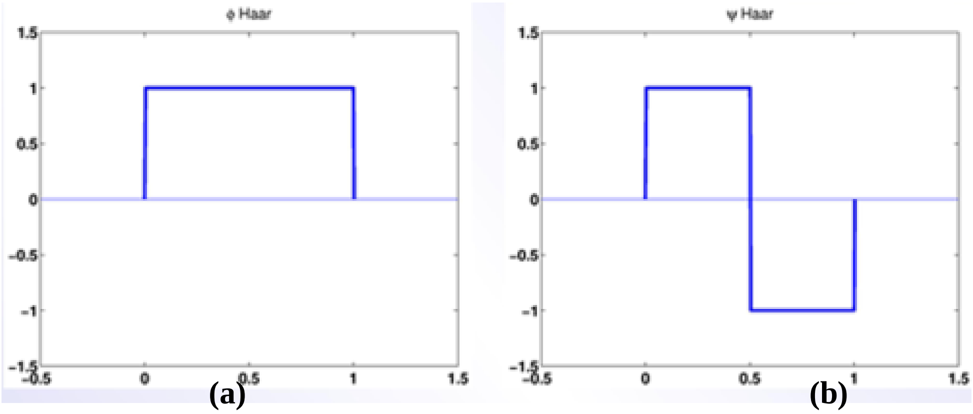
$$d\left(x_{2j+1}^{k+1}\right)=\sum_{j=1}^{N}\beta_j f\left(x_{2j+1}^{k}\right) \qquad \text{Eqn 2. 15}$$

In general, when performing the transform the coefficients are reordered such that elements of *s* and *d* are separated into two arrays of half the length of the original function. As defined in Eqn 2. 13 the process can be repeated again, this time using *s* as the function that is to be transformed. Use of wavelet transforms in order to decompose a function more than once is known as Multiresolution Analysis (MRA), and is of much interest because it allows for the discovery of multiple scales at each point of the function.

In the case of the Haar wavelet the wavelet and scaling filter coefficients, denoted as α_i and β_i respectively, can be found to be:

$$\alpha_1 = \frac{1}{\sqrt{2}}, \alpha_2 = \frac{1}{\sqrt{2}} \qquad \text{Eqn 2. 16}$$

$$\beta_1 = \frac{-1}{\sqrt{2}}, \beta_2 = \frac{1}{\sqrt{2}} \qquad \text{Eqn 2. 17}$$



(a) (b)

Figure 2. 3 The Haar Wavelet: (a) Scaling function, (b) Wavelet function (courtesy of www.cnx.org, Kileen Cheng).

A graphical representation of the Haar wavelet is shown in Figure 2. 3.

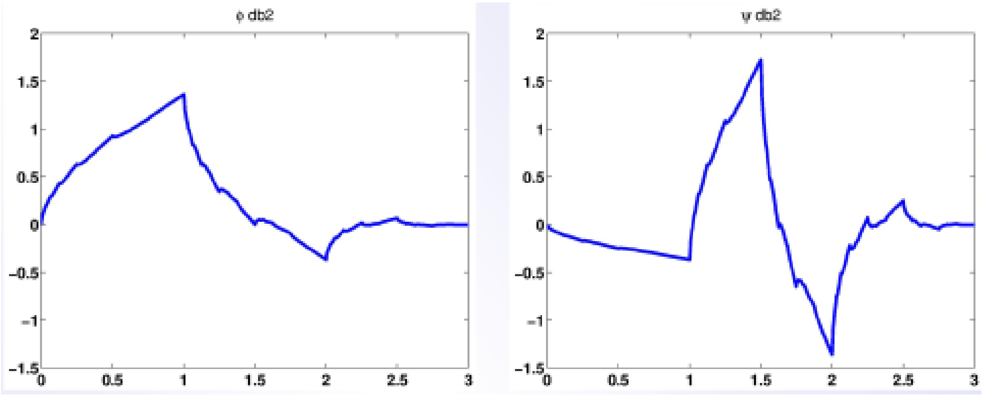
In the case of a Daubechies 4 transform the wavelet filter coefficients α_i are as follows:

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \qquad \text{Eqn 2. 18}$$

and the scaling filter coefficients β_i are:

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \beta_2 = \frac{-3-\sqrt{3}}{4\sqrt{2}}, \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \beta_4 = \frac{-1+\sqrt{3}}{4\sqrt{2}}, \qquad \text{Eqn 2. 19}$$

A graphical representation of the Daub 4 wavelet is shown in Figure 2. 4.



(a) **(b)**

Figure 2. 4 The Daubechies 4 Wavelet: (a) Scaling function, (b) Wavelet function (courtesy of www.cnx.org, Kileen Cheng).

The discrete wavelet transform is very simple to employ. The filter coefficients simply need to be multiplied as shown above with the corresponding function sample points and then shifted along the function. In the case of orthonormal wavelets, the shift occurs without overlap. (In this section the two orthonormal wavelets described above will be presented for simplicity, however, it should be noted that these properties are not a necessity for wavelet transforms. In fact there are entire classes of biorthogonal wavelets, such as the Daubechies 9/7 wavelet, which require that one of the filters be shifted more times than the other, because orthogonality is obtained through the overlapping of the scaling and wavelet filters.) After each full pass of the filter coefficients the function, *s*, is again down-sampled, which creates a dilate of the wavelet and scaling function, and the process is again repeated.

2.2.2 Properties

There are several mathematical constraints which are required in order to properly define a wavelet, as well as several properties which may be highly desired. For instance, the Haar and Daubechies wavelets are both orthonormal wavelets. This implies that that the scaling and wavelet filters (and functions) of the Haar and Daubechies 4 wavelets do not change the energy of the signal as it undergoes transformation,

$$\alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 = 1, \text{ and } \beta_1^2 + \beta_2^2 + \beta_3^3 + \beta_4^2 = 1 \qquad \text{Eqns 2. 20}$$

As is the case of the Fourier transform, the wavelet function filter coefficients are orthogonal to the scaling filter coefficients, thus [14]:

$$\sum_{i=1}^{N} \alpha_i \beta_i = 0 \qquad \text{Eqn 2. 21}$$

However, unlike the Fourier transform, the wavelet transform is not global. A local change of the function to be approximated will only locally change the wavelet representation obtained. Compact support allows us to retain useable information about the location of the elements (due to the translation of wavelets) at a hierarchy of scales, with rapid local changes recorded in the wavelet contributions at finer scales (due to the dilation property of the wavelets). The wavelet transform is therefore fairly well suited to the current requirements of identifying localized structures that may be present at various scales and to providing multi-scale representation of the fields.

Another property of the wavelet transform that may be prescribed is the choice of a vanishing moment. Vanishing moments may be described as follows:

$$\int_{-\infty}^{\infty} x^{p}\psi_{j}^{k}(x)dx = 0 \qquad \text{Eqn 2. 22}$$

where, p is the number of poles of the wavelet and thus $0 \leq \mathrm{p} < \mathrm{m}$. The number of vanishing moments contained in a wavelet allows the detail coefficients to be resistant to a certain underlying function [11]. In other words a wavelet containing one vanishing moment (the Haar transform) corresponds to a resistivity in constant changes, while a wavelet with two vanishing moments shows no change in detail coefficient magnitude when there exists an underlying function which is linearly increasing, and so on.

Most functional approximation methods have the ability to compress data through some type of filtering technique. In the case of the Fourier transform one could eliminate noise or higher frequencies in a signal through use of a band pass or band stop filter. The same concept is valid for wavelet transforms through the thresholding of detail coefficients at different scales. “Small” detail coefficients show that the function in the neighborhood of the filter is well represented by the choice of wavelet basis. “Small” detail coefficients can therefore be set to zero which will cause the reconstruction of the actual function to be very slightly changed.

2.2.3 Common Uses in Engineering

The wavelet transform has many uses in engineering; however, its main stronghold has been that of signal and image processing [11, 13, 14]. The ability of the wavelet to breakdown the components of an image or signal and efficiently denoise and enhance them has made wavelets invaluable to signal and image processing [13, 14, 15, 16].

The wavelet can be used to enhance or breakdown signals and images in multiple ways. If one desires the edges of an image to be enhanced, the higher scale detail coefficients can simply be multiplied by a factor greater than one before the inverse transform is taken [14]. The same technique can be used to enhance weak signals as well. Signals and images can also be filtered or highly compressed, such as in the JPEG 2000 compressed picture format, through the truncation of the detail coefficients at certain scales [14]. Investigation of the detail patterns obtained on multiple scales is also frequently performed in signal and image recognition software.

2.2.4 Uses in CFD

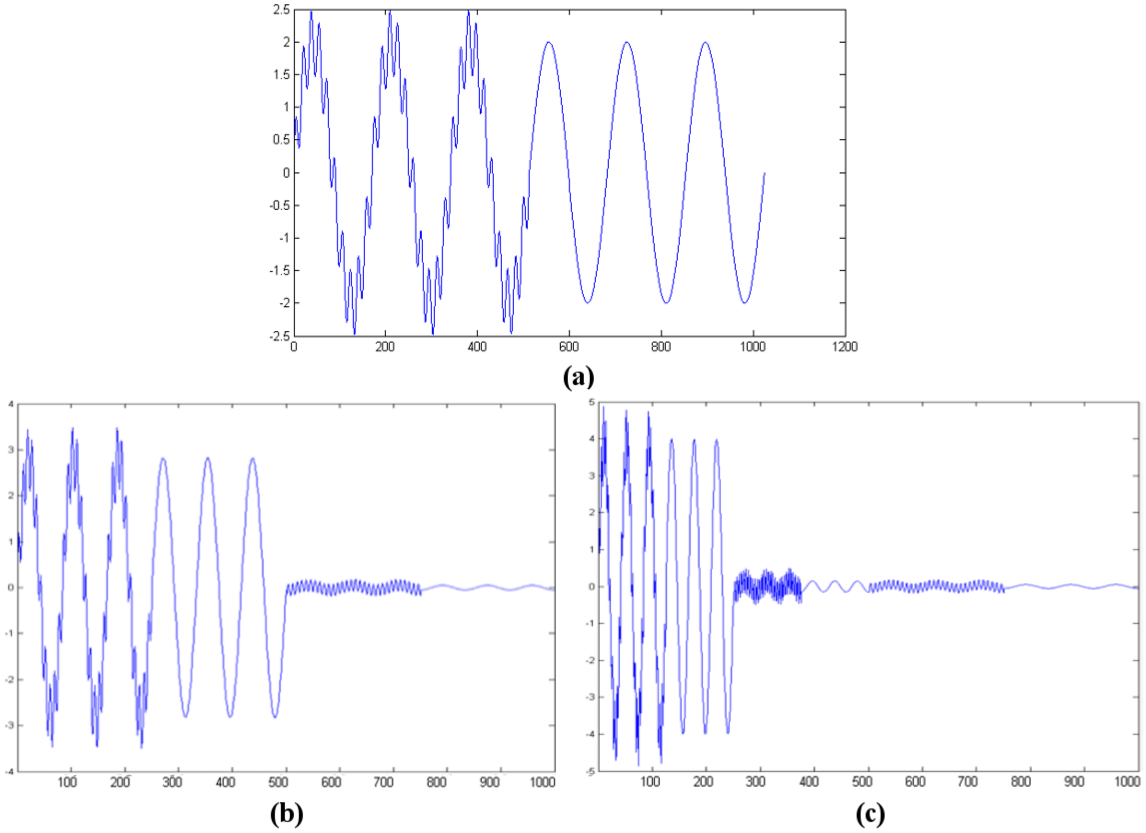
Wavelet transforms like Fourier transforms have their uses in CFD. As with the Fourier transform the Navier-Stokes equations can be converted into wavelet space equations, and then solved as in [12]. There are some serious computational drawbacks to solving the wavelet space Navier-Stokes equations, since, just as with the Fourier transform Navier-Stokes equations, the wavelet space of the nonlinear term must be specially treated, as described in [6]. New methods are continually being developed in order to attain better computational proficiency when solving the wavelet-space Navier-Stokes equations.

Another major drawback of many wavelet based methods in CFD is that orthogonal wavelets require evenly spaced data point placement in order to satisfy the defined vanishing moment condition. There have been methods devised for ensuring the occurrence of vanishing moments on non-equidistant data sets through a creation of wavelet filter coefficients based on the particular spacing given at any point in the decimation. Dynamic creation of orthonormal wavelet filters requires that one both calculate the filter coefficients and somehow store the coefficients used, which may prove expensive.

Due to the decimation style wavelets are also limited to use on square grids containing a multiple of 2^n points in each direction. Methods such as padding or wrapping about the domain points can easily solve the issue of having the correct size, but the domain would still need to be nearly square in shape in order to avoid boundary effects caused by excessive over padding. The constraint of a near square domain may result in unnecessary calculations being performed in regions of little to no consequence to the embedded objects.

2.2.5 Examples

To illustrate the functionality of the wavelet transform and to highlight its advantages over the Fourier transform, let us reexamine the function given above in Eqn 2. 10. The Haar wavelet transform is shown in Figure 2. 5.



(a)

(b) **(c)**

Figure 2. 5 The Haar Transform of the Function Given by Eqn 2. 10: (a) Original function (b) First level Haar transform (c) Second level Haar transform.

The above wavelet decomposition is shown in an index value format after the separation of the scaled signal, *s*, and the wavelet coefficients, *d*. The detail coefficient array (indices 513-1024) show the filtered small scale (higher frequency) oscillations that occur at the current resolution. The down-sampled function (indices 1-512), is then used recursively to obtain the second level transform shown in (c). The effects of the vanishing moment can be seen in the underlying low frequency obtained in the detail coefficients. If the Daubechies 4 wavelet is instead used, the detail coefficients display much less of an underlying frequency component because the vanishing moments are transparent to underlying functions of a quadratic nature rather than the support mean value.

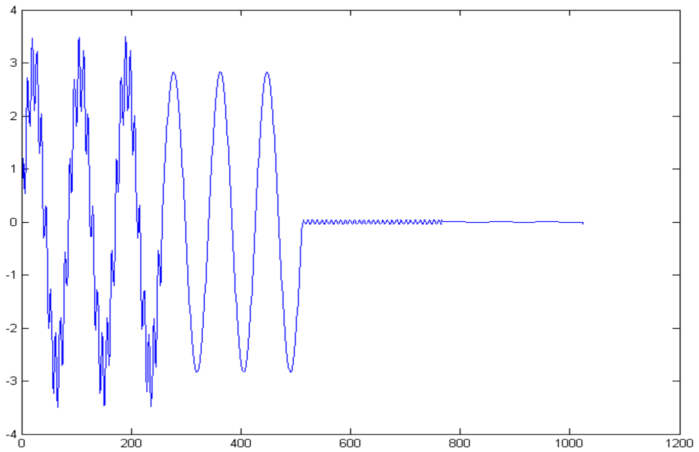


Figure 2. 6 The 1st Level Daub 4 Transform of the function given by Eqn 2. 10.

Figure 2. 7 illustrates the DWT of a square wave function for both the Haar and Daubechies 4 wavelets. Notice that because of the shape of the Haar basis (along with its extreme compactness: 1 vanishing moment) there exist no errors associated with some of the jump conditions (because they lie exactly on a translation point). The Daub4 wavelet has a wider support and a basis function which does not much match the behavior of the function, which means that the mapping of the function onto wavelet space will not be very continuous. The difference in basis dramatically changes the detail coefficients seen, as should be the case.

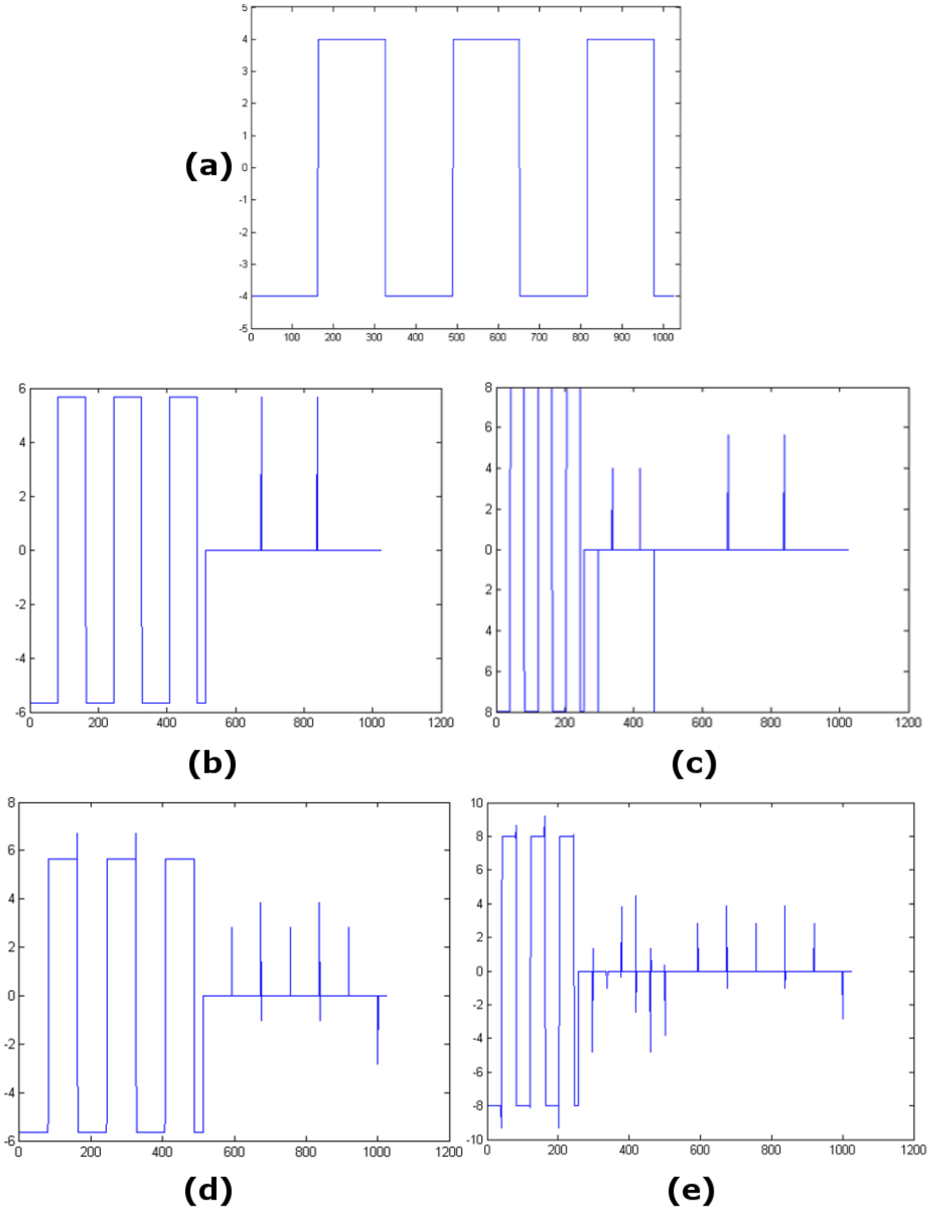
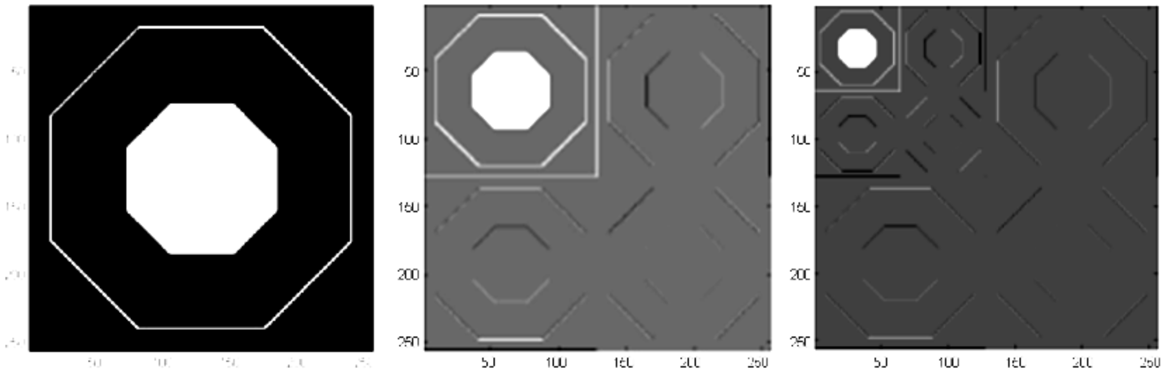


Figure 2. 7 DWT of a Square Wave Using Two Different Basis Pairs: (a) Original waveform (b) & (c) First and second Haar transforms, (d) & (e), First and second Daub4 transforms.

The wavelet transform can also be implemented in two or more dimensions by simply taking the one-dimensional transform in multiple directions as shown below.



(a) (b) (c)

Figure 2. 8 2D Haar DWT: (a) Original Image (b) First Level Transform (c) Second Level Transform.

The two-dimensional transform of the jpeg image (a) results in three separate regimes of detail coefficients. This is because the horizontal and vertical transforms overlap at some points of the dyadic selection. These points are therefore transformed in both the x and the y dimensions. Note that the detail coefficients directly below and above the down-sampled and scaled image (top left hand corner) in Figure 2. 8 (b) are approximating the vertical and horizontal components of the pixel field, respectively, and that the diagonal component has been approximated in both the x and y directions. It is easy to see this because the details located diagonally below the down-sampled and scaled original figure only detect discontinuities in the diagonal direction. Those directly below and right of the down-sampled image detect all of the discontinuities in the vertical and horizontal direction (which can be determined because lines of constant value in the direction of the transform are missed).

The examples above clearly show that the discontinuities can be spatially or temporally located in multidimensional functional spaces if the correct choice of basis is made. These properties make the wavelet transform an excellent choice for the current objective; however, it still has a few pitfalls. The application of wavelets to the specific task at hand is further discussed below.

2.2.6 Application to the Current Objective

The discrete wavelet transform has functionalities that are well suited to the type of applications of interest in this work. However, the LMR process that is used in the compressible [3] and incompressible [17] codes creates non-uniform positional increments and the adaptation of CFL number due to this refinement also creates non-equidistant time steps. As discussed above, this limits us to either remapping the domain in space and time and then applying the wavelet transform or finding a wavelet that can be used with non-uniformly spaced unstructured meshes. The remapping option for dealing with non-uniform data sets is mathematically and computationally rigorous. In fact Jansen [11] has noted that remapping is “generally impossible in two or more dimensions”. This facility demands the use of what are called Second Generation Wavelets Transforms (SGWTs) or lifting wavelets. Therefore, we must further generalize the concept of the wavelet transform.

Even if remapping could be carried out, the fit of the basis function to the function being approximated becomes an issue (refer back to Figure 2. 7), as the “best basis” function for a given purpose must be found. Methods for “best basis” selection have been developed as in [18]. Basis selection methods, however, are usually based on minimum entropy [18], or other requirements which are generally only useful for compression, and in our case would lead to missed feature detection.

The multiresolution transform, which falls in the broad class of second-generation wavelet transforms, is a generalization of the wavelet transform that can be used for the analysis of non-uniform data sets. It proves to be a robust method and can be specially adapted to deal with discontinuities [19]. It is this type of transform that will be used in the work to follow.

2.3 The Multiresolution Transform

A Multiresolution Transform (MRT) is a functional transform which typically consists of a dyadic coarsening operation and a reconstruction operation. The MRT can easily be applied to non-equidistant data points or cell structures, and can be adapted to use Essentially Non-Oscillatory (ENO) stencils or Piecewise Polynomial Harmonic (PPH) reconstruction when dealing with discontinuities and piecewise functions as in [4, 20]. All of these transform properties are very well suited to the type of functions found in high speed compressible flowfields. The recursive use of the MRT in order to decompose a function into its size and scale components is known as Multiresolution Analysis (MRA). Multiresolution analysis can be used to efficiently filter and compress structures from an input data source.

The MRT was developed by Harten [19, 21], and stems from the concept of lifting [22]. Lifting generally implies that a decimation and recreation step is used to lift information (in this case scales) from the function. The lifting concept can easily be seen through use of the Haar wavelet filter coefficients, as shown in Eqns 2. 16 and 2. 17. It can be seen that the Haar wavelet filter coefficients are nothing more than a normalized computation of the average value found between two points and difference between the average found and the actual value.

For example let us consider a function, *f(x)*, obtained at a certain sample size, n_0, (represented by starred points in Figure 2. 9). The function may then be evenly down-sampled to a sample size of n_1 and then used to predict the functional data at the finer sample size n_0. The error of the prediction attained from the down-sampled function is then recorded at each eliminated point. These error measurements are defined as the detail coefficients $d(n_i)$, as taken from the wavelet analysis. Figure 2. 9 shows a visual representation of the basic concept of a one dimensional point based MRT (exaggerated as it may be to create enough error to be visible). The difference in height between the actual function (line) the function predicted (dashed line) by the down-sampled points used (the boxed points shown) and the original sample points (the *'s) are the detail coefficients of the function. The linear prediction case of the MRT crudely represented in Figure 2. 9, is in fact the Haar wavelet transform without a normalization coefficient. The prediction scheme can simply be taken to a higher order in the MRT.

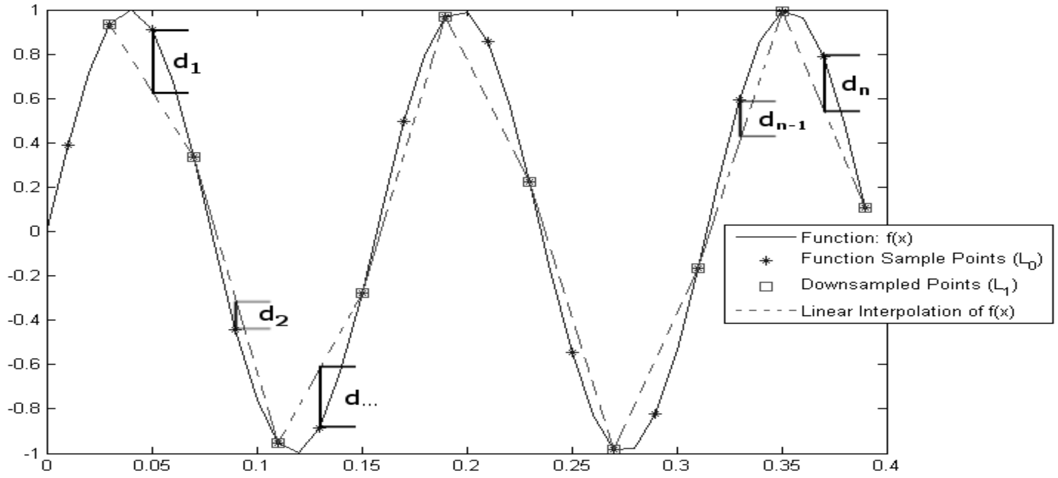


Figure 2. 9 The Basic Concept of the MRT.

