4-2004

# Multiresolution Wavelet Based Adaptive Numerical Dissipation Control for High Order Methods

Bjorn Sjögreen
*Lawrence Livermore National Laboratories*, sjogreen2@llnl.gov

Helen C. Yee
*NASA Ames Research Center*, yee@nas.nasa.gov

*Journal of Scientific Computing, Vol. 20, No. 2, April 2004 (© 2004)*

# Multiresolution Wavelet Based Adaptive Numerical Dissipation Control for High Order Methods

**B. Sjögreen[1] and H. C. Yee[2]**

*Received September 18, 2002; accepted (in revised form) January 8, 2003*

The recently developed essentially fourth-order or higher low dissipative shock-capturing scheme of Yee, Sandham, and Djomehri [25] aimed at minimizing numerical dissipations for high speed compressible viscous flows containing shocks, shears and turbulence. To detect non-smooth behavior and control the amount of numerical dissipation to be added, Yee *et al.* employed an artificial compression method (ACM) of Harten [4] but utilize it in an entirely different context than Harten originally intended. The ACM sensor consists of two tuning parameters and is highly physical problem dependent. To minimize the tuning of parameters and physical problem dependence, new sensors with improved detection properties are proposed. The new sensors are derived from utilizing appropriate non-orthogonal wavelet basis functions and they can be used to completely switch off the extra numerical dissipation outside shock layers. The non-dissipative spatial base scheme of arbitrarily high order of accuracy can be maintained without compromising its stability at all parts of the domain where the