2.3.1 Definition

2.3.1.1 Point-Based MRT

Mathematically the MRT is very simple. A one-dimensional function with an even sample size, n_0, can be transformed by use of an interpolation scheme to approximate odd numbered function values through the even function values. A Lagrange interpolating polynomial is used as a prediction operator as in [21, 5] for the reconstruction of n_0 from n_1. As is the case in Figure 2. 9 above, the prediction function slides along the remaining points to predict the value of the down-sampled point centered between them. The Lagrange polynomial interpolation $P(x_c)$ about an odd point, x_c, centered in between the interpolation points for such a scheme may be expressed as:

$$P(x_c) = \sum_{j=-m/2+1}^{m/2+1} f\left(x_j^k\right) l_j$$

$$l_j = \prod_{j=-m/2+1,\, j\neq i}^{m/2+1} \frac{x_c - x_j}{x_i - x_j}$$

Eqn 2. 23

Detail coefficients as defined in the MRT are simply the difference between the original function and the prediction of the functional value at x_c. The point based MRT may be expressed as below in Eqn 2. 24.

$$s_j^{k+1} = f\left(x_{2j}^k\right)$$

$$d_j^k = f\left(x_{2j-1}^k\right) - P\left\langle f\left(x_{2j}^k\right); x_{2j-1}^k \right\rangle$$

Eqn 2. 24

Note that the detail coefficients, d_j, are grouped together at the end of the transform. This regrouping of terms (again, as is the case for point based wavelet transforms) is typical for point based transforms because it provides good visual feedback and allows for easier recursion.

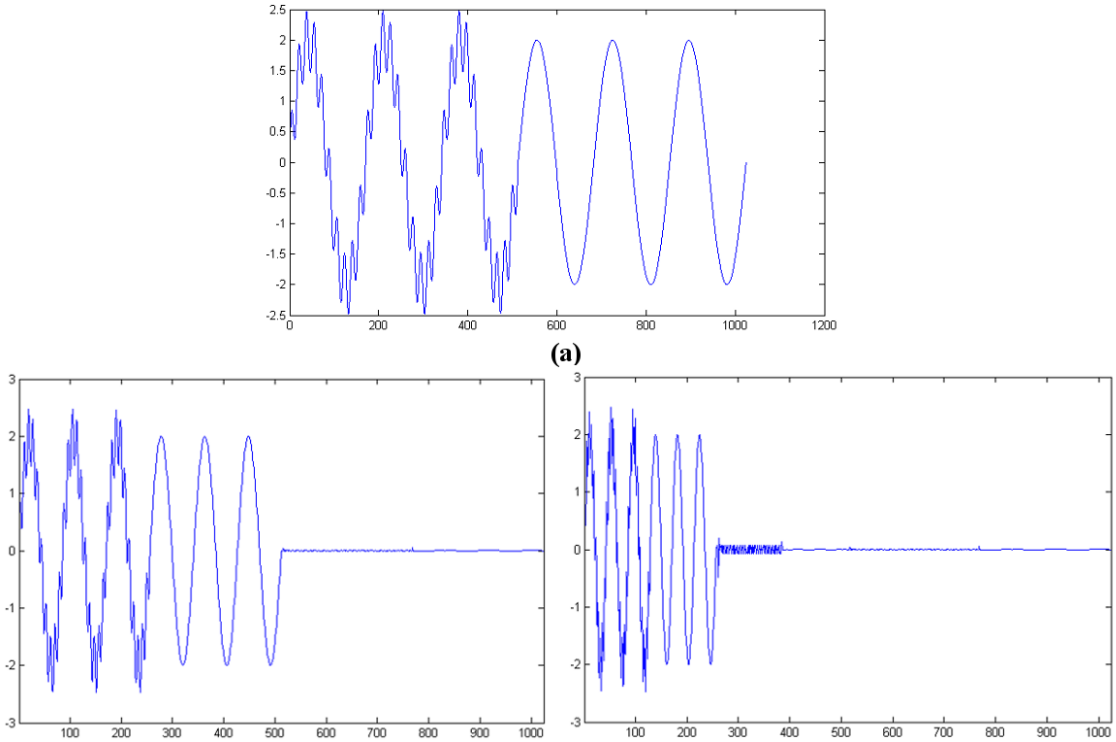
Recursion can be done by again dyadically down-sampling the remaining portion of the signal s. In doing so, the second transform now contains error data from a smaller (and thereby coarser) sample. The recursion can continue until there are only *m* points left in the signal, where *m* is the number of points used for the interpolation.

The inverse MRT may then be found by simply reversing the multiresolution process as given in [4]:

$$\begin{aligned} s_{2j}^{k-1} &= s_j^k \\ s_{2j-1}^{k-1} &= d_j^k + P\left\langle f\left(x_{2j}^k\right); x_{2j-1}^k \right\rangle \end{aligned} \qquad \text{Eqn 2. 25}$$

Examples

Figure 2. 10 shows the first (b) and second (c) level point-based multiresolution transform of the original function (a). Notice that there are still enough points left after the first down-sampling to semi-smoothly reproduce the original function, as seen by the detail coefficients on the right side of the first level transform shown in (b). (The transform below is represented in a reordered index format, however, it should be noted that the spatial or temporal information of the detail coefficients are retained and unaltered. The transform may also be represented in a high and low pass band format.) The effects of the loss of orthogonality can be easily seen in the transform when compared to the Haar wavelet representation. The detail coefficient behavior in the Haar transform of the same function (Refer to Figure 2. 5) displays the underlying periodic function along with the arrangement of detail coefficients, due to the Haar vanishing moment ignoring the signal average at each set of translation points. The discontinuity can also more easily be seen in the MRT representation of the detail coefficients.

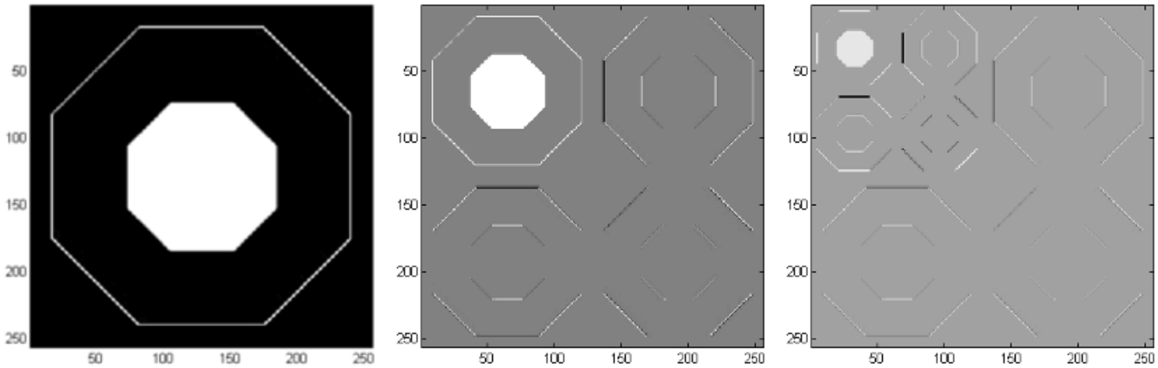


(a)

(b) **(c)**

Figure 2. 10 1D MRT (a) Original Function as defined in Eqn 2. 10 (b) First level (c) Second level MRT.

The MRT can also be extended to two dimensions by simply applying the one dimensional transform in two directions. Figure 2. 11 below shows the first and second level two dimensional transform of an image with uniform grid spacing. The original image is decreased in resolution from a 256^2 pixel image to a 128^2 pixel image through the dyadic down-sampling in the first transform, seen on the top left hand side of (b). The rest of the outputs in the first level transform are the detail coefficients. Details on the top right are the product of the horizontal transform (d(x)), details on the bottom left are the products of the vertical transform (d(y)), and details on the bottom right are the products of both vertical and horizontal directional transforms (d(xy)). The process is then repeated for the second level MRT (shown in (c)) using the 128^2 pixel picture as an input signal.



(a) (b) (c)

Figure 2. 11 2D Point Based MRT: (a) Original Image (b) First Level Transform (c) Second Level Transform.

2.3.1.2 Cell-Based MRT

The one-dimensional cell-based transform is much like the one-dimensional point based transform; however, the decimation process is modified to encompass a parent-cell based coarsening scheme. This is done by predicting values using the average point value of two adjacent cells (parent cell value), rather than a single point value. The cell based MRT may be described by the equations given in [4].

$$\begin{aligned} s_i^{k+1} &= \bar{f}\left(x_i^{k+1}\right) = \frac{1}{2}\left[f\left(x_{2j-1}^k\right) + f\left(x_{2j}^k\right)\right] \\ d_i^k &= f\left(x_{2j-1}^k\right) - P\left\langle \bar{f}\left(x_j^{k+1}\right); x_{2j-1}^k \right\rangle \end{aligned} \qquad \text{Eqn 2. 26}$$

, where $P\langle\rangle$ is the prediction operator (the Lagrange interpolant) acting on $\bar{f}$ (the average functional value of two neighboring points) for the point x.

The inverse transform is then found by again simply reversing the steps taken during the decimation process, and is given by:

$$\begin{aligned}\bar{f}^{\,k}_{2j-1} &= P\left\langle \bar{f}^{\,k-1}_{j} ; x^{k-1}_{2j-1} \right\rangle + d^{\,k}_{j} \\ \bar{f}^{\,k}_{2j} &= 2\bar{f}^{\,k-1}_{2j-1} - \bar{f}^{\,k}_{2j}\end{aligned} \qquad \text{Eqn 2. 27}$$

In two dimensions the decimation process on an equally spaced grid would be as shown in Figure 2. 12. Arrows represent the merging or averaging of two cells to the central point. The prediction function would use these central point values as the *k+1* level. It should also be noted that the transform in the second direction only produces one central value, and therefore, the two dimensional cell based MRT will produce three independent detail coefficients per cell at level *k+1*.

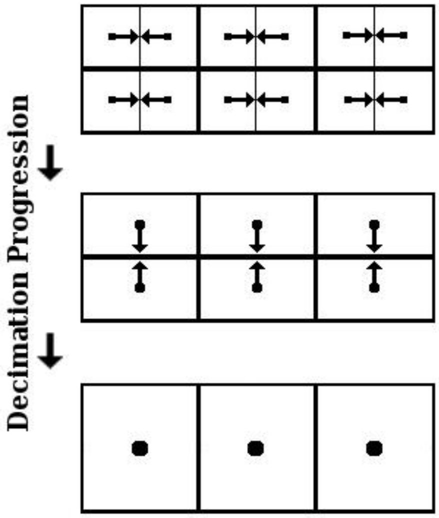


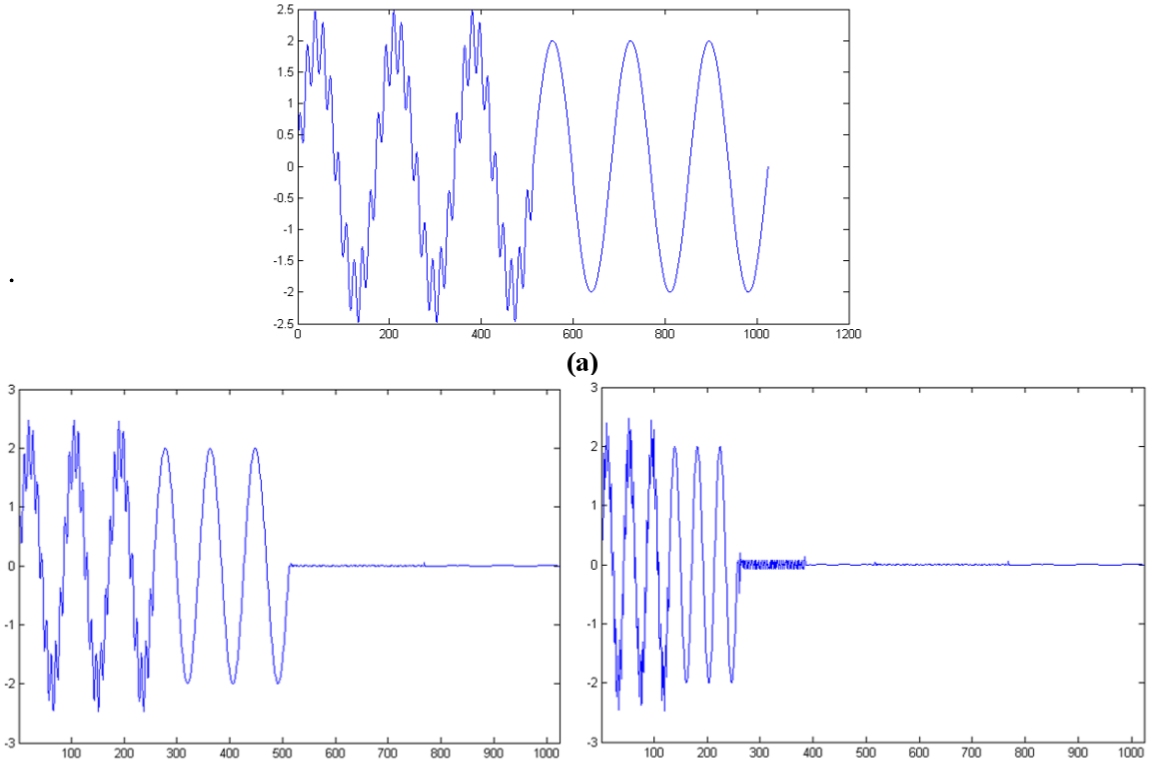
Figure 2. 12 Cell Based MRT 2D Decimation Process.

Examples

The one-dimensional transform of the function described in

Eqn 2. 10 above is shown in Figure 2. 13. The cell based transform visually gives about the same results. As the point based transform in one-dimensional space. In two dimensions the transform cannot however be represented in the same way as above.

.



(a)

(b)

(c)

Figure 2. 13 1D Cell Based MRT of the Function Given by Eqn 2. 10: (a) Original function (b) First level 3rd order MRT (c) Second level 3rd order MRT.

Again the two-dimensional transform can be obtained through use of the one-dimensional transform in two dimensions. Figure 2. 14 shows the transform of an image in a high and low pass manner.

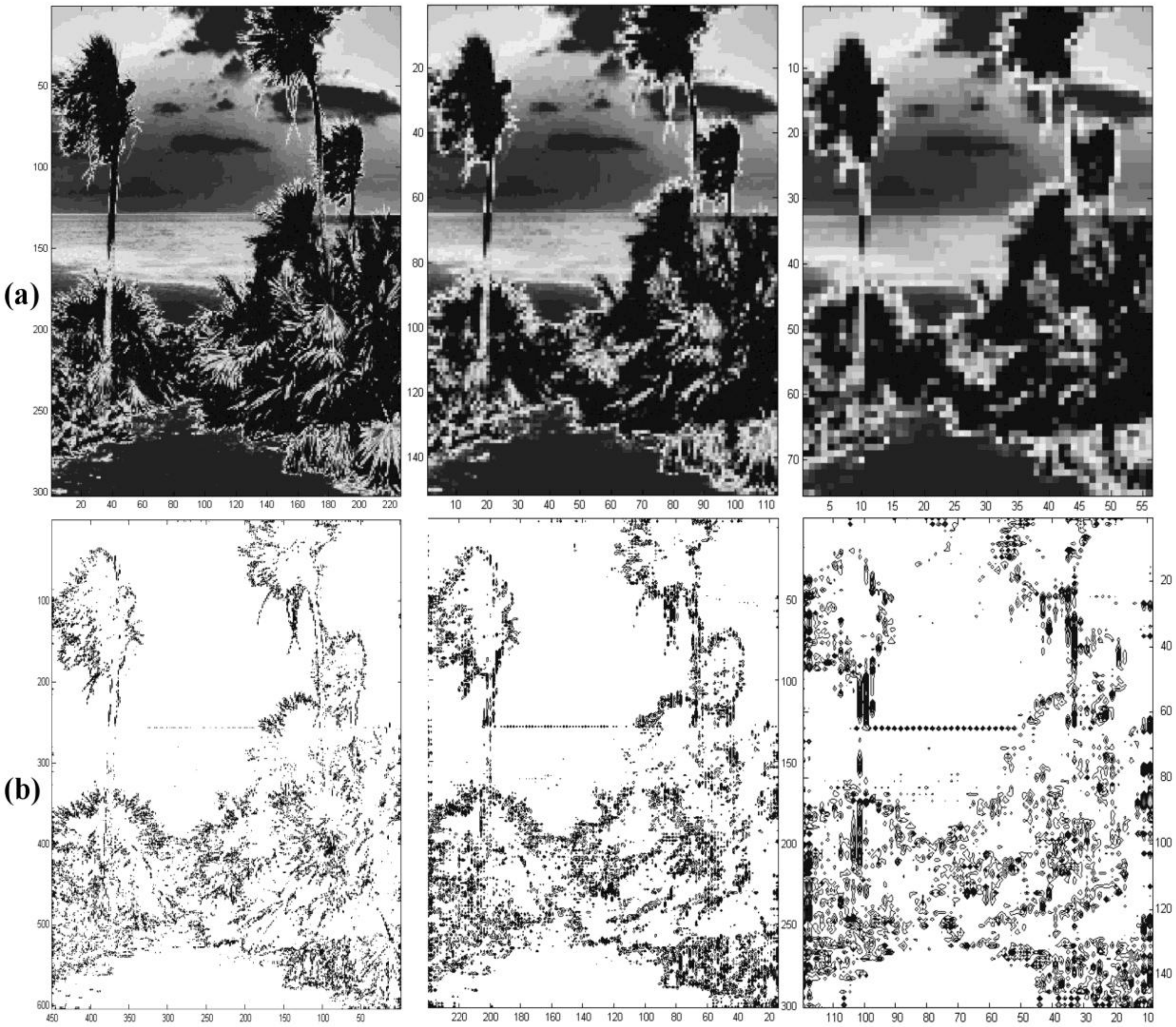


Figure 2. 14 Cell Based MRT of an Image (a) The subsequent low pass images (b) Contour lines of detail the detail coefficients for each MRT level.

Figure 2. 14 shows the first through third levels of transform of a jpeg picture. Row (a) contains the average signals after the first, second, and third transforms. Row (b) contains the contours of detail coefficients associated with the corresponding average signals. Row (a) and (b) may be viewed as a low and high pass filter of the signal. The averaging of cells during the cell based MRT causes a successive smoothing of the field, which can in turn be seen in the detail contours by a widening of the area over which the detail value occurs.

2.3.2 Properties

The first thing that should be noted about the multiresolution transform is that unlike the Fourier and wavelet transforms, it does not depend on a set of orthogonal basis functions. The use of the Lagrange interpolant basis enables it to have compact support, while also allowing it to be used on non-equidistant data sets. The use of a scaling function is also not needed, and thus the representation of data at lower resolutions is immediate (because there exists no scaling function in the MRT’s definition). The loss of orthogonality also surprisingly does not decrease the MRTs ability to compress data as compared to the wavelet transform as noted in [20], although it does decrease its ability to denoise systems with the same efficiency (Since the basis functions are no longer orthogonal the thresholding parameter is not simply linear with respect to the transform level. This makes it slightly more difficult to find the “correct” thresholding parameters).

2.3.3 Common Uses in Engineering

Since the transform properties of the MRT are similar to that of the wavelet transform, the MRT has many of the same applications. These applications include signal processing and recognition, data compression, and mesh smoothing.

The MRT is often used in engineering for 3D modeling and rendering [23, 24]. In many cases CAD or 3D models may be inaccurately represented by a coarser scale mesh during editing phases of modeling. In such cases the degradation of surface quality while editing may be due to a MRT mesh coarsening routine. Upon exit of the editing phase, or perhaps upon entry of a rendering phase, the model may be reconstructed in a manner that prevents an excess number of nodes from being used for generally smooth areas and allows high concentrations of nodes in areas of large gradient. This is done by a simple thresholding procedure, which is dependent on the visual accuracy set by the user. Instances of the MRT being used as a CAD or 3D modeling tool can be viewed in many works, including [23, 24].

2.3.4 Uses in CFD

The original intent of the MRT was to provide decompositions for use with hyperbolic conservation laws. As described in [5] the advantages of the use of the MRT in compressible flow calculations comes from the ability to disregard complex flux computations where the flowfield is shown to be smooth by MRT decomposition. In addition, the ability of MRT to operate on data containing non-uniform sample spacings and even triangulated unstructured meshes [11] renders it a suitable choice for CFD applications. Furthermore, ENO-MRT is specifically suited to handling discontinuities such as shocks in the flowfield and blends nicely with the use of the ENO scheme for the advection of fields in compressible flows.

2.3.5 Application to the Current Objective

The presence of discontinuities, as shown above for the Fourier transform, often creates large error in the functional approximation. It was shown that for wavelet transforms, the use of the Haar wavelet, due to the shape of its basis functions, can minimize this error when dealing with a pure step function, but that another basis would need to be selected in dealing with smoother functions. For these reasons the selection of “the best” basis functions can be a deciding factor in the error or detail coefficients obtained when using a wavelet transform.

The MRT is subject to the same error increase when used across a discontinuity, however, as the Lagrange interpolant produces Gibbs-like oscillations about the discontinuity as depicted in [4]. These oscillations are further worsened by increasing the interpolation order due to Runge's phenomenon [25]. Discontinuities if present must therefore be given special treatment when using the MRT, in order to correctly spatially (or temporally) place detail coefficients. The next two subsections deal with two methods commonly used to prevent the Gibbs-like oscillation patterns that corrupt the detail placement.

2.3.5.1 Use with PPH Reconstruction Schemes

The Piecewise Polynomial Reconstruction technique as defined by Amat et al. in [20], is a scheme which seeks to reconstruct the areas around a jump discontinuity in order to suppress an excess of detail coefficients in the area surrounding it.

Reconstruction is done using a 3^{rd} order centered Lagrange interpolant, in which the harmonic mean of neighboring cells is used in order to reconstruct the jump discontinuity, rather than the arithmetic mean. The idea behind reconstruction in such a manner is that harmonic mean is always bounded in regions at or near a finite jump condition. Refer to [20] for a more in depth explanation.

2.3.5.2 Use with ENO Schemes

Essentially Non-Oscillatory (ENO) schemes can also easily be incorporated into the MRT process through the use of interpolatory stencils. ENO methods sample function values around the point being interpolated (x_c) in order to select the interpolation scheme containing the smallest amount of discontinuity. These samplings are known as stencils. The least discontinuous ENO stencils are chosen using the minimum Newton's divided difference method, *Đ*, as used in [4, 26], and defined as below in Eqn 2. 28.

$$Đ_j = \frac{f(X(i+1) - f(x(i))}{x(i+1) - x(i)}, \quad Đ_{j+1} = \frac{Đ_j}{x(i+2) - x(i)}, \ldots, Đ_{m-1} \qquad \text{Eqn 2. 28}$$

The *m* data points corresponding to the minimum divided difference are then used for the interpolation stencil.

The advantages of using ENO schemes with MRT can be seen after applying the inverse MRT transform to a discontinuous function as shown in Figure 2. 15(a). Figure 2. 15(b) shows the second level transform of a square wave, and (c) is the attempted recovery of the square wave without the use of ENO. Notice the Lagrange interpolation causes an oscillation about the discontinuity better seen in Figure 2. 15(d). The Gibbs-like oscillation of the detail coefficients can be seen in Figure 2. 15 (b).

This behavior is also noted in [4, 27] and is caused by the fact that the Lagrange interpolation is piecewise smooth. The ENO-MRT, however, reproduces the exact function when the inverse is taken because the interpolatory stencils were shifted left or right of the discontinuity (Refer to Figure 2. 15(e) and (f)).

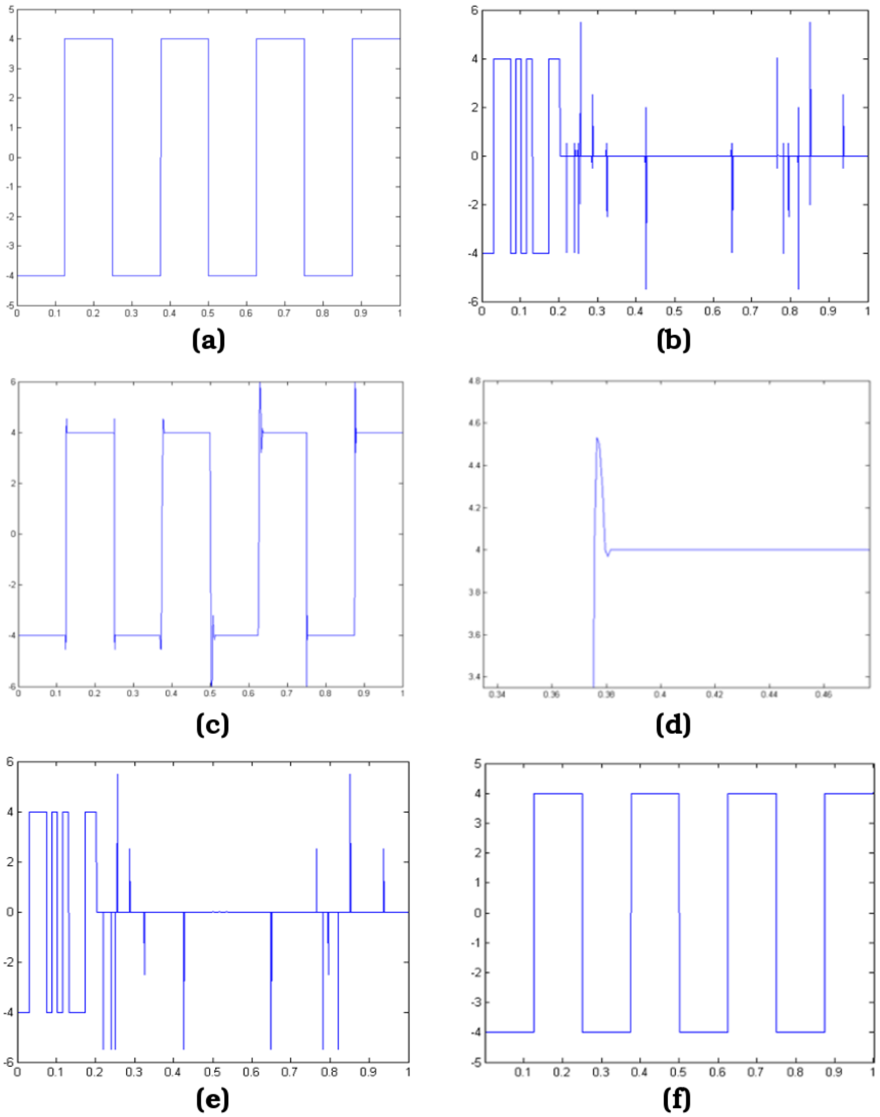


Figure 2. 15 MRT and Inverse MRT with and without the Use of ENO Schemes: (a) Original function, (b) MRT without ENO, (c) MRT inverse without ENO, (d) Gibbs-like oscillation upon inverse, (e) ENO-MRT (f) ENO-MRT inverse.

The same behavior occurs when the detail coefficients are created during the forward transform; therefore ENO is necessary in order to achieve accurate, spatially localized detail coefficients when dealing with discontinuous functions. The Gibbs-like oscillation of the detail coefficients can be seen in Figure 2. 15 (b). The downside to such a method is that if the stencil selected is off centered, then the same stencil must be used in order to regain the signal, *s*, and therefore an integer definition of the stencil choice needs to be stored in order to correctly remap the function (e.g. a centered stencil has value 0; shifted left one cell a value -1; shifted right one cell a value of 1).

The ENO-MRT transform described above is chosen in the present work due to its ability to handle non-uniform point spacings and the facility to handle shocks and other discontinuities in the flowfield. The challenge in devising a suitable ENO-MRT scheme is the development of criteria for identifying and classifying structures according to their scales and strengths. An extensive study of the use of ENO-MRT to analyze flowfields is presented in the following chapters.

CHAPTER 3

USE OF THE MRT IN THE ADAPTIVELY REFINED CARTESIAN GRID FRAMEWORK

While the implementation of the MRT for equidistant data sets in the one and two dimensional frameworks may be straight forward, the implementation of the two and three-dimensional cell-based ENO-MRT within the octree/quadtree mesh structures contained in the present CFD framework is quite a bit more challenging. There are certain constraints which need to be addressed, including cell averaging, data storage methods, spatial remapping, and thresholding and refinement parameters.

The use of the MRT is very basic; it functions as a post- or pre-processor (depending on the application for which it is intended). It does not involve changes to the Navier-Stokes solver itself. The MRT is applied to the computed flowfield so that regions of sufficiently high gradient of the flux conserving solution at the current time step can be located and flagged for refinement or coarsening as needed. Refinement and coarsening indicators are specified through the use of the MRT and the refinement and coarsening operations themselves are carried out in the same manner as in the gradient-based method which includes flux conservation (We are therefore not concerned with flux conservation while performing the MRT, because its use more resembles that for image processing). The MRT is used as a pre-processor that uses as its input the numerically calculated flowfield (and operates on a designated flow variable, such as density, velocity or pressure) in order to identify local features and their scales.

3.1 The Octree/Quadtree Mesh Structure

First the underlying structure of the octree/quadtree mesh must be fully understood. Octree/quadtree mesh structures are based on a uniform Cartesian base mesh. When refinement is triggered, a base cell is divided into eight/four (in 3D/2D) leaf cells. The eight or four leaf cells then reference the uniform base cell in order to gain neighbor data. If one of the leaf cells needs further refinement, the leaf cell again subdivides, and then references its parent. The result is a hierarchical tree-like structure, which eventually narrows to become the nodes of the uniformly spaced base mesh. Figure 3. 1 depicts the quadtree mesh structure used.

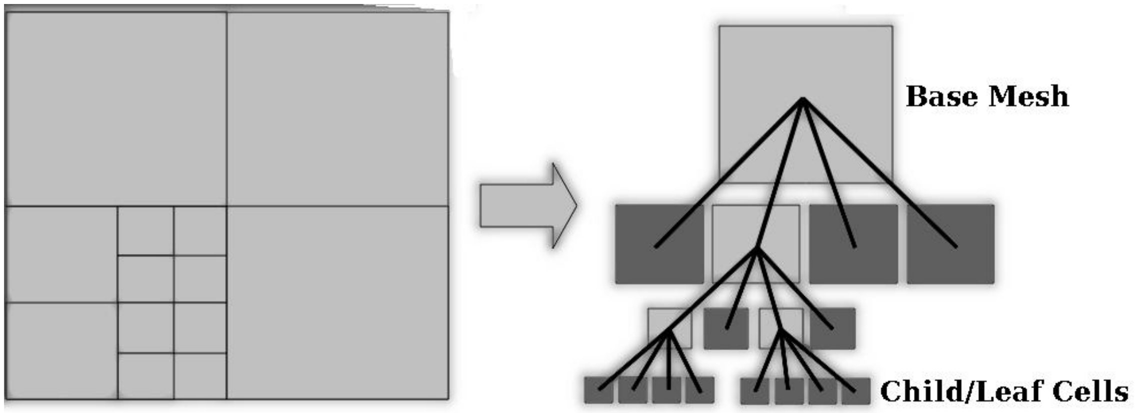


Figure 3. 1 Quadtree Cell Structure.

Within our current framework, the leaf cells (darker cells in Figure 3. 1) contain all data associated with the flow field. One may either directly access the leaf cells or the base cells; all other levels between the two are used for cell connectivity information only (i.e. no data is stored nor computations performed at these intermediate cell levels).

3.2 Gathering Needed Cells

The first task that needs to be performed is the gathering of necessary data points in order to form an ENO stencil. In order to do this cells at the highest level of refinement, k_o, must be transformed followed by those of the next lowest refinement level, $k+1$, and so on, thus, progressively coarsening (termed decimating) the grid until it is one of uniform spacing. Figure 3. 2 shows the progression of a mesh being coarsened until it has reached a uniform mesh. Once a uniform mesh has been reached the process can be terminated or continued (in which case the decimation process is simplified to an equal point spacing scheme).

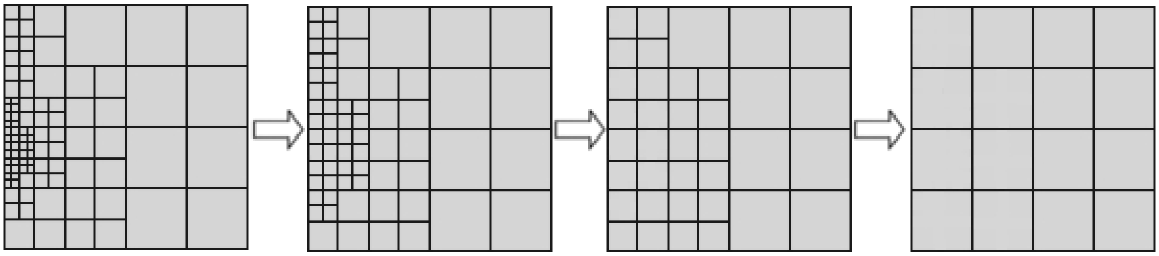


Figure 3. 2 Cell Agglomeration Process on a Locally Refined Mesh.

Once this pattern of coarsening is clear, one can start the selection of the central points, x_c, about which the interpolation will be performed. Since there is a cell averaging that occurs as well as an ENO stencil selection, a total of $2(m-1)$ points, where again m is the number of points needed for the interpolation (or the interpolation order plus one) must be selected on either side of the point of prediction, x_c. In order to account for changes in levels during this cell gathering phase of the cell-based MRT, cells at levels coarser than the current finest level are considered "pre-averaged" values and are counted as two cells in the stencil. Averaging of cells at resolution levels coarser than the current one of interest are not needed and do not reflect the next coarser representation of *f(x)*. The pseudo-code for the procedure is given in Appendix A under *Cell Selection*. A visual representation of the stencils cell count is also shown in Figure 3. 3.

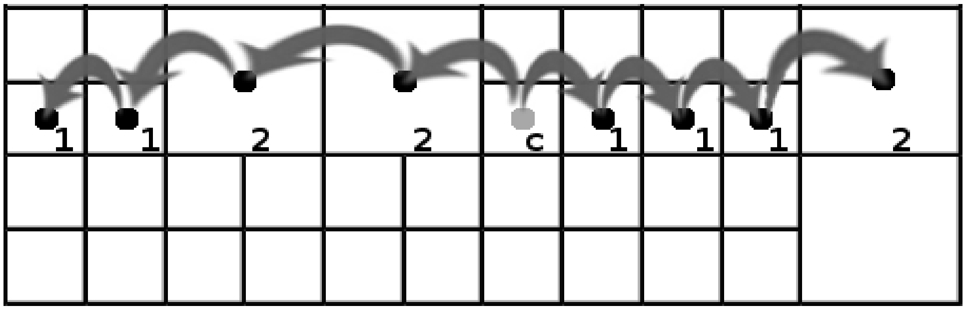


Figure 3. 3 Major Stencil Selection Procedure: Point c is the central point.

Another possible occurrence when building the major stencil on a locally refined grid is a fluctuation of cell refinement level. Suppose a cell at resolution *k* has a neighboring cell at *k+1* (*k* being the finer level), which in turn has a neighboring cell at resolution *k*, as depicted above to the left of the central point c (the cells become coarser and then refine within the range of the stencil). In such a case the cells used for the stencil should be in the same row as the cell being processed to avoid the possibility of surplus fluctuations of *f(x)* in the current stencil.

3.3 Cell Averaging

In the octree/quadtree framework, the eight/four cells that make up a parent cell at level, $k+n$, must only be averaged with one another. One must check the hierarchical tree structure associated with the mesh in order to make sure that only cells at the same level and which share a common parent are averaged. (See Appendix A, *Cell Averaging*, for the basic averaging pseudo-code).

If this check is not performed then the resulting first transform will yield hanging nodes with positions which do not reflect the geometry of the parent mesh. Hanging nodes during a transform will produce details with positions that cannot be remapped to the existing parent nodes.

3.4 Stencil Selection

Stencil selection is done by simply computing the divided difference of the points contained in the averaged or minor stencil. The last Newton’s divided difference may be done as shown in Appendix A under *Divided Difference*. The stencil containing the minimum divided difference is then used to compute the detail coefficient.

3.5 Detail Coefficient Storage

For unfiltered construction or lossless compression it was shown in Chapter 2 that all of the detail coefficients obtained by either a wavelet or multiresolution transform must be stored. All of the details at all levels k to $k+n$ must therefore be somehow stored. The storage of details on all levels may be done in a number of ways, storage of details on the leaf/child cells, storage of details on the current level of transform on the octree/quadtree tree structure, or even storage in a separate matrix.

The storage method used in the current CFD framework is to place data at the leaf/child cell level for all levels, due to several algorithmic advantages over storage on the current tree level and storage in an arbitrary matrix format. The reasoning and methodology for the storage system used is described below.

If the detail coefficients are stored on an arbitrary matrix, the problem becomes storage of multiple levels (perhaps using multiple inputs), which are not of congruent dimensions. This forces the detail coefficient matrix to either store all of the cells in the domain (most of which have zero value) or somehow create a matrix structure that can contain *k* matrices or arrays of different dimensions. In the context of most programming languages this is fairly difficult. The argument can be made that one could create a structure, which can account for all levels, but the number of places needed would have to be either pre-computed or allocated on the fly which may also present problems. Both high and low pass data would also need to be re-associated with the cells in some way, again leading to even more memory overhead. The access of the data in this manner, however, would probably prove to be the fastest as there are no searching routines needed in order to retrieve the data.

Storage of the details on each tree level is another solution for this problem. Tree level storage would provide an easy and fairly efficient means of storage. The problem in storing the coefficients in such a manner lies in the fact that the leaf/child cells may not be defined in the same manner as the parent cells. If this is the case, which in our context it is (Our method only allows one to directly access the leaf/child and base mesh cells. Refer to Krishnan [17].), the use of the code recursively at each level would be difficult, since during the first MRA iteration the leaf/child cell would need to be called and at all intervals thereafter the parent cells would need to be called. A way to avoid this would be to traverse up the tree from the leaf/child cell level while storing the detail, but then this becomes a problem because the number of nested loops required would be equal to *k+1* (One loop over all cells, one loop over all references).

The current storage method uses the leaf/child cells to store details on all levels. The required cells are first found using an array of the cells at the current level, where only the desired cells are stored in a temporary array. A certain cell out of any $2^{2(k+1)}$ cells gathered is allowed to be marked as having a lower MRT transform level, and the process is then repeated. This eliminates unnecessary traversing of the tree in order to store coefficients, as they are instead simply stored on only a few of the leaf/child cells, and in an array that only stores the needed amount of details. The storage procedure can be described by Figure 3. 4.

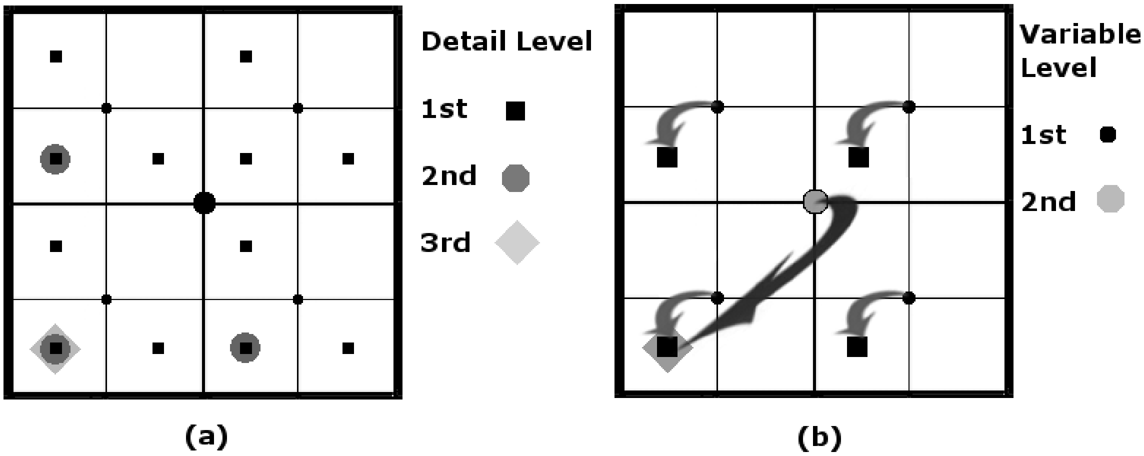


Figure 3. 4 Detail Coefficient and Averaged Value Storage Schemes Used.

3.6 Spatial Remapping

After all detail coefficients for all levels are stored a spatial remapping to the mesh must be made in order to determine which of the current cells contained relevant details. A level-by-level remapping of the details obtained on all levels of multiresolution mesh to the actual locally refined mesh must therefore be done.

It is important to note that there is a certain level of uncertainty in the exact location of the non-smooth data to within the next coarser representation of the data. The uncertainty is due to the averaging of the cells and the two-dimensional tensor product formulation used in the cell based MRT (there is no longer a clear bidirectional component in the formulation, and hence we lose some of the ability to gather data that is not perfectly vertical or horizontal). The detail coefficients obtained from the transform give an estimate of how smooth the region is between the two cells being averaged, i.e. one of the cells must have a predicted value with an absolute value far greater or less than the average value for a large error to exist. However, if the detail coefficient obtained in one cell is large one can only say that there is a discontinuity which lies between this cell and the neighboring cells. One may also see that if the cells were averaged in the other direction first, then the detail coefficients would be different yet would remain of the same magnitude [4]. We must therefore consider that all $2^{2(k+1)}$ cells may be contributing to the presence of the detail coefficient, and must select all for refinement or coarsening. In order to simplify this process *only the maximum of the 3 coefficients obtained at each level*, is mapped back onto the space which they occupy at the current level of transform.

3.7 Threshold Types

As mentioned in Chapter 2, the functional approximation may be truncated in order to remove features which are of little or no interest to the needed or correct function. The information gathered by each level of the MRT reveals features at different scales. We wish to resolve the relevant features found on their respective scales based on the magnitudes of the detail coefficients obtained; however, even with the use of ENO or

PPH reconstruction, there may still exist small bands of detail coefficients with small magnitudes (associated with Gibbs-like oscillations) about non-smooth areas of the solution at each scale. These small detail coefficients may also be due to numerical noise in the solution and may not be relevant to the refinement process. In either case, using all of the detail coefficients to spur refinement can cause large amounts of over-refinement (an example is shown later in Figure 4. 16).

There are two types of thresholding procedures which are commonly used in conjunction with wavelet and multiresolution transforms: hard thresholding (otherwise known as truncation), and soft thresholding.

Hard thresholding or truncation, as the names imply, is the use of a static cutoff parameter ε(k). The pruning of details is performed by setting:

$$d_j^k = \begin{Bmatrix} 0, & \left|\left(d_j^k\right)\right| < \varepsilon(k) \\ d_j^k, & \left|\left(d_j^k\right)\right| \geq \varepsilon(k) \end{Bmatrix} \qquad \text{Eqn 3. 1}$$

Thus a detail coefficient under the level dependent threshold value, ε(k), for all levels, *k* to *k*+*n*, is set equal to zero, and those above the threshold are left as is.

Soft thresholding methods are usually defined as:

$$d_j^k = \begin{Bmatrix} 0 & , & \left|\left(d_j^k\right)\right| < \varepsilon(k) \\ sign(d_j^k)\left|d_j^k - \varepsilon(k)\right|, & & \left|\left(d_j^k\right)\right| \geq \varepsilon(k) \end{Bmatrix} \qquad \text{Eqn 3. 2}$$

This is sometimes referred to as the wavelet shrinkage technique [15]. In this case those detail coefficients that are above the threshold limit are then modified by a subtraction of the threshold from the current detail coefficient. Soft thresholding can therefore be used when it is important to determine the relative size of the detail to the thresholding parameter.

3.8 Selection of Threshold Parameters

The selection of a thresholding parameter can be fairly challenging when the solution parameters and required accuracy are unknown prior to simulation. Prediction of the gradients of the solution can be difficult, if nearly impossible, due to jumps in the flow field variables that arise near shocks and solid boundaries. Fortunately, the selection of the thresholding parameters for the MRT is not as undefined, and there are theoretical guidelines as to the relationship of detail coefficients to flowfield behavior [4, 14, 15, 19, 20].

Arandiga and Donat [20] show that in the cell averaged MRT framework, the detail coefficients of a cell contained near the center of the interpolatory stencil (i.e. the point must not be an endpoint in the stencil) in regions of smoothness can be described by:

$$d_j^k = O\left(h_{k-1}^m\right) \qquad \text{Eqn 3. 3}$$

On the other hand for regions containing jumps in the function:

$$d_j^k = O\left(f^p\right) h_{k-1}^p \qquad \text{Eqn 3. 4}$$

, where $p \leq m$ and h is the cell size.

f^p is the pth derivative of *f*, and therefore a measure of the smoothness of the function, i.e. the change in slope within the current support length p (i.e. detail coefficients are high in regions of large changes in function steepness. Refer back to the examples in Chapter 2). This qualitative measure of the behavior of the details makes selection of a threshold parameter possible, especially in the linear setting.

In the linear case, using a centered stencil MRT (without use of the ENO stencil) the thresholding parameter is shown by [4] to be, ε(k), where:

$$\varepsilon(k) = \eta q^{k} \text{ ,where} q = 2^{D/p} \qquad \text{Eqn 3. 5}$$

,where D is the number of dimensions. The quantity η is then left to be determined, setting *p*=*m* would minimize the thresholding parameter ε. The threshold parameter, ε(k), is therefore usually defined as:

$$\varepsilon(k+1) = \frac{1}{2}\varepsilon(k) \qquad \text{Eqn 3. 6}$$

(for third order schemes), where is ε(k) is then the value η(k), which is found using a level by level comparison of the detail coefficients values and applying a curve fit [4].

The addition of ENO schemes to the MRT produces a nonlinearity in the transform which obviously has an effect on the validity of the thresholding parameter, due to a slight change in the magnitude and number of the details produced. The nonlinearity of the prediction operator therefore makes finding the quantity η difficult.

Numerous methods have been devised [14, 15, 28, 29] in order to avoid the need for human intervention when picking the detail truncation or thresholding parameter, simply because *a priori* knowledge of the relative strengths of the detail coefficients is needed to ensure all of the desired features are captured. It is important for our purposes that we retain important features at their correct scale, while refraining from any over refinement of other cells.

Error control methods as described in [4, 20] can give an idea of the final error in the entire reconstruction. This is done by setting the thresholding parameter in the following manner:

$$\left\| \bar{f}(x)^k - \hat{f}(x)^k \right\| \leq \varepsilon \qquad \text{Eqn 3. 7}$$

,where $\hat{f}$ is the representation of the reconstructed function $\bar{f}$ at level, k, after the thresholding procedure has occurred.

In order to perform this type of error control there must be a reconstruction at points where the detail coefficient has been truncated, in order to determine the error with respect to $\bar{f}$. ε(k) must then be iterated over until the desired amount of prediction error is reached. The combination of reconstruction and iteration creates a computationally slow procedure, and this type of procedure, while valuable, is not optimal for the selection of a threshold parameter.

Other means of obtaining a thresholding parameter include the use of statistical analysis of the detail coefficients at a given level. In particular the mean, maximum, standard deviation, median average deviation (MAD), and variance [14, 15] of the detail coefficient parameters may be used to select a statistically sound choice from the detail coefficient profiles. Common statistical thresholding techniques include:

- The TAWS algorithm as defined in [14]. The thresholding is done according to:

  $$\varepsilon(k) = \frac{\sigma(d^k) * \sqrt{2log(ncells_{LEVEL})}}{8} \qquad \text{Eqn 3. 8}$$

  , where $\sigma(d^k)$ is the standard deviation of the detail coefficients found at level k, and $ncells_{LEVEL}$ is the number of cells contained at the current transform level.
- The SURE method as defined in [15]. The SURE algorithm is designed to minimize the Mean Squared Error (MSE) of the detail coefficient profile given before and after the cutoff has been applied. The SURE method is defined below:

  $$\varepsilon = 0 \leq t \leq \sigma(d^k)\sqrt{2log(ncells_{LEVEL})}$$

  $$SURE(\varepsilon, d_j^k) = ncells_{LEVEL} - 2\mathbf{I}\langle |(d_j^k)| \leq \varepsilon \rangle + \sum_{j=1}^{ncells_{LEVEL}} min(|d_j^k|, \varepsilon)^2 \qquad \text{Eqn 3. 9}$$

The thresholding parameter ε is allowed to vary, and the corresponding to the minimum value of SURE is chosen to be the threshold. The operator *I* is an indicator function which returns a value of 1 if the argument is true and 0 otherwise.

- The average magnitude of the detail coefficients can also be used in order to select large detail coefficients in the flowfield. In the case of compressible flow, large detail coefficients in the domain correspond to regions containing well defined shocks, when density is used as the refinement parameter. Because of the sparseness of the detail coefficients while using ENO-MRT, it is suggested that the average be computed using cells contained at the current transform level which have a non-zero magnitude.

$$Mean(d^k) = \frac{SUM(d^k)}{ncells_{LEVEL>0}} \qquad \text{Eqn 3. 10}$$

- A simple variance or standard deviation test may also be performed by simply computing detail coefficients above $\zeta\sigma^n$, where ζ is a desired constant and n may be set to 1 or 2 to take on the definition of the standard deviation or variance. If the distribution of the detail coefficients is assumed to be the Gaussian normal, and the mean of the magnitude of the detail coefficients is assumed close to zero, then for n=1, a ζ value equal to 2.576 would neglect 99% of all of the detail coefficients, and if ζ is set to 1.00 then 68.26% of all detail coefficients should be neglected, and so on.

Other refinement methods could include a direct user defined truncation [5], in which a threshold is simply defined by the user input threshold parameter and ε is set using the definition in Eqn 3. 6. The use of manual truncation parameters might give insight into the general size of the detail coefficients captured and perhaps even allow us to view what type of other methods could be used to determine a “good” threshold value. Static user input thresholding obviously assumes the user has a priori knowledge of the detail coefficient profile, and cannot account for the fluctuations that may occur during cell averaging.

In this work we test the applicability of the above thresholding criteria in devising a robust refinement trigger in the framework of cell-based ENO-MRT transforms.

3.9 Refinement Criteria

The cell refinement criteria used are of great importance to the accuracy and computational cost of the flow field solution. It is, therefore, of great importance to select refinement and coarsening criteria, which produce the fewest number of cells while retaining a desired level of accuracy and computational cost.

The implemented refinement and coarsening criterion are as follows.

In the case of truncation or hard thresholding:

IF ($|d^k_j| \geq \varepsilon(k)$) refine to level k-1 (with k=0 being the finest mesh size)

IF ($|d^k_j| < \varepsilon(k)$) allow cell to coarsen if needed

In the case of soft thresholding, the above still applies, only ε(k) is set to zero after a the detail coefficients have been shrunk according to the soft thresholding definition in Eqn 3. 2. Thus, if the magnitude of the detail coefficient in a particular cell at level, *k*, is large enough to suggest a jump discontinuity may be present, then let it be refined to the next highest level, *k-1*.

A set of coarsening criteria must also be enforced so that areas with active, but fading, structures are not suddenly coarsened. The coarsening criteria described in [30], were adopted in order to retain cells that are possibly needed to describe weaker features.

1. The Wavelet Criterion: The wavelet criterion is described in [30] as measure of the local detail coefficient energy to within a neighborhood of 4 cells of the current cell at the current level. Defined by:

$$\sum_{j=1}^{80} \left| d_j^k \right| < \varepsilon(k) \qquad \text{Eqn 3. 11}$$

   If the detail coefficient energy present in the region exceeds that of the current threshold the cell should remain at the current level because locally there are structures present which may create other prominent features in the flow field.

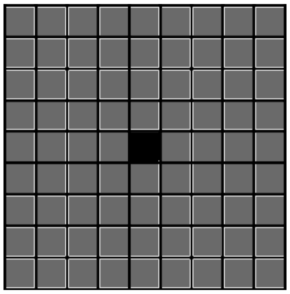


Figure 3. 5 The Wavelet Criterion: The black cell represents the current cell being queried for coarsening; all other dark gray cells in the domain would be used in the energy summation operation.

2. The Resolution Criterion : The resolution criterion states that the cell can be coarsened if all of the corner cells in the range of the next lower

resolution are either already at a coarser mesh size, or have not been selected for refinement at the current level (i.e. the detail coefficients within those cells are less than the truncation parameter given for the current level of transform).

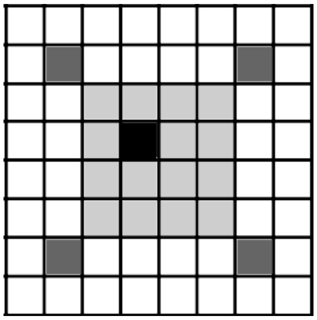


Figure 3. 6 The Resolution Criterion: The black cell represents the current cell being queried for coarsening; The light gray cells show the cells which describe the coarser transformation level, and the dark gray cells are the cells which are of interest.

3.10 Overview

The combination of all of the above constraints into the ENO-MRT framework should result in a fully functioning transform method for a locally refined mesh. The main goal of this method, again, should be that the important flowfield structures be correctly resolved at their individual scales, without triggering excessive over-refinement, and without any user input other than the threshold selection procedure. It is desired that the threshold selection method also be applicable to a wide variety of flow situations.

In the next chapter a number of statistically based detail coefficient thresholding parameters will be investigated for use with the ENO-MRT and resulting mesh refinement will be compared to refinement gathered for a constant human defined threshold, as well as for that given by the current gradient-based refinement scheme. These schemes are based on one or a combination of the above statistical methods, and include:

1. The TAWS Thresholding Algorithm (TAWS)
2. A Two-Pass Average-Outlier Thresholding Combination (TAO)
3. A Multivariable Average Thresholding Scheme (MA)

These methods are described in detail and their success in producing optimally resolved flow computations are evaluated in the next chapter.

CHAPTER 4

COMPUTATIONAL FRAMEWORK AND RESULTS

4.1 ENO-MRT Summary

The steps needed to perform the used ENO-MRT LMR scheme are as follows:

1. Select the flow variables to be operated on by the MRT and truncation scheme to be employed to trigger refinement/coarsening. As shown later, depending on the type of flow the more than one flow variable may need to be used to correctly identify flow structures.
2. On the first transform level and transform direction, assign all cells a MRT cell and a transform level counter. Initially assign this array to be the cells actual refinement level or a zero MRT level.
3. Find all of the necessary cells at the current MRT level. (Remember that only 3 out of every 4 cells will need to be selected. 2 are needed for the first directional transform, and 1 is needed in the second directional transform. Be careful not to select the same cells during different directional passes).
4. Loop over each of these cells and perform the following operations on them:
    a. Gather the major stencil of size 4(m-1) + 1 and temporarily store the following information:
        i. The cells current level of refinement
        ii. The connectivity of the cells obtained in the gathered array (Which cells can it be averaged together in the stencil? Is this cell a boundary cell?)
        iii. The cell position
        iv. All of the cells user defined variable values

b. Take the first 4(m-1) ordered entries in the major stencil created above

c. Correctly average the position and variable value information stored above, and store this information on a smaller minor stencil.

d. Store the average position and average values of the central cell in an array or on the cell structure for later transforms (the next coarsest signal representation).

e. Use Newton's divided difference (Refer to Eqn 2. 28 or Appendix A: *Divided Difference*) to determine the stencil which contains the least amount of discontinuity. Store the *m* positional averages and values in another temporary array. (As mentioned in [26], using the divided difference of the averaged parameters may limit visibility of higher order discontinuities and may be performed on the major stencil instead if needed).

f. Store the stencil selection on the cells (if needed for reconstruction).

g. Perform an *m*-1 order Lagrange interpolation on the *m* points returned by the divided difference in order to find the predicted value of the user defined variables at x_c. (See Eqn 2. 23 or refer to Appendix A: *Lagrange Interpolation*).

h. Calculate the detail coefficient as given in Eqn 2. 27.

i. Store the detail coefficient on the cell structure or in an array.

j. Flag 1 of the 4 child/leaf cells so that it may be ghosted as being on the next coarser level during the next pass (i.e. reduce the current MRT cell level counter assigned in *Step 2* by 1.).

5. Repeat for transforms in other directions.

   a. If in a second or third directional pass update the y and/or z components of the stored average position and value on the cell chosen to retain the information.
6. Repeat steps 1→5 for each transform level, until the desired number of levels is reached.
7. Loop backwards from the coarsest level and do the following:
   a. Plot maximum detail coefficient of the 3 obtained at each level back onto the actual mesh via a temporary array or temporary cell structure component (i.e. a detail coefficient will need to occupy $2^{2(k+1)}$ cells at each increasing level ($k = 0$ being the finest mesh transform).
   b. Loop through the array or cell structure containing the detail coefficient values to gather the statistical data needed.
   c. Once statistical values of the detail coefficients are found apply a threshold.
   d. Loop back over the cells to compare the detail coefficient size with the tresholding parameter. Apply the refinement and coarsening criteria at the same time through two Boolean flag arrays at each level (i.e. in a refinement array define 1=refine 0=coarsen).

It should be noted that in the case of flow solution around embedded objects the levelset narrow band [17] was completely omitted from any ENO-MRT calculations. This retains the highest level of refinement in a narrow band around the embedded object. This highly refined band is required to provide adequate resolution of the interface (i.e. the levelset field) and to accurately advect the levelset field in the case of moving boundary problems. From the MRT standpoint, omitting the levelset narrow band around the embedded interfaces is necessary because the treatment of solid objects in the current computational framework omits the interior of the object from the flow solution. Therefore, there is an intrinsic discontinuity in all the flow variables at the solid embedded boundary. . Even if the interior o0f the embedded object was not omitted from the flowfield as in the case of a solid body, there may be an extremely large detail coefficient present in the narrow band (for example in the case of a water droplet with $\rho_s = 1000.0$, and surrounding fluid (air) with $\rho_l = 1.0$). Very large details at the interface would cause a large upward shift in the statistical properties of the detail field, and possibly eradicate all other features in the flow field when using a globally determined statistical measure. Therefore, from various considerations, only detail values outside of the narrow band are used in evaluating the strength and scales of structures in the flowfield.

4.2 Test Cases

Numerical cases encompassing a wide variety of Mach numbers and possible interaction schemes were used to test the robustness of the MRT refinement scheme. Two classes of test cases were run using a single MRA reference functions in order to determine the ENO-MRTs expected detail coefficient magnitudes and the ability of different threshold selection routines to refine regions of high activity. These two classes are:

1. An immersed solid boundary case
2. A fluid-fluid interaction case

All of the parameters associated with these two cases are presented below.

4.2.1 Shocked Solid Cylinder Case

The first test case presented is that of a stationary solid cylinder computed on a half domain. Reflective boundary conditions were used at the half plane as well as on the top wall. Figure 4. 3 depicts the initial domain of the case and Table 4. 1 (below) sums up the relevant parameters used to define the problem.

A schematic of the prominent features found after impact of a shock wave with a solid cylinder is shown below. Features which we desire to be resolved in this simulation include the incident shock wave (I), the reflected shock wave (R, R’,R’’), the Mach stem (M, M’), baroclinic vorticity in the cylinder wave region, and the slip lines (S, S’) created by shear layer rollup [31].

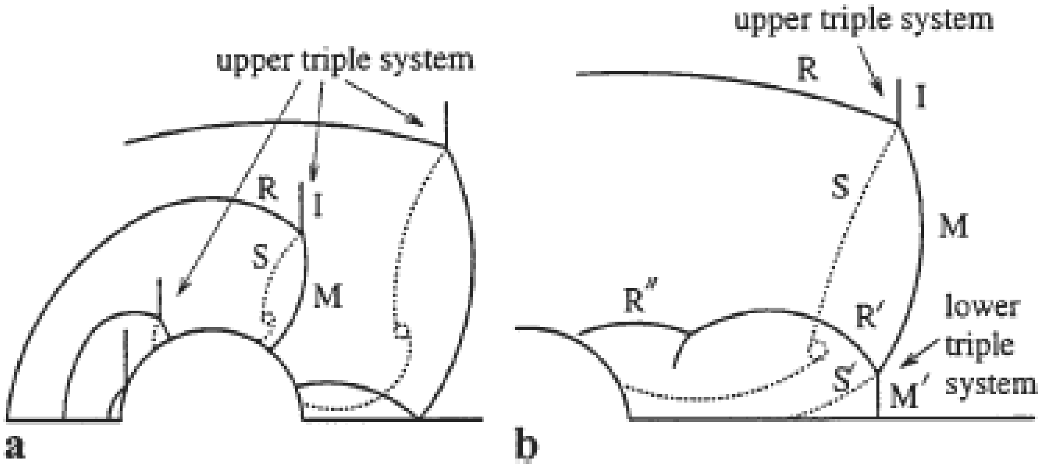


Figure 4. 1 **Schematic of Prominent Features of a Cylinder in a Schocked Flow Over Time: Incident shock (I), Mach Stem (M, M’), Slip-lines (S,S’), Reflected shock (R, R’, R’’) As shown in [31].**

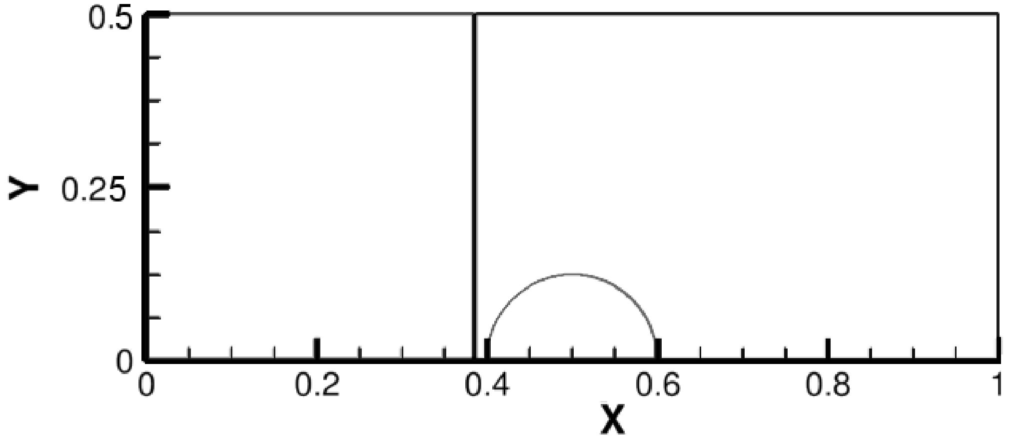


Figure 4. 2 **Cylinder test case: Initial geometry. Shock propagation from left to right.**

Table 4. 1 **Cylinder Test Case Parameters.**

Mach Number	2.81, 4.22		
Cylinder Attributes	Solid		Stationary
Cylinder Center Coordinates	x = 0.5	y = 0	r = 0.1
Initial Shock Location	x = 0.385		
Boundary Conditions	Left = Pressure Outlet		Right = Pressure Outlet
	Top = Reflective		Bottom = Reflective
Base Cell Size	dx,dy = 0.01,0.006,0.0033		4 levels of refinement

4.2.2 Shocked Helium Bubble Case

The second test case that is presented is a simulation of a helium bubble (also computed on a half domain) in air interacting with a shockwave. A reflective boundary condition is placed on the bottom domain boundary. All other boundaries are treated as pressure outlets. Figure 4. 2 depicts the initial flow field of the test case, and Table 4. 2 describes the geometric and computational parameters used.

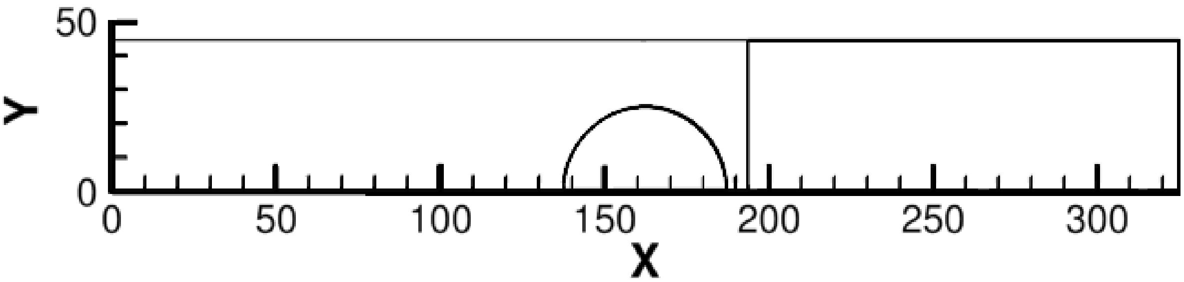


Figure 4. 3 **Helium bubble test case: Initial geometry. Shock propagation from right to left.**

Table 4. 2 **Helium Bubble Test Case Parameters.**

Mach Number	1.22		
Bubble Attributes	Soft		Stationary
Bubble Location	x = 162.5	y = 0	r = 25
Initial Shock Location	x = 193.75		
Boundary Conditions	Left = Pressure Outlet		Right = Pressure Outlet
	Top = Reflective		Bottom = Reflective
Base Cell Size	dx,dy = 1.00		4 refinement levels

The structures which should be located and marked for refinement of some degree in the case of a shock-bubble interaction are labeled in Figure 4. 4. Initially the predominant features include the bubble interface itself, the incident shock (I), the reflected shock (R), and the refracted shock (Rf) [32]. The expansion fan may also be required to refine, as many lower energy, high frequency waves may exist in this region. As the solution progresses the Mach stem (MS) and contact surface (S) should traverse through the bubble.

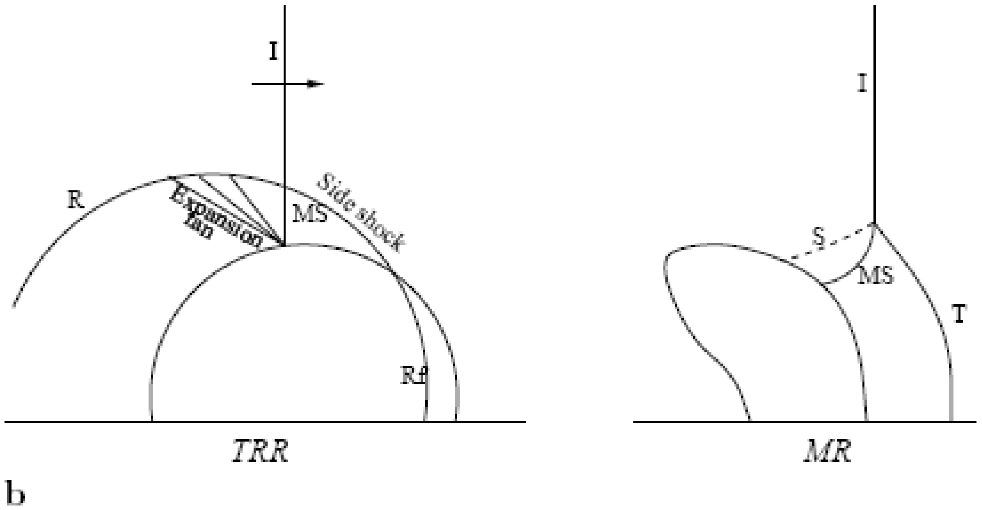


Figure 4. 4 **Schematic of Prominent Features of a Spherical Bubble in a Schocked Flow Over Time: Incident shock (I), Mach Stem (MS), Contact Surface (S), Reflected shock (R), Refeacted shock(RF) As shown in [32].**

4.3 Detail Coefficient Size

Relative sizes of detail coefficients obtained through computational simulations are of the utmost importance when determining a thresholding parameter or noise cut off parameter for refinement and data compression purposes. A study of the behavior of detail coefficient size with respect to Mach number was conducted. Table 4. 3 shows the trend in the average maximum detail coefficient of density over 600 time-steps for the solid cylinder test case.

Table 4. 3 Detail Coefficient Magnitude wrt Mach Number: Average maximum over 600 time-steps for the cylinder test case.

Mach Number	Avg. Detail Coefficient Magnitude	Normal Shock Density Ratio
1.5	0.2244	1.8621
2.5	0.8116	3.3333
3.5	1.3944	4.2609
4.5	1.7757	4.8119

It should be noted that there were fairly large oscillations in the detail coefficient size due to shock propagation through the mesh and the averaging constraints presented above. If the jump condition lands between two adjacent parent cells (of the current level), then there is no averaging allowed across the shock and thus a high detail coefficient is present. On the other hand if the discontinuity runs through the center of a parent cell (of the current level), averaging can take place across the shock, and details will be lower. The combination of the cell averaging constraints and the numerical representation of the shock itself passing through the mesh results in an oscillating detail coefficient pattern in time which is dependent on the CFL number.

It should also be noted that the average maximum detail coefficient presents a strong correlation to the normal shock density ratios due to the correlation between the size of the jump discontinuity and detail coefficient size described in [19, 26].

Contour plots of the detail coefficients captured at each scale are shown below.

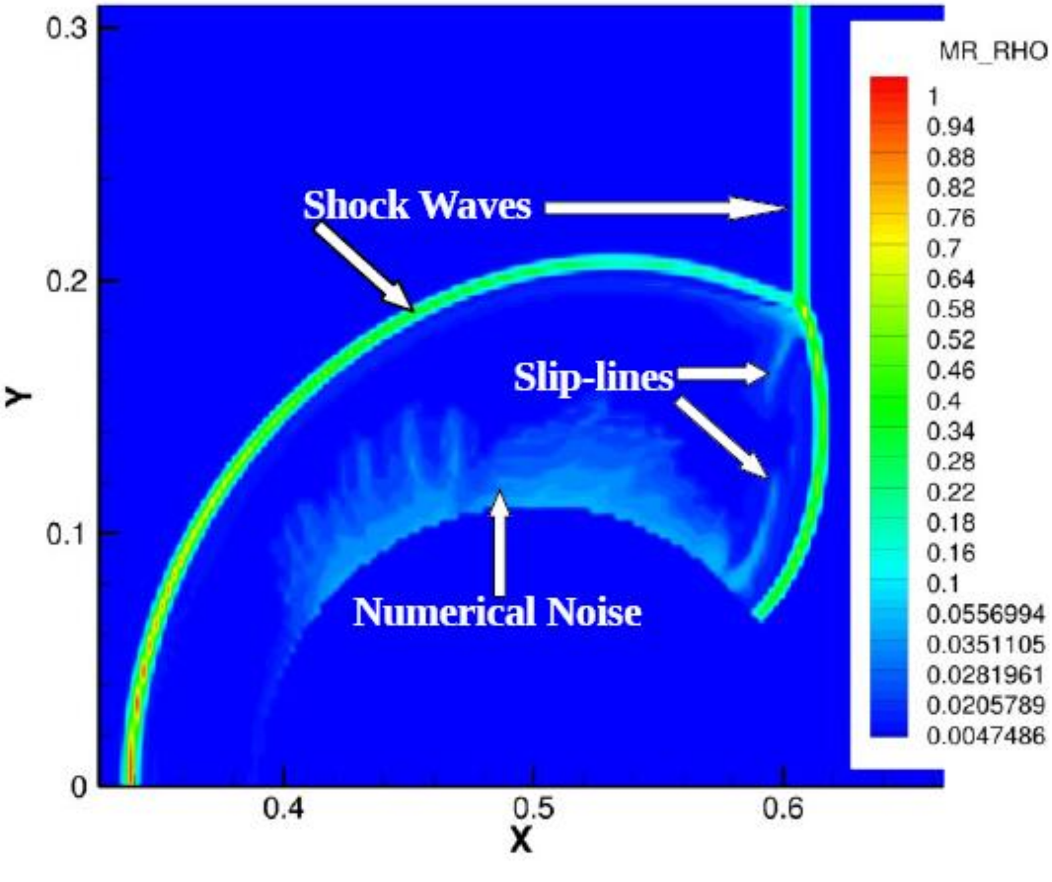


Figure 4. 5 Detail Coefficient Contour of the Finest Level for the Cylinder Test Case at Ma = 2.81, for Density Function Input.

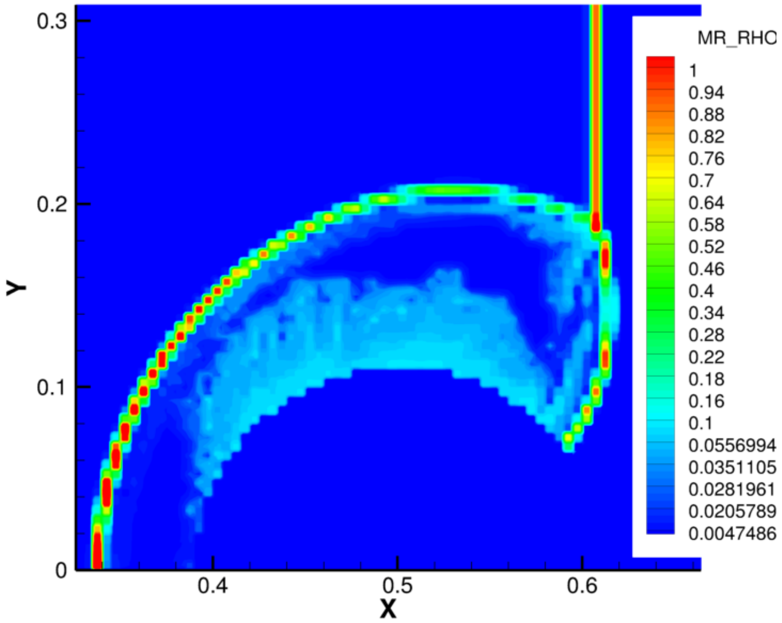


Figure 4. 6 Detail Coefficient Contour of the 1st Level for the Cylinder Test Case at Ma = 2.81, for Density Function Input.

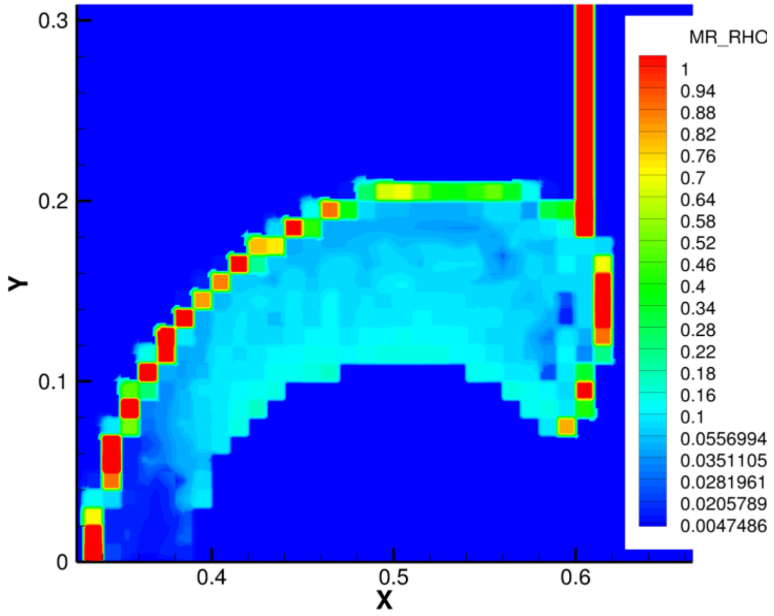


Figure 4. 7 Detail Coefficient Contour of the 2nd Level for the Cylinder Test Case at Ma = 2.81, for Density Function Input.

The percolation of detail coefficients from one level to the others should also be noted. Large discontinuities continue to produce large detail coefficients on coarser levels as verified by [28].

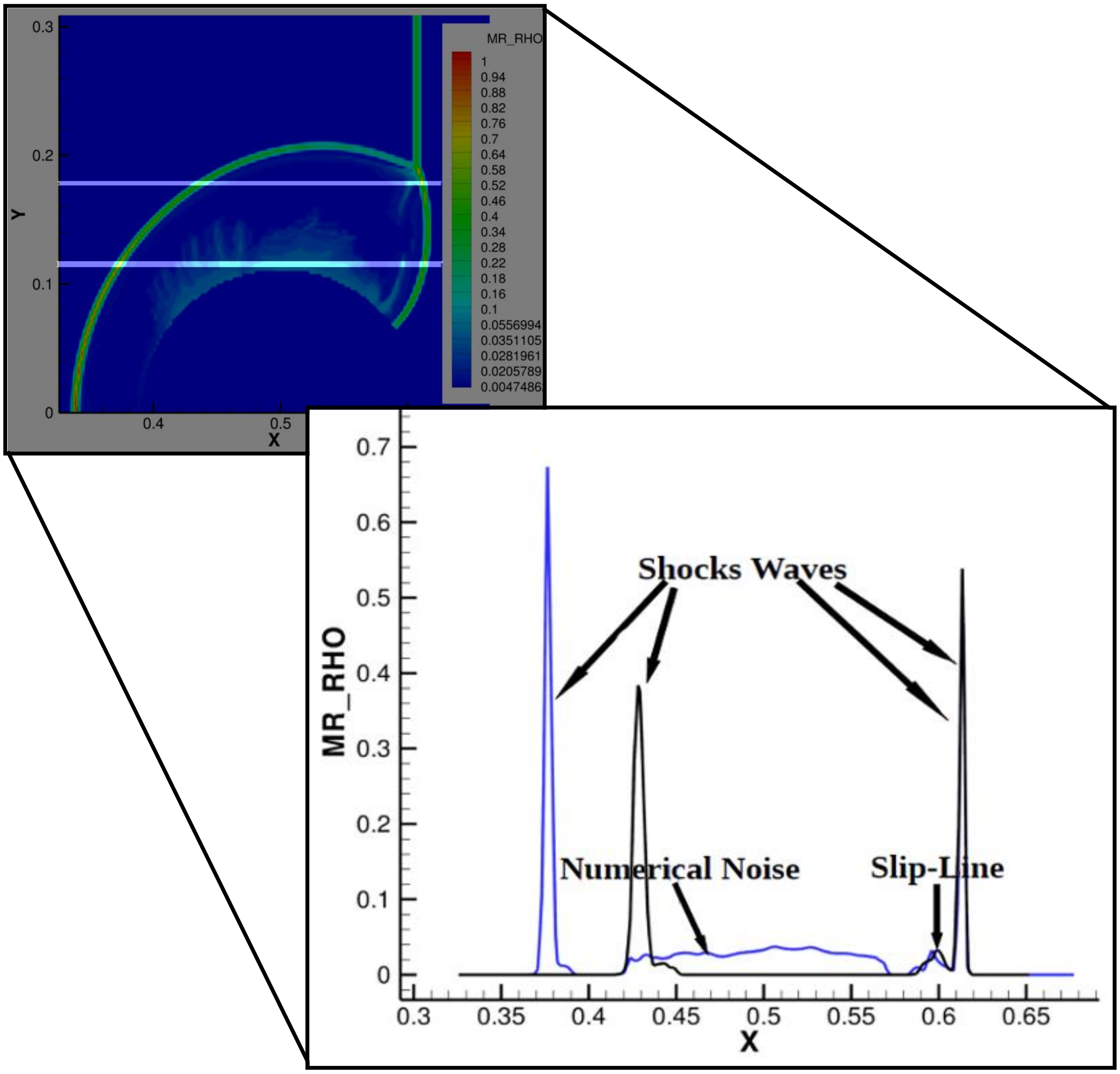


Figure 4. 8 **Slices of the Contour Plot in Figure 4. 5 at Y = .1625 (blue; first peak and noise) and Y = .1125 (black, second peak and slip-line).**

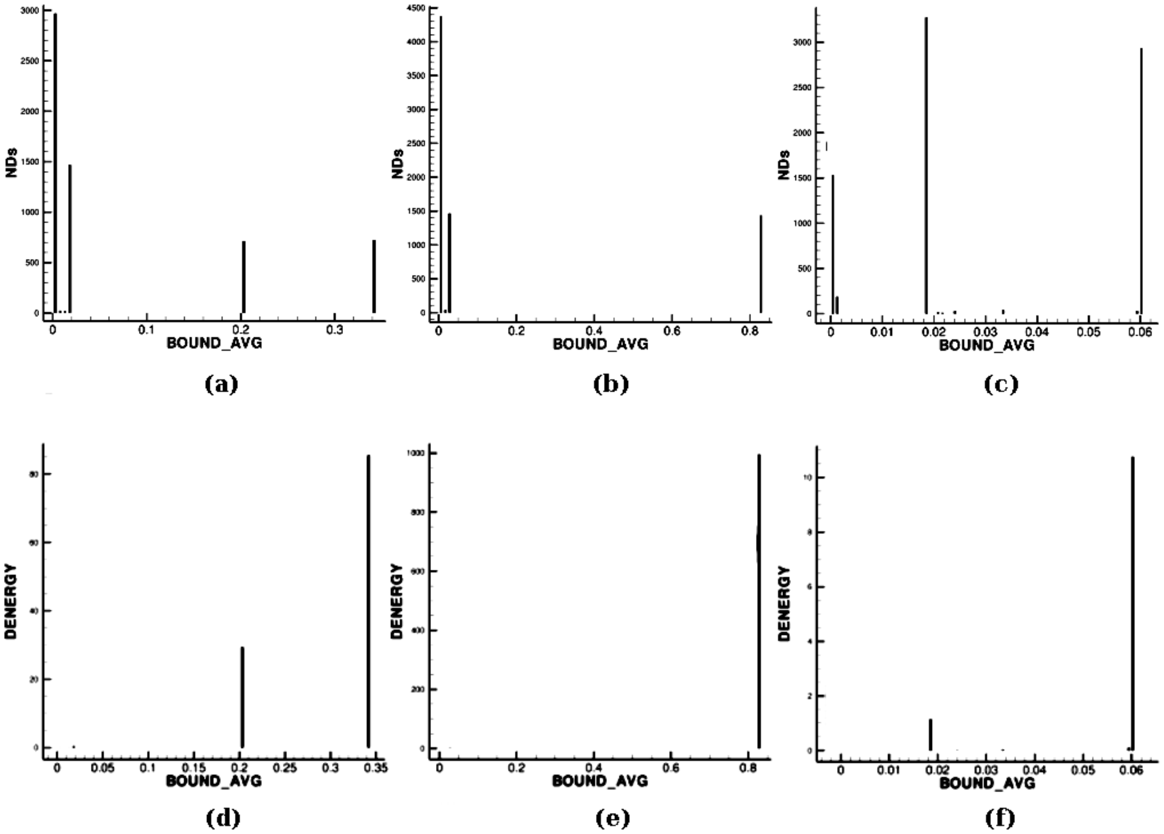
Figure 4. 8 shows two one dimensional slices of the finest level contour across lines of constant y near the cylinder and Mach stem. At any cell size the slip-line is clearly of an entirely different magnitude than the shocks. It was further found that the presence of the slip-line at larger cell sizes was nearly completely masked by numerical noise in the region of the cylinder (refer to the above). Both of these density field characteristics further complicate weak feature detection as discussed in the next section. It is possible to use another flow field variable in order to assist in the capture of the slip-lines while avoiding the numerical noise found in the density solution.

The time dependence of the detail coefficients is of great importance to the evolution of the flowfield, and cannot be overlooked when dealing with moving frames. If a feature is coarsened to a scale lower than its prominent scale, a numerical diffusion of the shock will take place, and the feature will most likely never return to its correct scale or be completely coarsened away in the future. An investigation into the time dependency of the detail coefficients of density was consequently preformed for the cylinder test case at a Mach number of 2.81. Histograms of the detail coefficients captured at two time-steps, before and after the incident shock impacted the cylinder are shown in Figure 4. 9 and Figure 4. 10.

As expected, in the time-steps before the collision, the detail coefficients exhibit very few major amplitude components at all levels as seen in the histograms below. Notice that all of the detail coefficient energy, calculated as:

$$e = \sum_{j=1}^{N} | d_j^k |^2 \qquad \text{Eqn 4. 1}$$

is contained in one or two bands on all levels of transform when only the shock is present.



(a) (b) (c)

(d) (e) (f)

Figure 4. 9 **Detail Coefficient Magnitude Spectrum Before Shock Collision (TS 30): Number of details wrt size (a) Level 1, (b) Level 2, (c) Level 3; Detail coefficient energy (c) Level 1, (d) Level 2, (e) Level 3.**

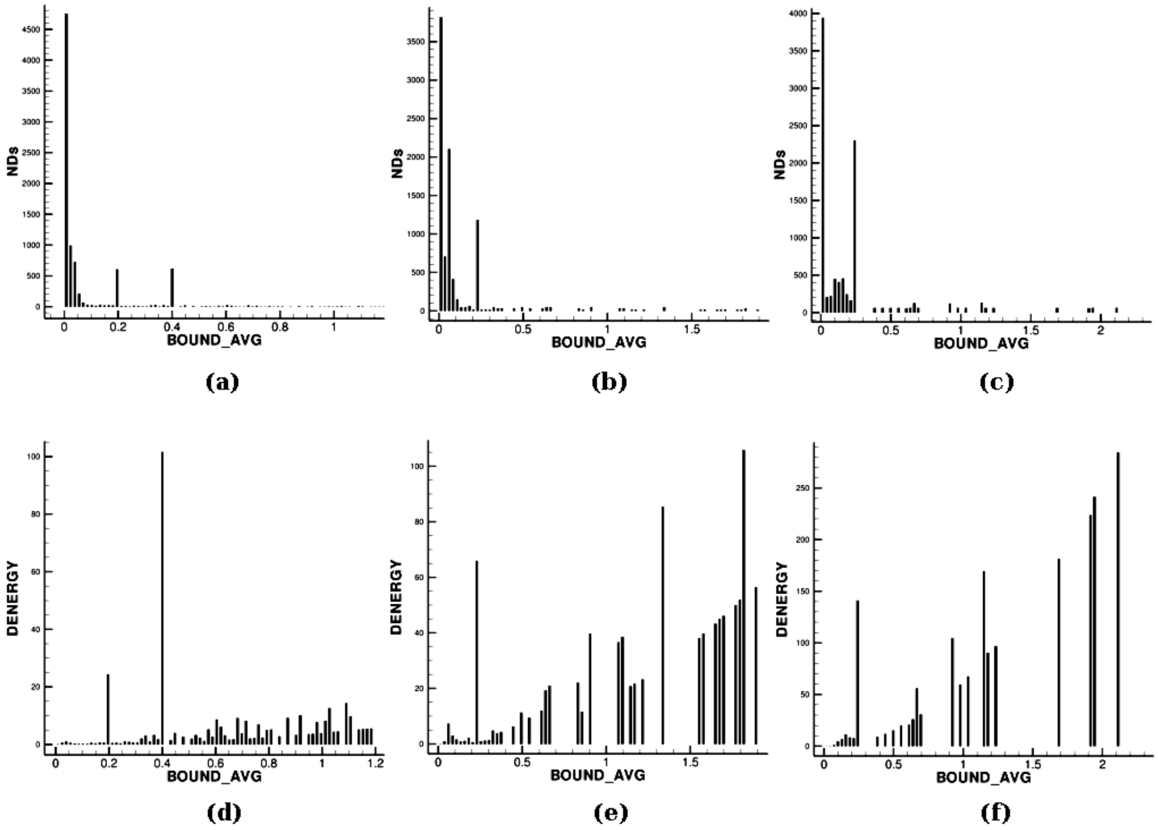


Figure 4. 10 **Detail Coefficient Magnitude Spectrum After Shock Collision (TS 240): Number of details wrt size (a) Level 1, (b) Level 2, (c) Level 3; Detail coefficient energy (c) Level 1, (d) Level 2, (e) Level 3.**

As the shock is dispersed into multiple regions after impact with the cylinder, the the amplitude components of detail coefficient energy, slowly scatter (Figure 4. 10). This is due to the curvature of the bow shock (which creates Gibbs oscillations of larger magnitude than those of a strictly horizontal or vertical shock [4]), as well as the addition of numerical noise, and weakening of the incident sock. It should be noted that the detail coefficients do not reflect a Gaussian distribution as noted in [28] (Again, further complicating the determination of a statistically designed thresholding algorithm).

The maximum detail coefficients also seem to follow a downward trend over time after the shocks impact with the cylinder. Figure 4. 11 describes the behavior of the maximum detail coefficient over time at varying Mach numbers.

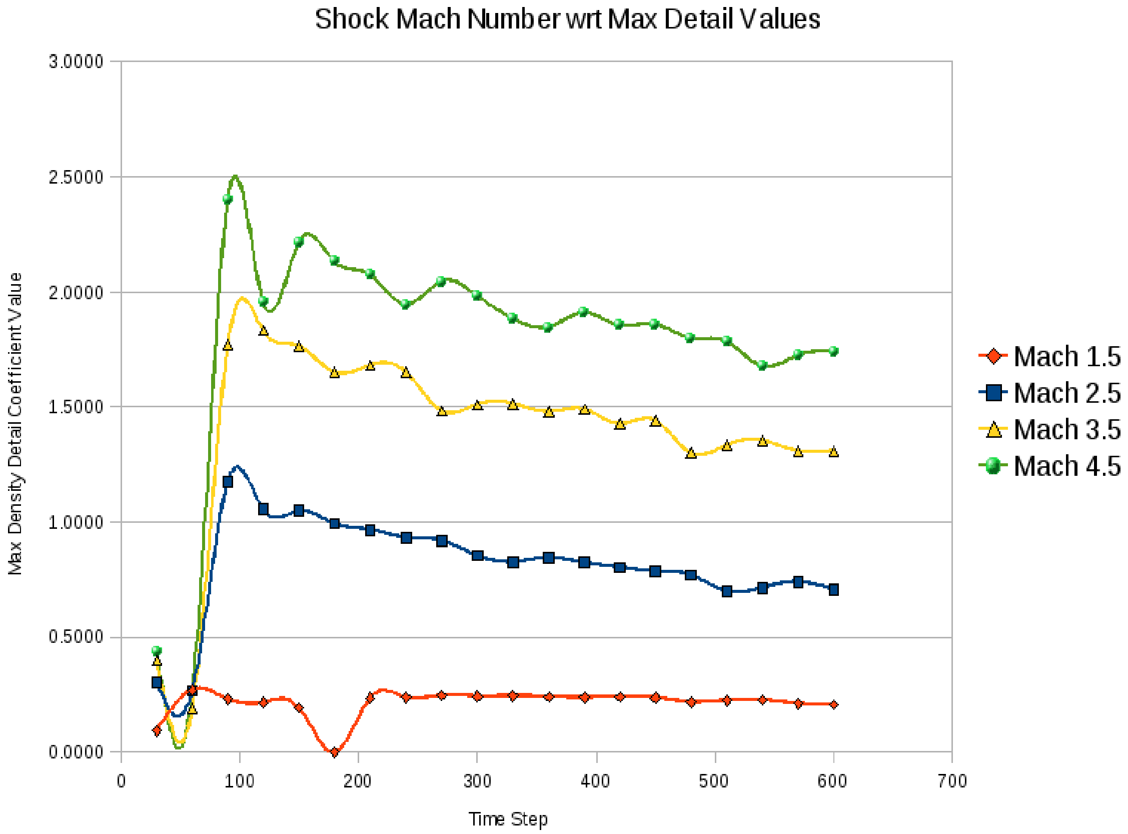


Figure 4. 11 Maximum Detail Coefficient Magnitudes wrt Time Step.

4.4 Generalized Selection of Truncation/Tresholding Parameters

There have been many studies regarding the selection of detail coefficient thresholding parameters for wavelets due to their excellent signal denoising and compression abilities. A majority of these methods, such as those presented in [14, 16, 28, 29] use simple statistical data concerning the detail coefficient distribution on the present level of transform *k-n*. The advantage of such methods is that the threshold may be set without user intervention, and with respect to only the current time-steps detail coefficients. Other advantages of this type of thresholding method are that the features are not as easily lost during the cell averaging oscillations described, and that new features being slowly introduced into the flow field should be better detected. The resulting mesh refinement of the two test cases presented above, using the current gradient based method, static user defined threshold parameters and automatic level specific TAWS, Two-pass Average/Outlier (TAO), and multivariable average (MA) pass thresholding algorithms are presented below.

The best achieved gradient-based refinement for the two test cases can be seen below for the current gradient based refinement scheme. The gradient based refinement presented may not be optimal, however, the “efficient” ranges of the gradient thresholding parameters are very large and multiple values within those described in [3] were attempted. The refinement parameters set for the case presented in Figure 4. 12 slightly over refined in the area around the cylinder, yet, the slip-lines were still not completely resolved. The number of cells generated in time-step 540 is 38367, and at time-step 1140 is 40058.

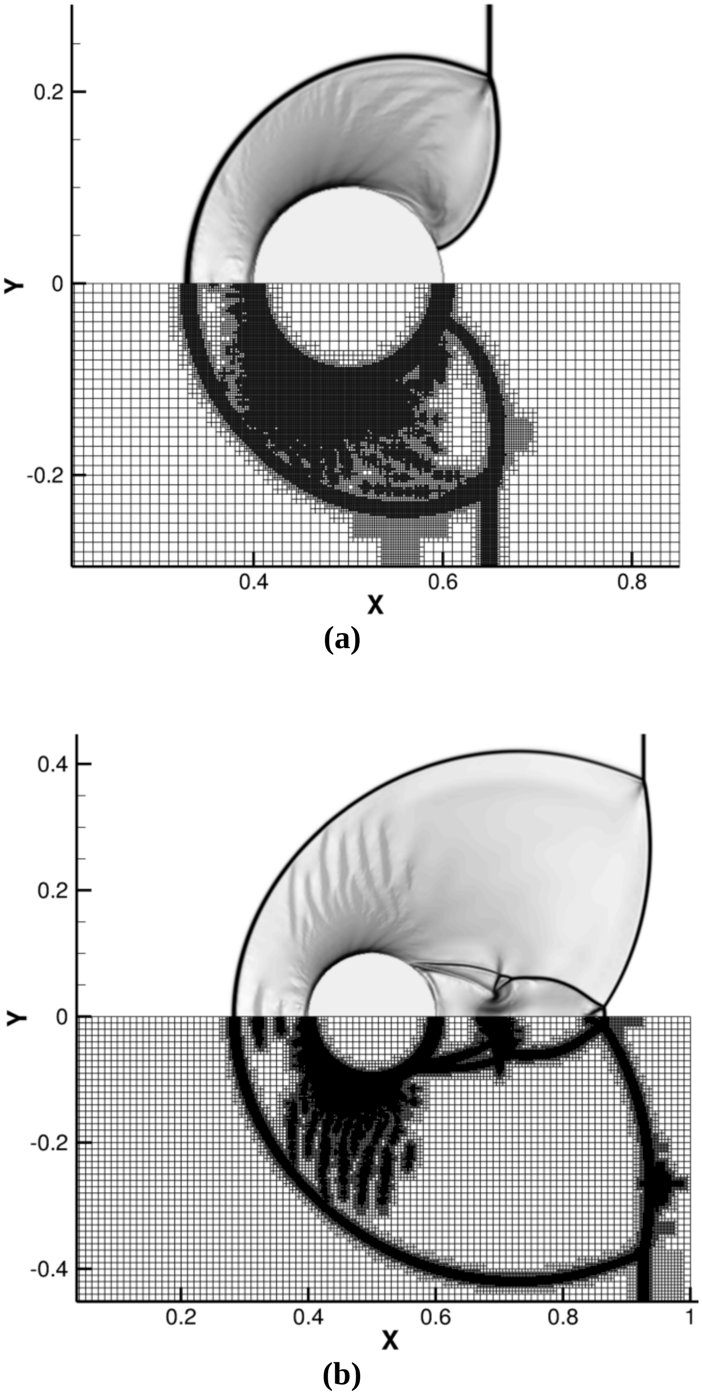
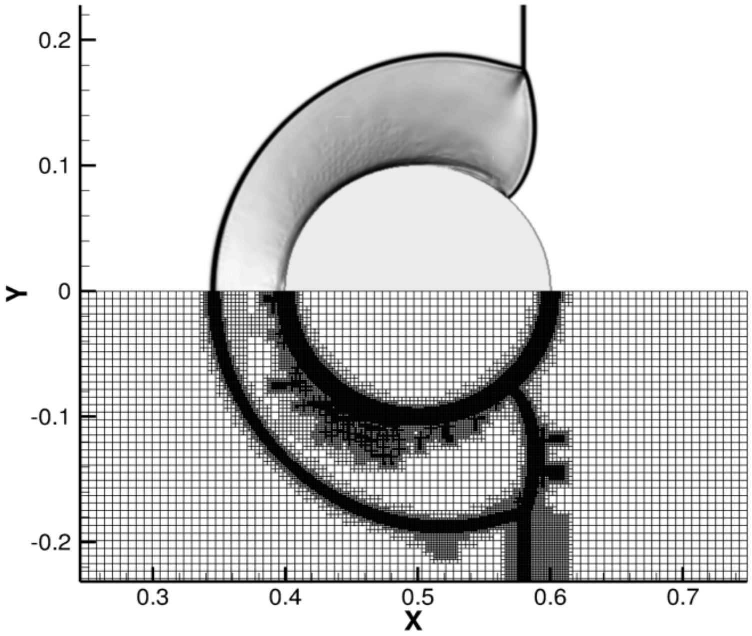
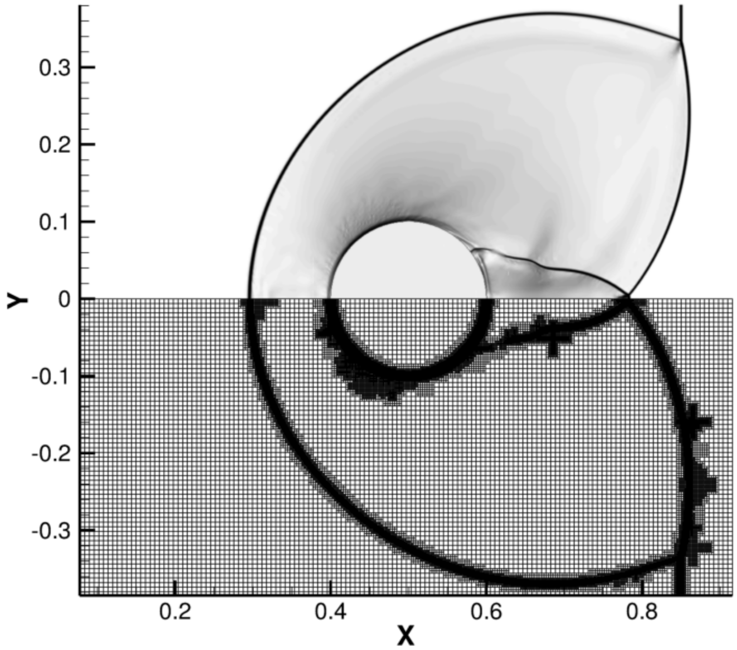


Figure 4. 12 Best Achieved Gradient-Based Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.01; (a) Time-step 540, (b) Time-step 1140.

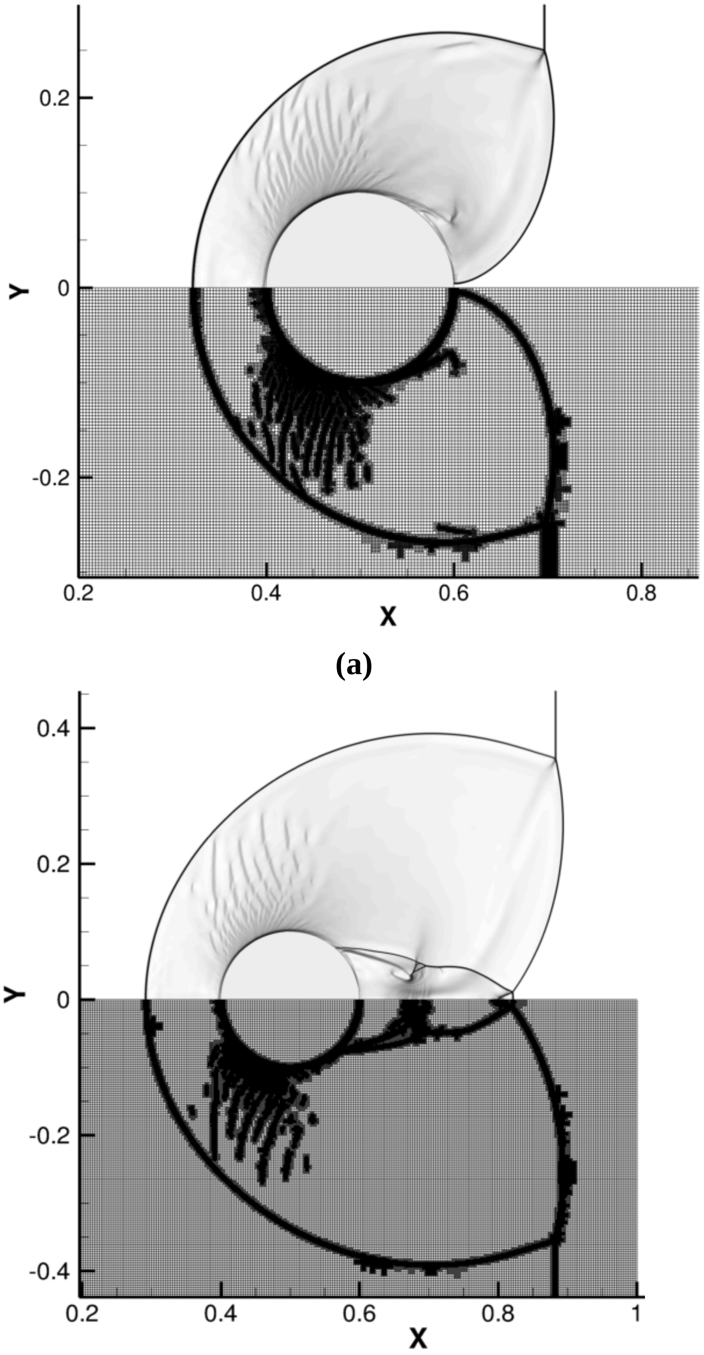


(a)



(b)

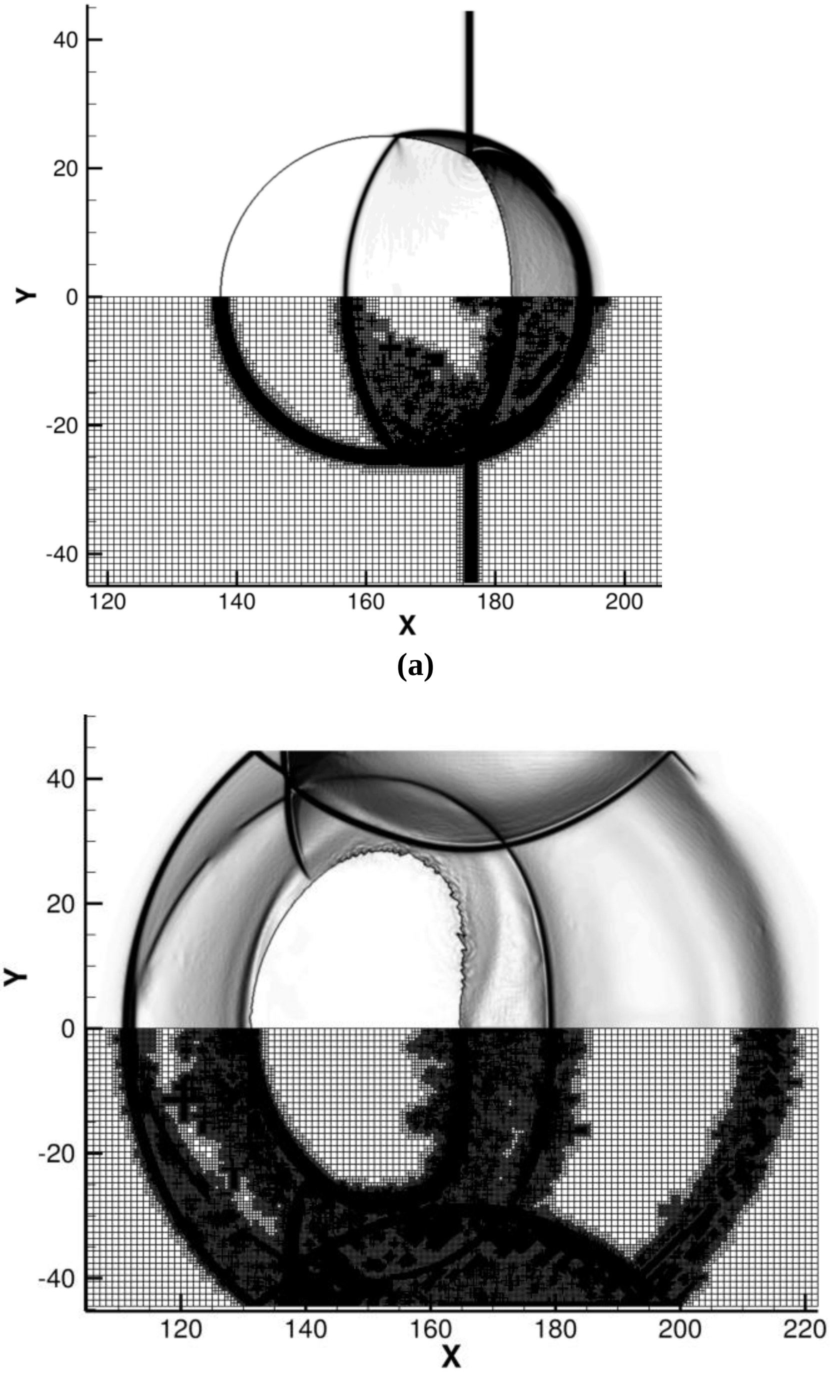
Figure 4. 13 **Best Achieved Gradient-based Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.006; (a) Time-step 600, (b) Time-step 1680.**



(a)

(b)

Figure 4. 14 Best Achieved Gradient-based Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.0033; (a) Time-step 600, (b) Time-step.



(a)

(b)

Figure 4. 15 Best Achieved Gradient-based Refinement and Numerical Schlieren Image of the Bubble Test Case: (a) Time-Step 770 (b) Time-step 2760.

The refinement provided in the Figure 4. 13 provided too little refinement of the weaker structures in the flow field, including the wake. The wake structure must be well resolved in order to properly estimate the drag force on objects in the flow field. The well resolved grid case in Figure 4. 14 displayed the same behavior as in the other two cases. Too much refinement was allocated to areas of meaning less noise inside the bow shock. The bubble test case, shown in Figure 4. 15, was highly over-refined in areas of little consequence to the solution due to large amounts of numerical noise present in the solution. At time-step 770 there already exists 64098 cells, and by time-step 2760 this number has ballooned to 157042 cells. Small changes in either one of the refinement parameters δ_1 or δ_2 resulted in a fairly drastic change in the obtained refinement.

The inability to obtain efficient and meaningful refinement of key features in the flow-field (after multiple attempts with different δ_1 and δ_2 parameters), without dramatically over refining the mesh, while using gradient-based refinement scheme shows it is not optimal and even hard to predict at times. It is clear that the structured segmentation of a gradient-like parameter would be very useful in decomposing the flow field function so that a more decisive refinement criterion could be found without difficulty. The ENO-MRT scheme provides just such a functionality.

As a first trial, mesh refinement behavior was computed using static user defined cutoffs of 1E-10, .005, .01, .025, and .05 for a density only reference. The resulting mesh refinements are shown below in Figure 4. 16 and Figure 4. 17.

Due to the lower Mach number, and hence smaller overall discontinuity, the bubble test case fails to pick up the reflected shock at a much smaller cutoff than in the cylinder test case. The same trends were seen when simulating the shocked cylinder at varying Mach numbers as well, and should be expected.

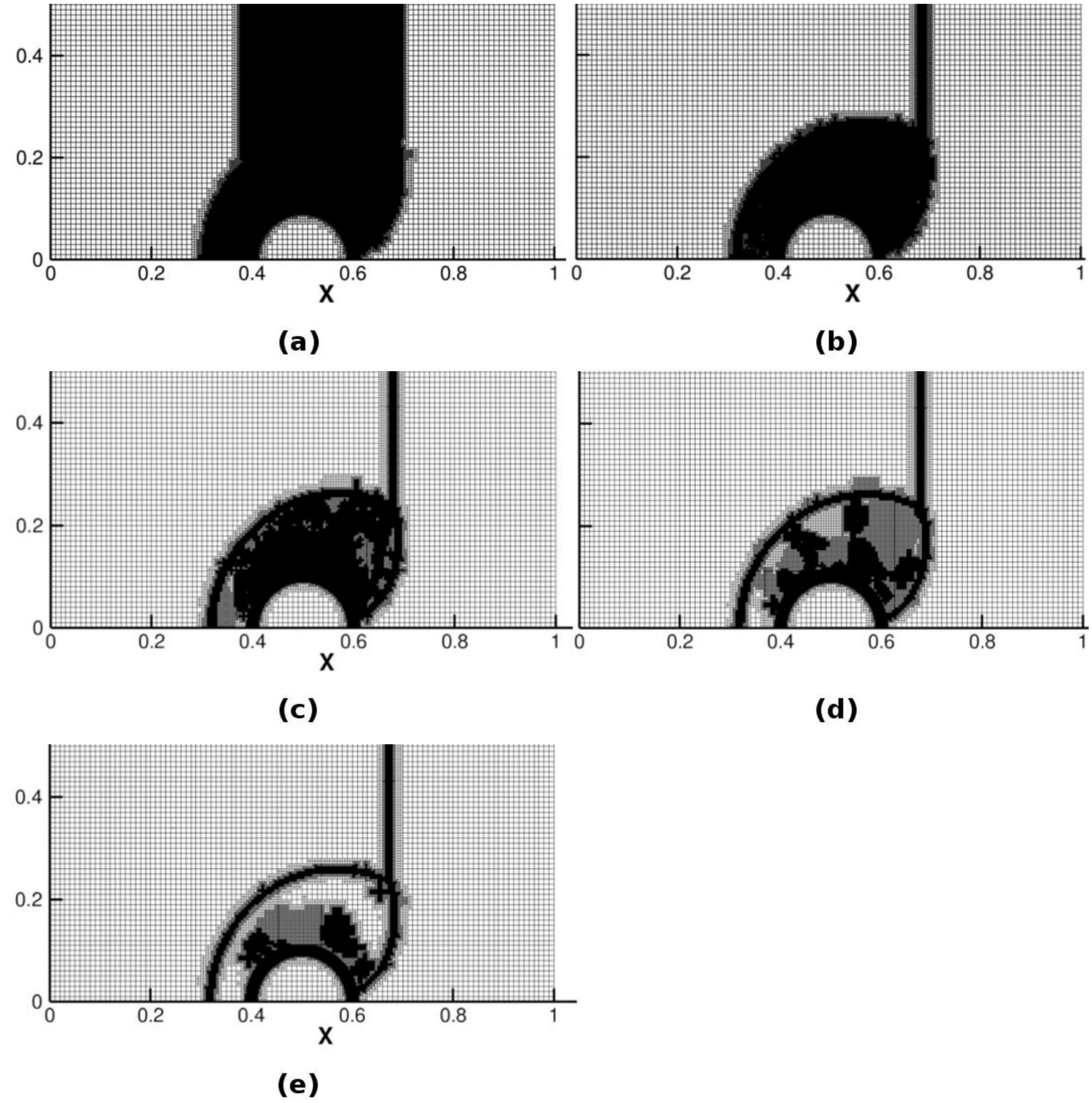
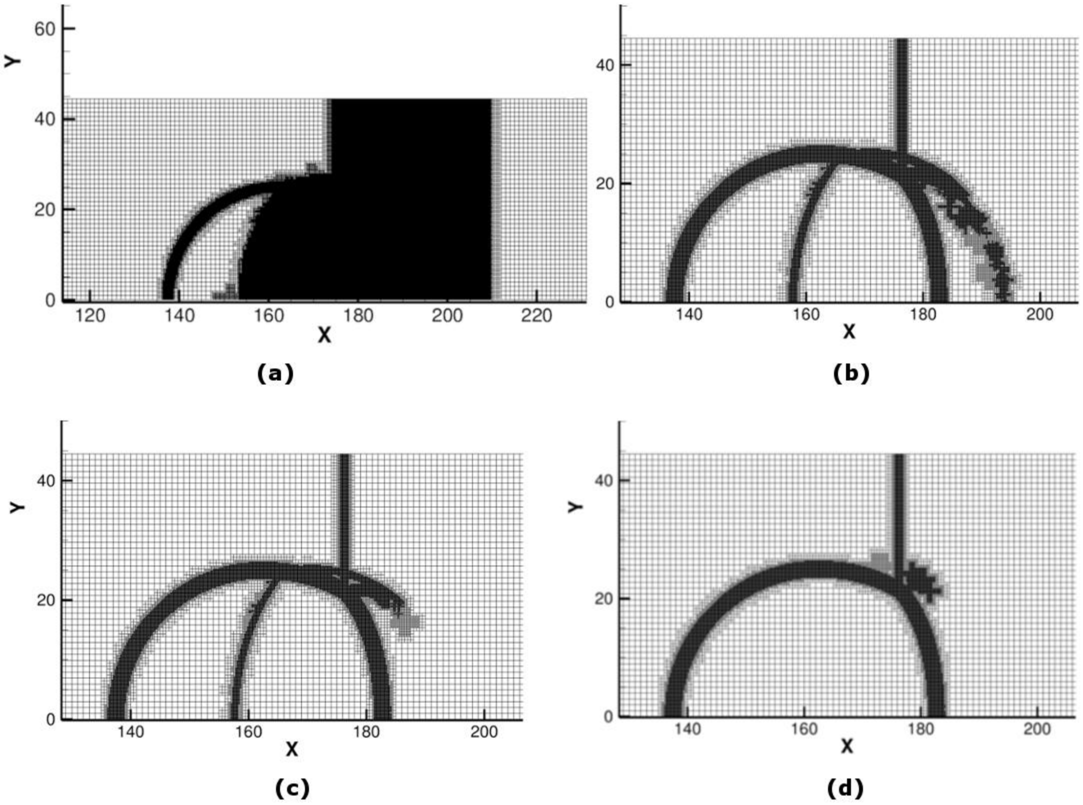


Figure 4. 16 Mesh Refinement of Cylinder Test Case, Manual Thresholding: (a) ε = 1E-10, (b) ε = 0.005, (c) ε = 0.01, (d) ε = 0.025, (e) ε = 0.05



(a) (b)

(c) (d)

Figure 4. 17 Mesh Refinement of the Bubble Test Case, Manual Thresholding: (a) $\varepsilon = 1\text{E-}10$, (b) $\varepsilon = 0.005$, (c) $\varepsilon = 0.01$, (d) $\varepsilon = 0.025$

It should be noted that when using the density as a refinement reference parameter the change in density across interfaces will also change the size of the detail coefficient produced. In this case the density of the helium bubble is far less than that of the air. As noted in [26] the detail magnitude is dependent on both the smoothness of *f(x)* and the size of the magnitude of the jump condition. The normal shock equations show that the amount of discontinuity expected is related to γ (the heat capacity ratio) of the specific medium. Therefore, unless the density detail coefficients are corrected with respect to the γ contained in the current levelset the details will become “irrelevant”, and will be lost.

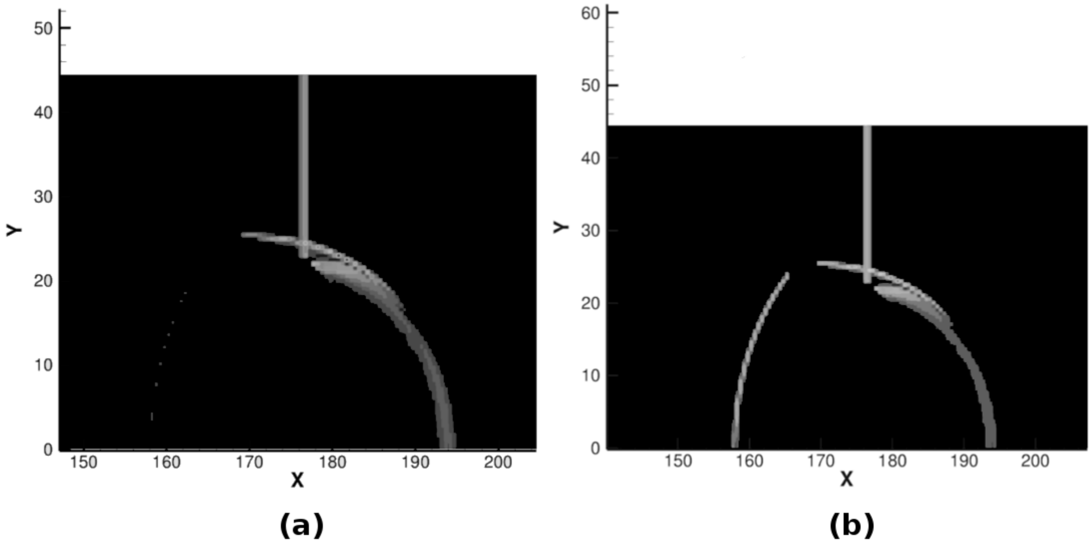
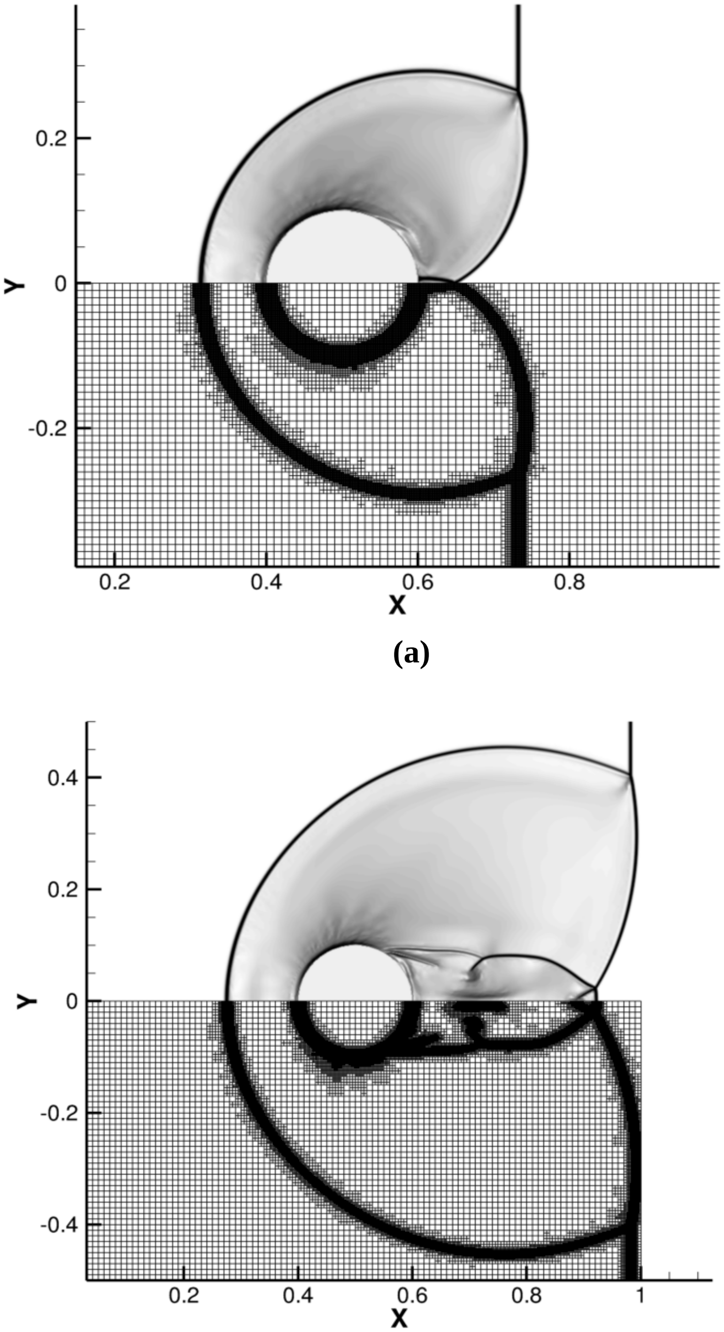


Figure 4. 18 Detail Coefficients of Density: (a) Before normalization with γ_{ls}, (b) After normalization with γ_{ls}.

The static truncation cases presented in Figure 4. 16 and Figure 4. 17 clearly show that attempting to set a user defined threshold might result in choppy and unpredictable refinement. This may be attributed to the translation of structures through the computational mesh and the cell averaging constraints continually creating fluctuations in the detail coefficients based on the CFL number. Even if this problem was somehow overcome, the user would still be required to find a Mach dependent detail coefficient size and set a threshold such that the numerical noise found in the solution would not be captured as well.

Figure 4. 19 and Figure 4. 20 show the cylinder and bubble mesh refinement using the TAWS algorithm in order to select "important" detail parameters.



(a)

(b)

Figure 4. 19 TAWS Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.0.1; (a) Time-step 720, (b) Time-step 1260.

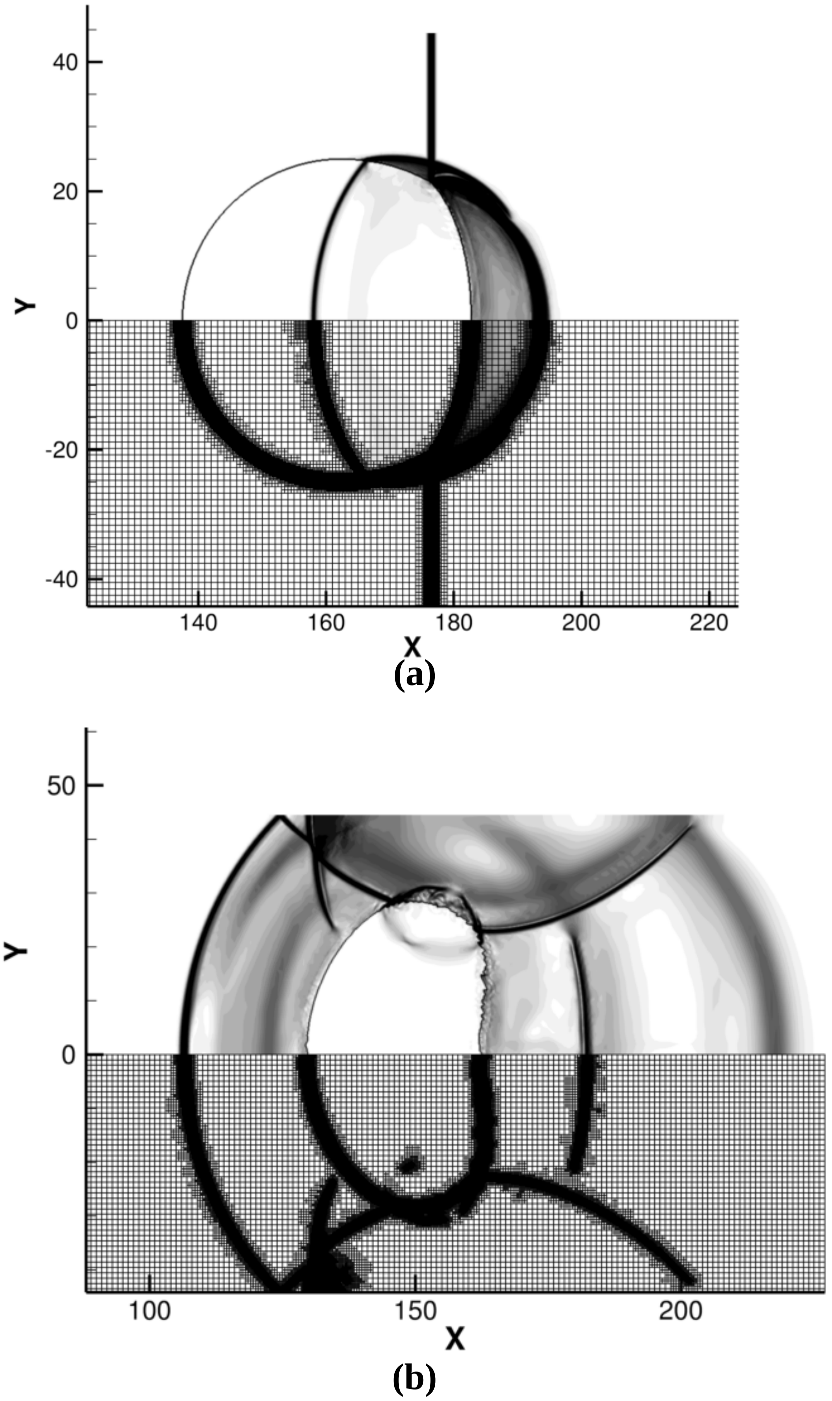


Figure 4. 20 TAWS Mesh Refinement and Numerical Schlieren Image of the Bubble Test Case: (a) Time-step 744, (b) Time-step 3000.

The TAWS algorithm performed very well in capturing all of the shock geometry smoothly, without high levels of noise, and without any guesswork or human intervention. The gradient based refinement case displays some high frequency noise near active regions of the flow field. Filtration of noisy data from the mesh refinement solution results in 12606 and 16983 fewer cells forming using the TAWS algorithm for refinement in the cylinder and bubble cases respectively (at a non dimensional simulation time of 7.939E-2). The occupancy ratio is lower in the MRT based methods for the TAWS cases 11.98 to 8.05 (gradient and TAWS respectively). The Occupancy Ratio (OR) is a measure of how many cells have been divided, and is calculated as follows [3]:

$$OR = \frac{ncells_{LMR}}{ncells_{BASE}} \times 100 \qquad \text{Eqn 4. 2}$$

Higher occupancy ratio means that the cells that are refined in the flowfield are scattered over a larger region, which may indicate a large active structure, but is usually associated with the presence of a scattered region of noise (as viewed in the bubble case). It should also be noted that the number of refined cells grows significantly in the later stages of the flow due to the formation of other major shock and wake region structures.

The ability of the statistically truncated ENO-MRT to smoothly capture the shock regions of two entirely different numerical test cases, with different geometries and Mach numbers show the robustness of this type of method. In order to further demonstrate the effectiveness of statistically based thresholding methods in general the cylinder simulation was also computed for a range of Mach numbers and cell sizes using the TAWS algorithm.

The cylinder test case was repeated at a Mach number of 2.81 and a cell size of 0.0033, and at a Mach number of 4.22 and a cell size of 0.006 in order to further asses the robustness of the TAWS detail selection method. The changes in cell size and Mach number did not reveal any degradation to this methods ability to cleanly capture prominent features in the domain.

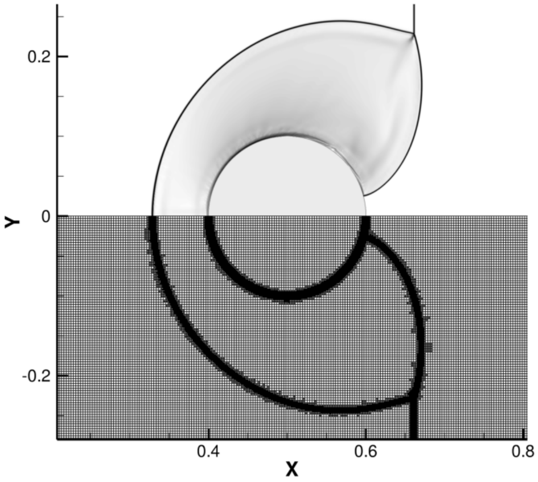


Figure 4. 21 **TAWS Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.0033, Time-step 2000.**

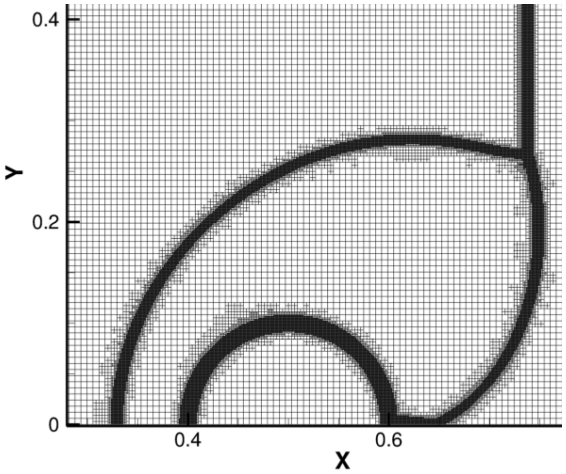
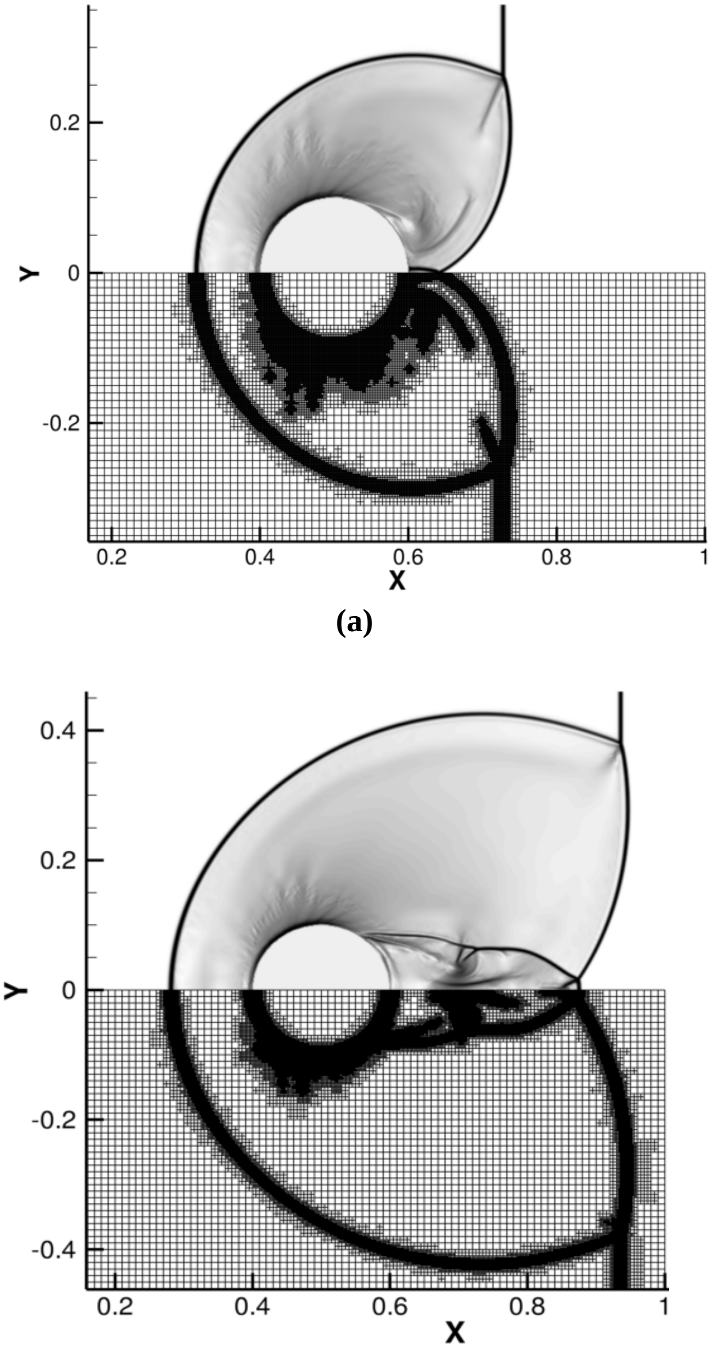


Figure 4. 22 **TAWS Mesh Refinement of the Cylinder Test Case: Ma = 4.22, dx = 0.006, Time-step 1200.**

Although the shocks and strong features were well captured by the TAWS method, one can see that the weaker features, such as the slip-lines, were not detected because the strength of the detail coefficients in the regions of the shock increases the standard deviation significantly, thereby causing the threshold to increase as well. The problem with lowering the threshold becomes that at larger cell sizes, the magnitude of the noise in the density solution is roughly equal to that of the slip-lines and other more subtle features (as shown in Figure 4. 8). The need arises then to perhaps consider a multiple amplitude selection method which first sorts out the large detail coefficients, and then filters the lower amplitude coefficients in order to avoid the lower amplitude Gibbs oscillations and noise from being included in the threshold limit. One such method is presented in [28].

A Two-pass Average-Outlier (TAO) method will therefore be attempted for filtering out both the shock while still maintaining a fair amount of information regarding the weaker features in the flowfield (if any are present). As the name suggests the method uses two passes, while filtering the detail coefficients for refinement. Using the average value of the detail coefficients as truncation parameters allows the first pass to capture the regions of the flow which contain shocks and other very strong features. Once these features have been identified, they are then ignored in the calculation of the second pass. The outlier method is then used to process any information left outside of a certain number of standard deviations (in this case 2, assuming a close to zero mean variance).

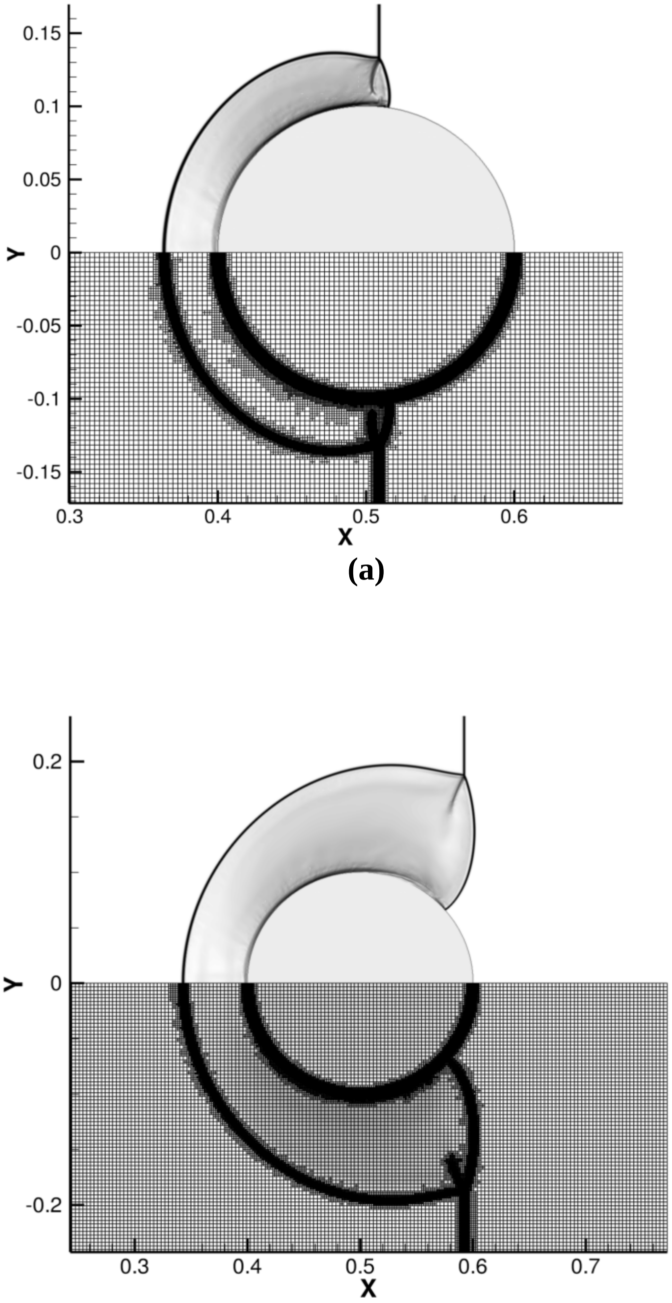
The TAO method works decently because, the dominant structures in the presented test cases will always be the shocks themselves. As shown in Figure 4. 23 the slip-lines are slightly more developed initially, but soon fade. This is because the density detail coefficient average is falling over time (refer back to Figure 4. 11), which significantly



(a)

(b)

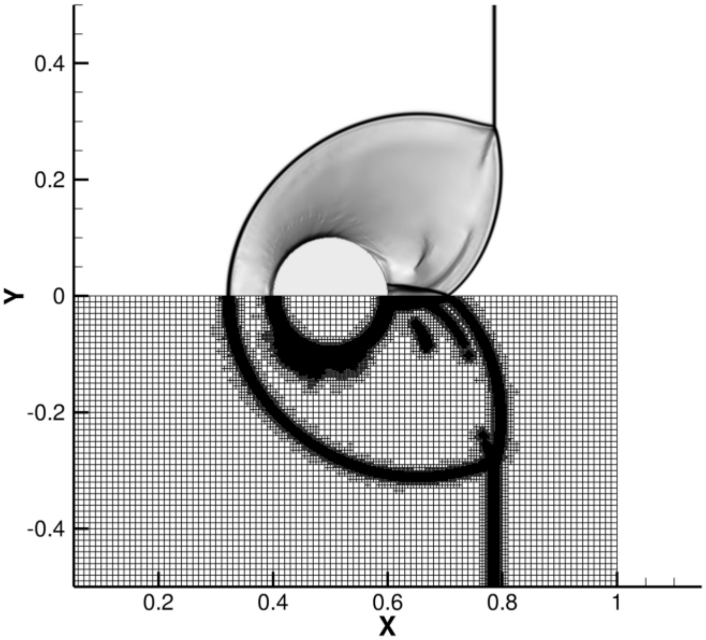
Figure 4. 23 TAO Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.01; (a) Time-step 710, (b) Time-step 1170.



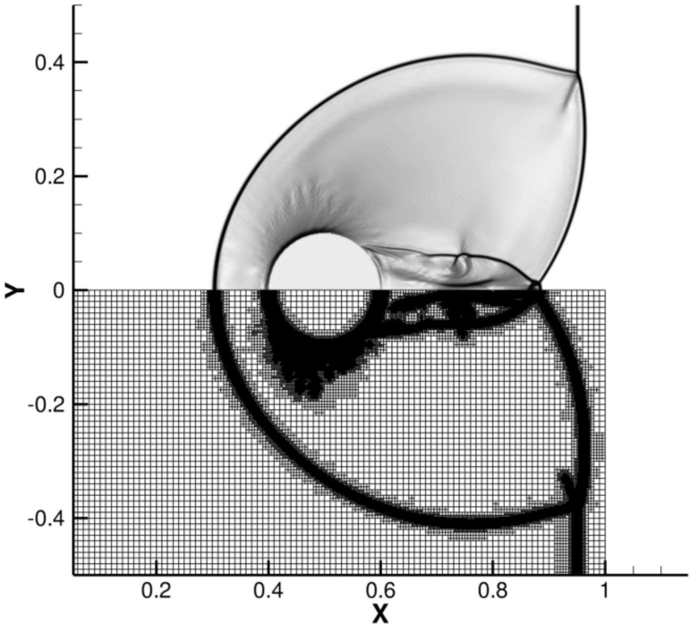
(a)

(b)

Figure 4. 24 **TAO Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.0033; (a) Time-step 720, (b) Time-step 3140.**



(a)

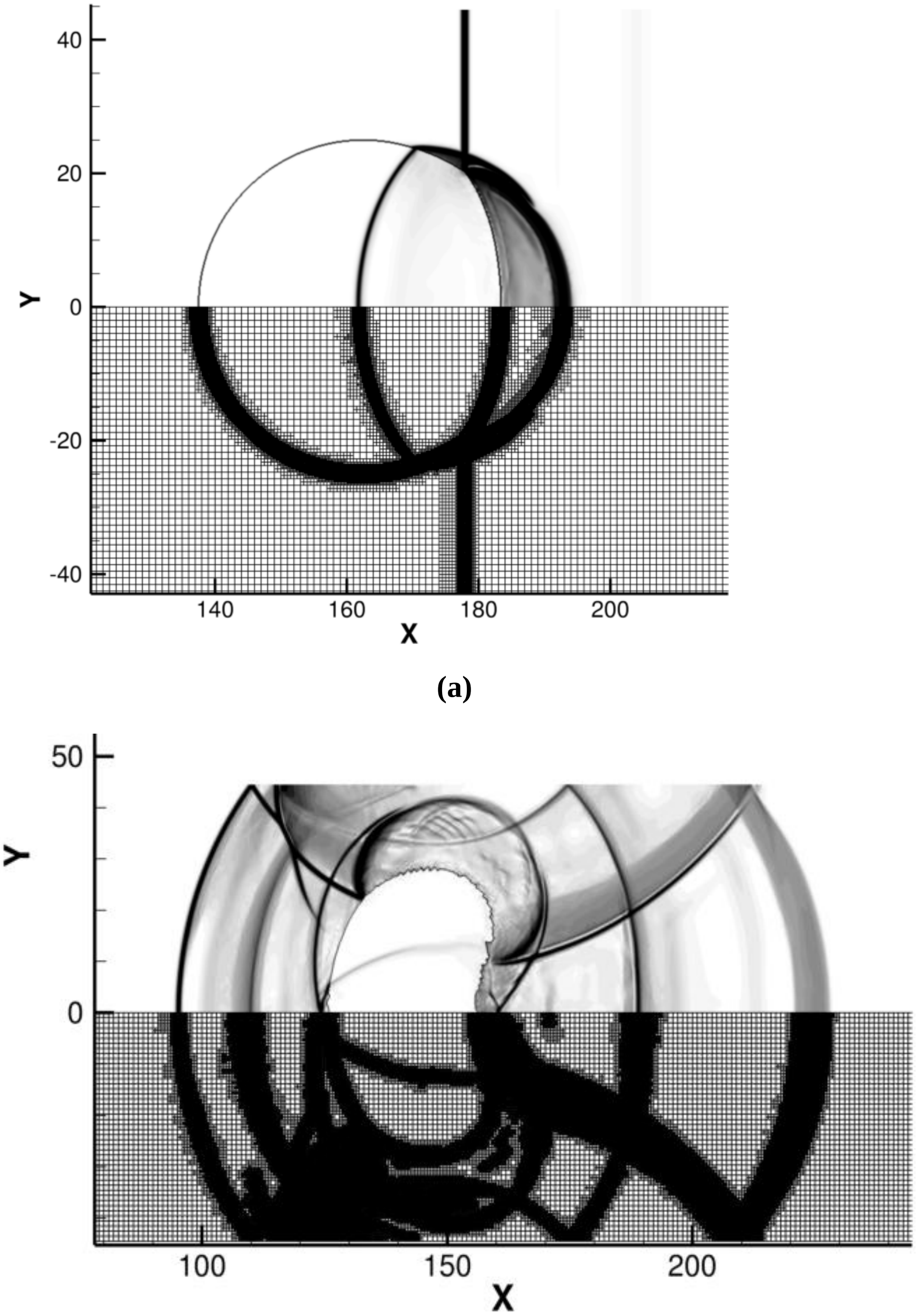


(b)

Figure 4. 25 TAO Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 4.22, dx = 0.006; (a) Time-step 840, (b) Time-step 1200.

increases the number of detail coefficients captured on the first pass. If the first pass captures most of the important features along with the upper bound of the weaker features the variance will be smaller, and only return values within two standard deviations (because none of the statistically outlying points were left for the outlier method to capture on the second pass). The variance dependence also causes the second pass to be less effective when using a finer mesh, because the noise near the cylinder reduces significantly. The noise reduction thus causes the average to capture the majority of the detail coefficients in the first pass, and again the second pass disregards the majority of the left over detail coefficients. In the case of the coarsest base mesh cylinder test case, the TAO method reduces the number of cells spawned from 38367 cells to 33169 cells (a reduction of 5198 cells). The occupancy ratio again decreased from the gradient based method from 11.98 to 10.37 (this is due to the refinement of the long finger-like structures in the gradient based refinement case).

The TAO method, however, seems to work well for the bubble test case. Figure 4. 26 depicts the refinement obtained by this method. The bubble case could benefit from the TAO approach because it contains many small fluctuations due to the fluid-fluid interactions which take place. These small fluctuations create a background noise for the second pass, as there exist many small and possibly relevant detail coefficients. The smaller fluctuations can also be seen in the gradient based case as large quantities of noise in the regions between the bubble interface and shock structures.) The fluctuations allow for the second pass to capture a significant amount of what was left behind by the first pass. When compared to the gradient based case, this method provides a much cleaner refinement of the flowfield features. The number of cells is decreased by 19866 cells at an early time instance, and by 34933 cells at the later time instance.



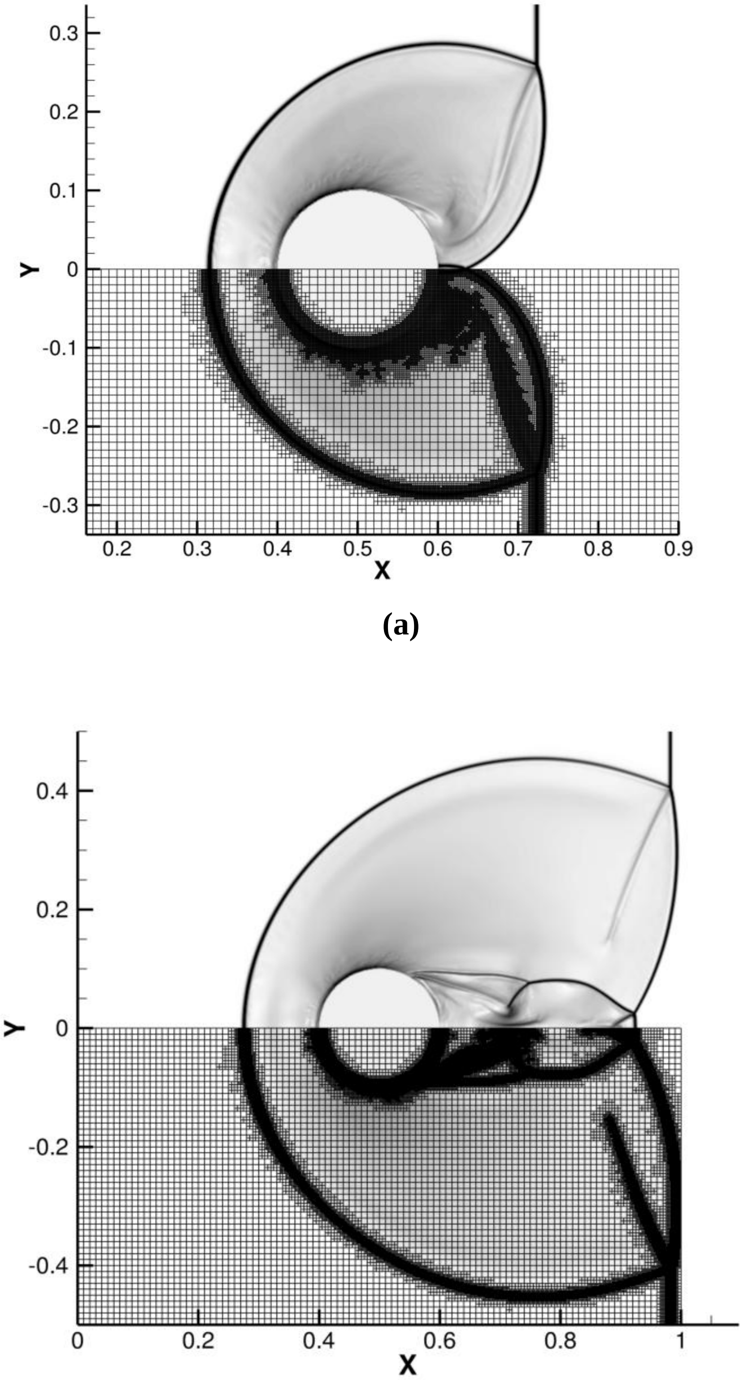
(a)

(b)

Figure 4. 26 TAO Mesh Refinement and Numerical Schlieren Image of the Bubble Test Case: (a) Time-step 677, (b) Time-step 3670.

Another possible way to capture features that remain unresolved in the density solution may be to perform a MRT on additional flow-field variables. As mentioned in Chapter 1, the current gradient based method uses the gradients of three flowfield variables in order to assist in the capture and refinement of features. Therefore, we may wish to use a combination of flowfield variables which accentuates the desired features in the domain for a given case. The addition of the vorticity to the ENO-MRT LMR scheme for the cylinder test case accentuates the slip-lines and wake regions because the detail coefficients of vorticity are high in these areas. The two variable refinement solutions are shown using the average density and average vorticity detail coefficients as the truncation parameter, and we will therefore abbreviate this method as a Multivariable Average (MA) method.

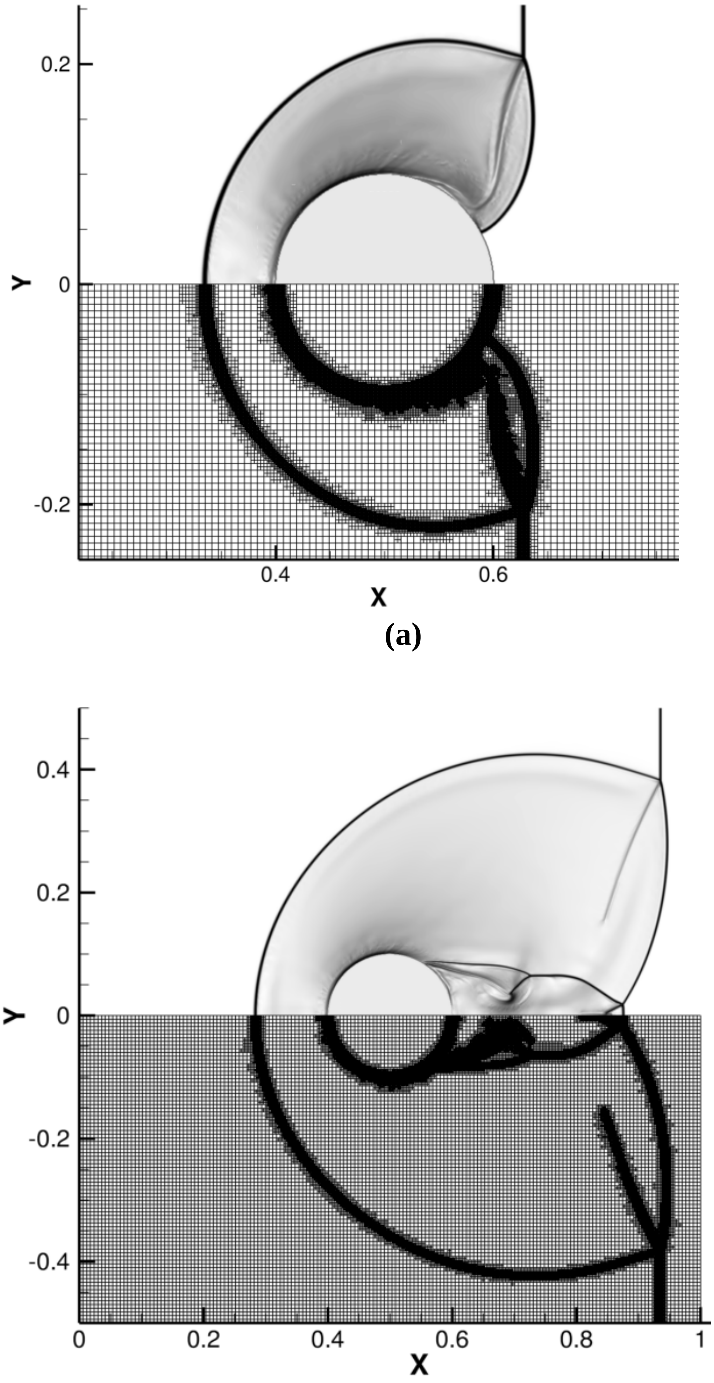
Figure 4. 27 displays the mesh refinement associated with the multivariable refinement scheme at a cell size of 0.01. The slip-lines are easily caught by the large vorticity detail coefficients in the area and the pressure solutions numerical noise is no longer an issue because there is no need to filter the slip-lines from the density detail coefficients. Trials at cell sizes of 0.006 and 0.0033 (below) were also conducted in order to further validate the resulting mesh refinement. Both methods performed very well, however, the slip-lines on the fine mesh break from the cylinder structure much sooner than for the other cases.



(a)

(b)

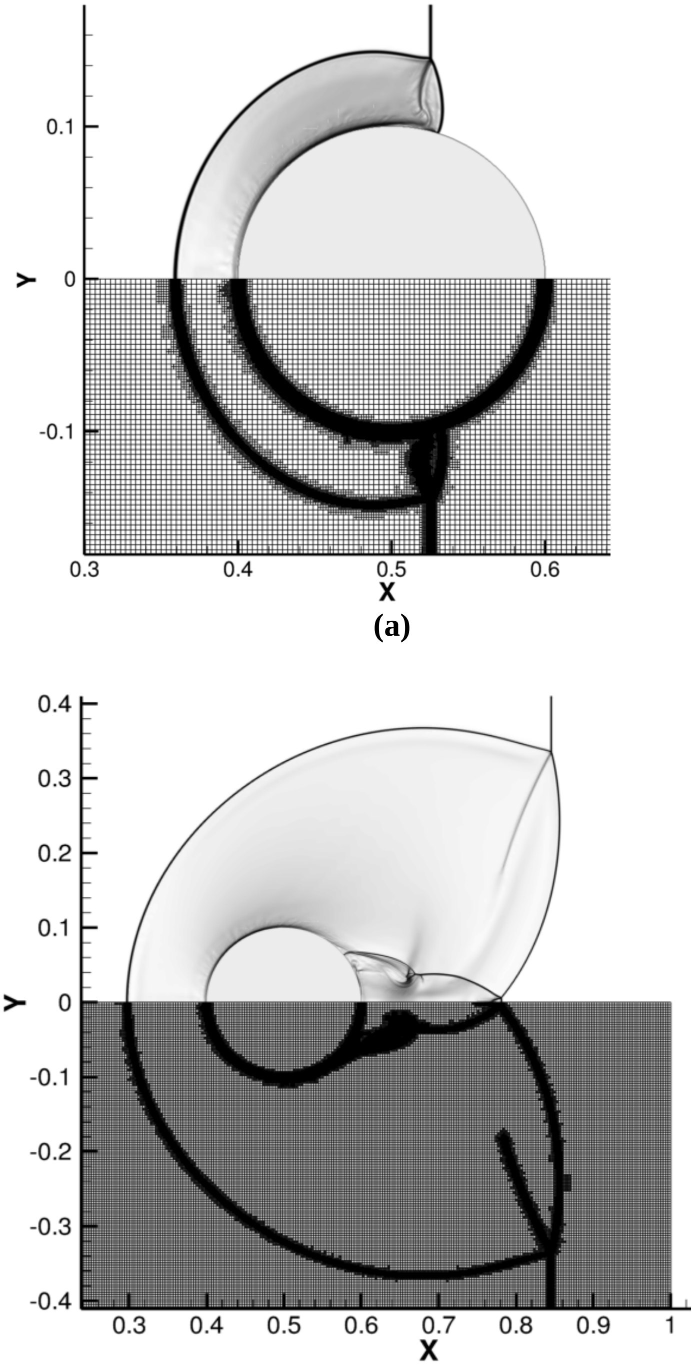
Figure 4. 27 MA Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx = 0.01; (a) Time-step 690, (b) Time-step 1260.



(a)

(b)

Figure 4. 28 MA Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx=0.006; (a) Time step 840, (b) Time step 2000.



(a)

(b)

Figure 4. 29 MA Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 2.81, dx=0.006; (a) Time step 840, (b) Time step 3140.

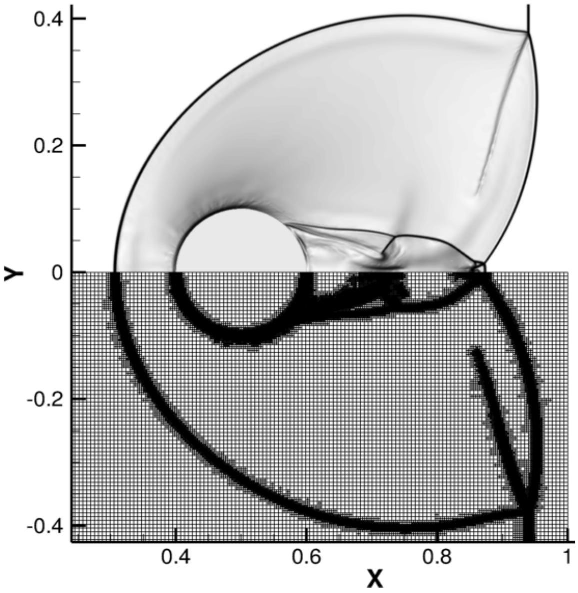


Figure 4. 30 **MA Mesh Refinement and Numerical Schlieren Image of the Cylinder Test Case: Ma = 4.20, dx=0.006.**

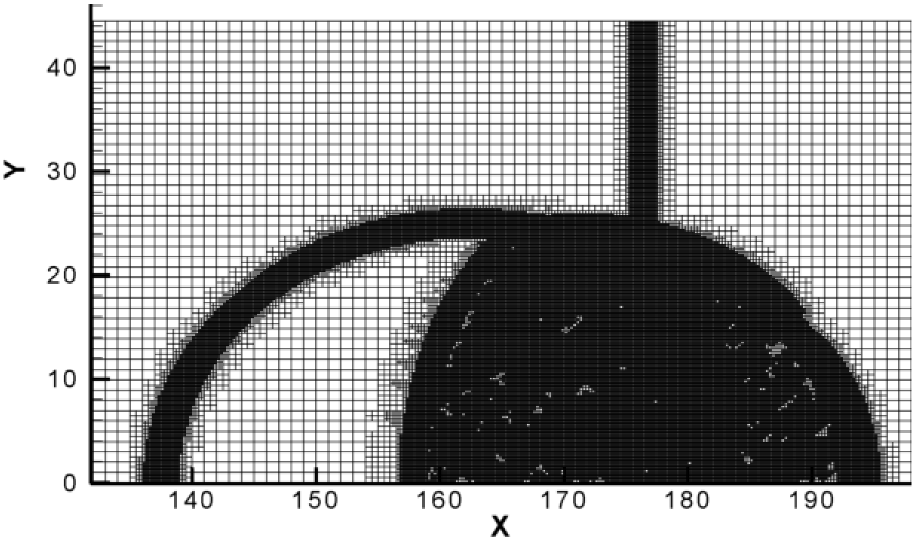


Figure 4. 31 **MA Mesh Refinement of the Bubble Test Case.**

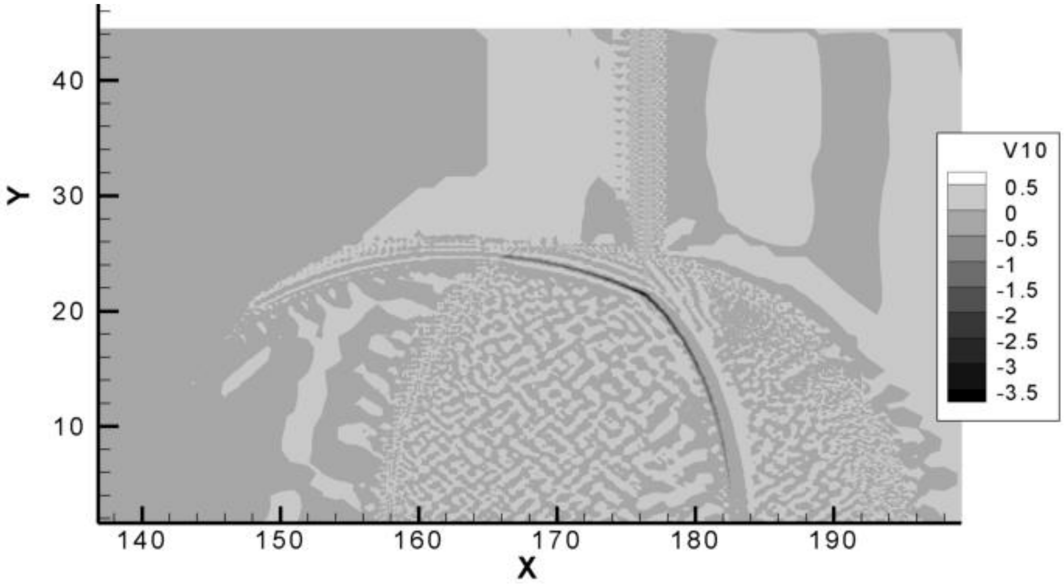


Figure 4. 32 Vorticity Contours of the Bubble Test Case.

Attempting to use the two variable density and vorticity scheme as described above for the bubble test case results in massive over refinement of the regions between the interface and the bow and transmitted shock waves. This is because there are many small vorticity perturbations in the flow-field due to small density and pressure changes occurring in these regions. It is therefore advised that another variable be used for a second parameter when dealing with fluid-fluid interactions. Figure 4. 31 and Figure 4. 32 show the over-refinement produced in this case as well as the vorticity contours obtained.

4.5 Determination of Feature Scale

The determination of the scale of an object is found by simply evaluating the magnitude of the detail coefficient. As stated in [12], a detail coefficient, d^k, at level k implies that locally the function or one of its derivatives changes rapidly”. Rapid change on the kth level likewise implies that the function is underrepresented by the sampling interval used n_k. We may take two approaches to determine if a feature has a detail coefficient large enough to warrant refinement, which of course depends on the users needed solution accuracy and computational resources.

4.5.1 Global Comparisons

Globally the presence of a detail coefficient implies, as stated above, that the feature is still discontinuous at the scale represented by the transform level. Global comparisons may be done by setting truncation parameters which have been affected by the entire domain, such as the mean, standard deviation, and max (the minimum detail coefficient is not a usable measure because in smooth areas it is always ~ 0). The methods used above are all of this type, and perhaps better refinement could be attained by using a local comparison.

4.5.2 Local Comparisons

Local comparisons could also be carried out. One may wish to capture weaker features, which are present at higher scales, but which have a small magnitude or relative strength. In order to do this a windowing method or other searching algorithm can be implemented in order to mark features based on a local evaluation of the detail coefficient magnitudes. Local comparisons tend to be computationally expensive, since windowed properties need to be recomputed upon a window shift and trace algorithms often require recursive algorithms and would introduce yet another threshold criterion. The wavelet-criterion and resolution-criterion defined in [30], and used in presented code are weak types of local detail comparison. In the cylinder test case these criteria provide enough support to keep the weaker bow shock region, 180 degrees from the shock propagation direction, from coarsening during TAWS thresholding.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORKS

5.1 Concluding Remarks

A detailed explanation of the cell-based ENO-MRT framework in 2D was presented in the above chapters, including detailed descriptions of the stencil selection procedure, the interpolation procedure, the information storage procedure, the resulting detail coefficient magnitudes, and the attempted threshold selection procedures.

The ability of the cell-based ENO-MRT scheme to easily identify strong gradients in one and two dimensional fields can be seen in the produced detail coefficient profiles throughout chapters 2 and 4. It was also noted that the detail coefficients produced in most wavelet and wavelet-like functional approximation techniques produce a distinctly non-Gaussian distribution of detail coefficients and that the use of ENO creates a nonlinearity in the detail coefficients which further deforms the detail coefficient distribution. The detail coefficient sizes found produced two separate magnitude regimes: a high and a low regime. The high regime is comprised of the shock structures and is fairly simple to filter out, and the low regime contains possible numerical noise and weaker features which we wish to distinguish from one another. The appearance of these two regimes is not unexpected. However, it presents yet another filtration problem, i.e. we most likely should consider these two magnitude regimes separately. (The TAO method was attempted in order to take into account these two separate regimes, but was found to be ineffective in filtering the low magnitude regime in the solid body case.) This then requires that a magnitude be found where one regime stops and the other begins.

Even if this magnitude divider may be found, it is currently unclear if the numerical noise found in the low regime of the density solution can be separated from the weaker features due to the fact that the magnitude of the detail coefficients found are nearly equal to that of the actual structures.

The resulting refinement attained by the attempted automatic thresholding procedures at each detail level was satisfactory for the TAWS, TAO, and MA thresholding procedures. As can be seen above, all of the algorithms capture all of the major shocks through the use of a density only MRT solution. Although the density based solution cannot capture all flow features, it was shown that for weaker features, or those that are typically not density based phenomena that the use of multiple MRT transform criteria or multiple pass thresholding sweeps can assist in gaining lost flow features without the addition of much computational cost.

The presented methods for filtration of detail coefficients have shown marked improvement in the number of cells refined and occupancy ratio; however, these methods of filtration should be improved upon because the loss of weaker features is occurring and is not desired. Figure 5. 1 shows the loss of an important feature in the cylinder test case MA trial.

Compared to the gradient based method used it is clear that the MRT-based refinement method in conjunction with a “good” truncation parameter is able to filter out much of the noise and high refinement deemed unnecessary, as well as provide a graded mesh in areas of larger scales. The main advantage of the MRT-based LMR method over the gradient-based LMR method, however, is the ability of the MRT to separate scales and thus allow for the filtration of detail coefficients through methods such as statistical approximation, etc. The ability to estimate the detail coefficient magnitude in such ways then provides a foundation for a robust and automatic truncation algorithm to be developed and used on each scale and each time-step, which spares the user from tuning multiple refinement parameters.

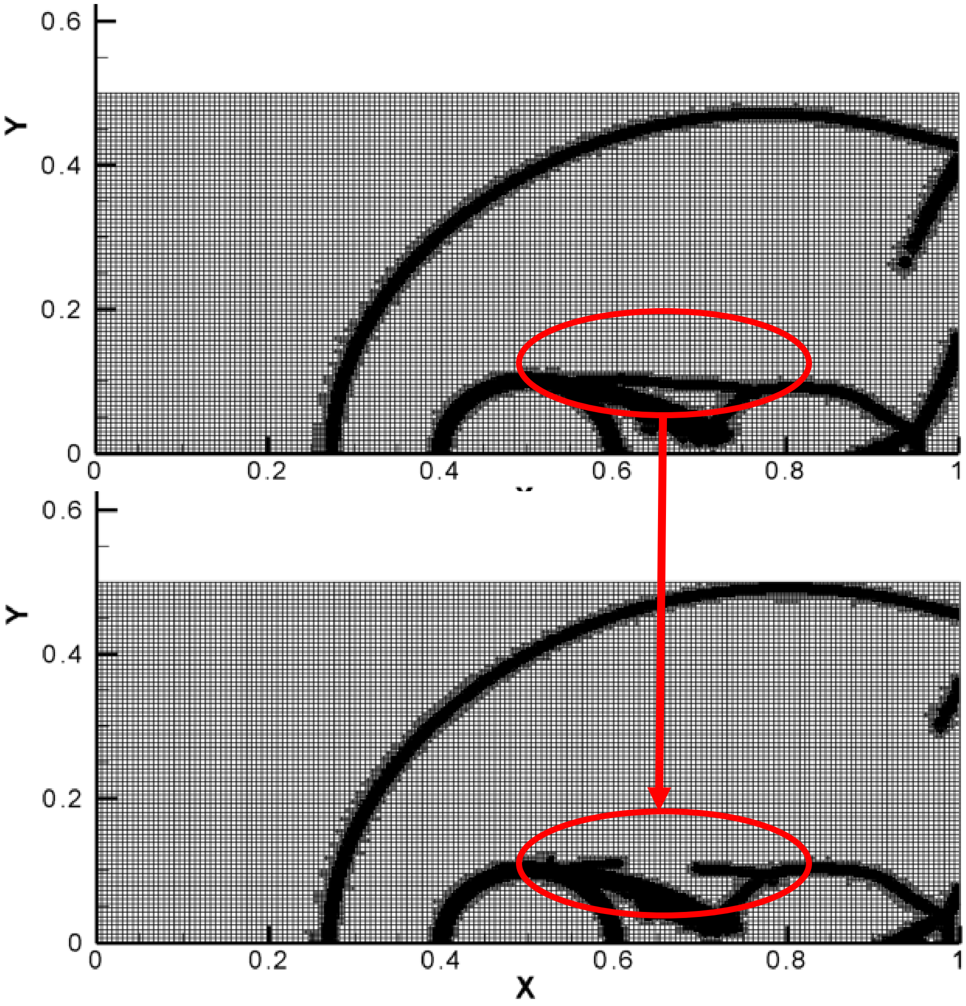


Figure 5. 1 Feature Loss Over a 120 Time-step Period: The feature encircled is desired and should not be lost over time. Additional measures for presenting such losses need to be explored.

Overall the results indicate that while the computational cost of performing the MRT on the mesh may be slightly higher than those of the gradient based method, the production of fewer cells alleviates computational time in the flow solver. At least with respect to the gradient-based LMR method, the computational costs should be significantly improved depending on the truncation method used. A summary of the resulting important refinement parameters for the cylinder and bubble test cases are summed up below in Table 5.1 and Table 5. 2. The gradient-based test case normalized occupancy ratio (OR) and total number of cells present (ncells) for two non-dimensional solution times is presented below for each test case.

Another way to further decrease the runtime of the ENO-MRT LMR scheme would be to reduce the number of transforms used. As might be apparent, most, if not all of the structures found during the compressible test cases were at the smallest scale (or highest refinement level). This could indicate that other levels of the MRT might be redundant, and eliminating these extra transform levels could improve the run speed of the MRT by up to 1.5 times (i.e. just take the 1st level transform of the function).

Table 5. 1 Refinement Results for the Cylinder Test Case: Ma = 2.81, dx = 0.01: Normalized wrt gradient-based case results.

Time (ND)	Grad OR	TAWS OR	TAO OR	MA
~7.939E-2	1.0000	0.6714	0.8650	0.7956
~9.511E-2	1.0000	0.6860	0.9650	0.8686
	Grad ncells	**TAWS ncells**	**TAO ncells**	**MA ncells**
~7.939E-2	1.0000	0.6714	0.8645	0.9159
~9.511E-2	1.0000	0.6928	0.9746	0.7620

Table 5. 2 Refinement Results for the Bubble Test Case:
Normalized wrt the gradient-based case results.

Time (ND)	Grad OR	TAWS OR	TAO OR
~12,009	1.0000	0.7350	0.7266
~37.195	1.0000	0.4479	0.7736
	Grad ncells	**TAWS ncells**	**TAO ncells**
~12,009	1.0000	0.7350	0.6900
~37.195	1.0000	0.4502	0.7775

5.2 Recommendations for Future Works

As mentioned above the MRT has a great deal of applications, which may lead to numerous future developments. Future work on the ENO-MRT code developed could include:

1. Development of a better thresholding scheme:

   The schemes attempted showed promising results, however, all of the statistical thresholding algorithms presented (with the possible exception of the TAO; which may not be the case if the mean is not sufficiently close to zero, and most of the weaker features have not already been removed by the first pass) are only valid under the assumption of the probability density of the details being Gaussian. The histograms presented clearly showed that this assumption is in fact incorrect, and the loss of some key features in the flowfield presents the need for a better model for the estimation of the truncation parameter to be found. Therefore, the Hidden Markov Method (HMM) proposed by [28] should better suit the data. Future works should include HMM testing and validation.

2. Completed inverse transform algorithm:

   A completed inverse MRT algorithm would be needed before any error based thresholding methods or data file compression methods could be attempted.

3. Data compression algorithms for post processing and file storage:

   Data compression could be done using both truncation and quantization of the data gained for file storage.

   a. Possible temporal compression:

      The data compression can further be enhanced through a temporal compression scheme in which two or more meshes for different time steps are transformed. All points showing little or no change for a given flow field variable over time can be compressed in the same way as in the spatial case.

4. Full integration of the ENO-MRT in 3D:

   The current code has not been fully adapted to function with three dimensional cases, partially due to computational constraints.

5. Integration into the incompressible flow code partition:

   The cases presented were all compressible flow cases, containing shocks and very sharp changes in flow field variables. It should be noted that although the ENO-MRT scheme is meant for this type of edge finding task that the MRT is not limited to use only in compressible flows. A simple bypass of the ENO stencil selection procedure and or lengthening of the stencil should provide ample detail information for incompressible calculations as well.

REFERENCES

1. Sambasivan, S.; UdayKumar, H.S.; In Publication *Sharp Interface Simulations of Compressible* Multiphase Flows with Local Mesh Adaptation.

2. Berger, M.J.; Collela, P.; Journal of Computational Physics *Local Adaptive Mesh Refinement for Shock Hydrodynamics* 1989, 82(1), 64-84.

3. Sambasivan, S.K. A Sharp Interface Cartesian Grid Hydrocode. PhD Dissertation, The University of Iowa, Iowa City, IA, 2010.

4. Amat, S.; Arandiga, F.; Cohen A.; Donat R.; Signal Processing *Tensor Product Multiresolution Analysis with Error Control for Compact Image Representation* 2002, 82(4), 587-608.

5. Chiavassa, G.; Donat, R; Muller, S.; Adaptive Mesh Refinement – Theory and Applications: *Multiresolution-based Adaptive Schemes for Hyperbolic Conservation Laws*, Springer-Verlag: Berlin, 2005; pp 137-159.

6. Cohen, A.; Dahmen, W.; DeVore, R.; Encyclopedia of Computational Mechanics: *Adaptive Wavelet Techniques in Numerical Simulation*; 1(2004), 157-197.

7. Bihari, B. L.; Ota, Liu Z.; Ramakrishnan, V.; AIAA Journal *Multiresolution Method on General Unstructured Meshes* 2002, 40(7), 1323-1330.

8. Boyd, J. P.; Chebyshev and Fourier Spectral Methods, 2nd ed; Dover Publications, New York , 2000.

9. Bracewell, R. N.; Scientific American *The Fourier Transform: DNA's double helix, the sunspot cycle, […]* 1989; 62-69.

10. Fetthauer F., Bryngdahl, O.; Optics Communications *Image Quantization by Windowed Iterative Fourier Transform Algorithm with Local Object Dependent Controll* 1997, 139(1), 187-192.

11. Jansen, M. H., Oonincx, P. J.; *Second Generation Wavelets and Applications* Springer-Verlag, London, 2005.

12. Wirasaet, D.; Paolucci, S.; Journal of Fluids Engineering *Adaptive Wavelet Methods for Incompressible Flows in Complex Domains* 2005, 127(4), 656-665.

13. Jensen, A.; A. la Cour-Harbo; *Ripples in Mathematics: The Discrete Wavelet Transform*; Springe-Verlag: Berlin, 2001.

14. Walker, J.S.; *A Primer on Wavelets and Their Applications*, 2nd ed.; Chapman and Hall/CRC, Boca Raton, 2008.

15. Donoho, D.; Johnstone, M.; Journal of the American Statistical Association *Adapting to Unknown Smoothness via Wavelet Shrinkage* 1995, 90(432), 1200-1224.

16. Moulin, P.; Liu, J., IEEE Transaction on Information Theory *Analysis of Multiresolution Image Denoising Schemes Using Generalized and Gaussian and Complexity Priors* 1999; 45(3), 909-919.

17. Krishnan, S.; Marella, S.V.; UdayKumar H.S.; International Journal of Computational Fluid Dynamics *Adaptively Refined, Parallelised Sharp Interface Cartesian Grid Method for Three-dimensional Moving Boundary Problems* 2009, 23(1), 1-24.

18. Coifman, R. R.; Wickerhauser M. V.; IEEE Transactions on Information Theory *Entropy Based Algorithms for Best Basis Selection* 1992, 38(2); 713-718.

19. Harten, A. *Multiresolution Representation and Numerical Algorithms: A Brief* Overview; NASA-CR-194949, 1994.

20. Amat, S.; Donat, R.; Lindrat, J.; Trillo, J.C.; Foundations of Computational Mathematics *Analysis of a New Nonlinear Subdivision Scheme: Applications to Image Processing* 2006, 6(2), 193-225.

21. Harten, A. Society for Industrial and Applied Mathematics *Multiresolution Representation of Data: A General Framework* 1996, 33(3), 1205-1256.

22. Sweldons, W.; The Lifting Scheme: Applied and Computational Harmonic Analysis *A Custom-Design Construction of Biorthogonal Wavelets* 1996, 3(15), 186-200.

23. Peng, J.; Strela, V.; Zorin, D.; Proceedings of the IEEE Conference on Visualization: *A Simple Algorithm for Surface Denoising*, San Diego, CA; 2001.

24. Guskov, I., Sweldens, W., Schroeder, P.; SIGGRAPH Conference Proceedings *Multiresolution Signal Processing for Meshes* 1999, 325-344.

25. Fornberg, B., Zuev, J.; Computers and Mathematics with Applications, *The Runge Phenomenon and Spatially Variable Shape Parameters in RBF Interpolation* 2007, 54(3), 379-398.

26. Arandiga, F.; Donat, R.; Numerical Algorithms *Nonlinear Multiscale Decompositions: The Approach of A. Harten* 2000, 23(1), 175-216.

27. Amat, S.; Busquier, S.; Trillo, J. C.; Numerical Algorithms *On Multiresolution Schemes Using a Stencil Selection Procedure: Applications to ENO Schemes* 2007, Springer-Verlag, 44(1), 45-68.

28. Crouse, M. S., Nowak, R. D, Baraniuk, R. G.; Signal Processing *Wavelet -Based Statistical Signal Processing using Hidden Markov Models* 1998, 46(4), 886-902.

29. Stark, J-L.; Murtagh, F.; The Astronomical Society of the Pacific *Automatic Noise Estimation from the Multiresolution Support* 1998, 110(1), 193-199.

30. Gross, M. H.; Gatti, R.; Staadt O.; Proceedings of the 6th IEEE Visualization Conference *Fast Multiresolution Surface Meshing* 1995.

31. Ofengeim, D. Kh., Drikakis, D,; Shock Waves *Simulation of Blast Wave Propagation Over a Cylinder* 1997, Springer-Verlag, 7(5), pp. 305-317.

32. Bagabir, A., Drikakis, D.; Shock Waves *Mach Number Effects on Shock Bubble Interaction* 2001, Springer-Verlag, 11(), 209-218.

33. Friedrich O.; Schoeder-Pander, F.; Sonar, T.; Numerische Matematik: *Generalized Multiresolution Analysis on Unstructured Grids*, Springer, 86(4), 2000; pp 685-715.

APPENDIX

PSEUDO-CODES

Wavelet Code:

```
%Perform the Haar wavelet transform
FUNCTION HaarTransform(f)
      S(1) = 1/SQRT(2);       %Define scaling filters
      S(2) = 1/SQRT(2);
      D(1) = 1/SQRT(2);       %Define wavelet filters
      D(2) = -1/SQRT(2);

      c = 0;                        %Counter
      DO j = 2, length(f)/2, 2      %Loop over ncells/2
            c = c + 1;
            %Apply the wavelet and scaling filters
            F(j-c)= S(1)*f(j-1) + S(2)*f(j)
            F(j+(n/2)-c) = D(1)*f(j-1) + D(2)*f(j)
      END
      %Final index notation output
      OUTPUT = F;

END FUNCTION
```

Then for recursion:

```
%Redefine the input function as the first half of the DWT
f(k+1) = F(1:length(f)/2)
```

Cell Selection Code:

```
%Count cells used for the stencil until 4*(m-1) cells are reached
DO WHILE (cellcnt .LT. 4*(m-1))
%If 2(m-1) cells have not already been selected to the right
IF (rightcnt .LT. 2*(m-1)) THEN
      MajorStencil(rightcnt) = CellRight(rightcnt)
      IF (CurrentLevel(CellLeft) .EQ. Kn) THEN
            cellcnt = cellcnt + 1;
            rightcnt = rightcnt + 1;
      ELSE
            cellcnt = cellcnt + 2;
            rightcnt = rightcnt + 2;
      ENDIF

ENDIF
%If 2(m-1) cells have not already been selected to the left
IF (leftcnt .LT. 2*(m-1)) THEN
      MajorStencil() = CellLeft
```

```
      IF (CurrentLevel(CellLeft) .EQ. Kn) THEN
            cellcnt = cellcnt + 1
            leftcnt = leftcnt + 1;

      ELSE
            cellcnt = cellcnt + 2;
            leftcnt = leftcnt + 2;

      ENDIF
ENDIF

ENDDO
```

Cell Averaging Code:

```
%If the cells are brothers and are of the same refinement level
IF ((Parent(Cell(1)) .EQ. Parent(Cell(2))) .AND.
    (Level(Cell(1)) .EQ. Level(Cell(2))))  THEN
      %Average the cell positions and values
      CellVarAvg =0.5*(Variable(Cell(1))+Variable(Cell(2)))
      CellPosAvg =0.5*(Position(Cell(1))+Position(Cell(2)))
ENDIF
```

Divided Difference Code:

```
%Find the divided difference of a stencil size = m
FUNCTION DividedDifference(x,fofx)
      n = m
      DO WHILE (n-1 .GT. 1)
            DO j = 1, n-1
                   dd(i)=(fofx(i+1)-fofx(i))/(x(cntr + 1) - x(1))
            ENDDO
            fofx(1:n-1) = dd(1:n-1)
            cntr = cntr + 1
            n = n-1
      ENDDO
      OUTPUT= ( dd_final = dd(1) )
END FUNCTION

DO i = 1, m-1
      x(:) = (Position(Cell(i)):Position(Cell(i+(m-1))))) =
      fofx(:) = (Variable(Cell(i)):Variable(Cell(i+(m-1))) =
      StencilPos = MIN(DividedDifference(x,fofx))
ENDDO
```

After the stencil is selected the detail value may be found as follows:

Lagrange Interpolation Code:

```
%Interpolate a function about x_c, using m number of points
FUNCTION LagarngeInterp(x_c,x,fofx)
      product = 0
```

```
      DO i = 1, m
            product = fofx(i)
            DO j = 1, m
                  IF (I .NE. j) THEN
                  product = product*((x_c-x(j))/(x(i)-x(j))
                  ENDIF
            ENDDO
      InterpValue = InterpValue + product
      ENDDO
RETURN(InterpValue)
END FUNCION

InterpValue = LagrangeInterp(x_c,x(StencilPos),fofx(StencilPos))
```

And then the detail coefficient can be found by:

```
Detail(x(center)) = Value(x(c)) – InterpValue
```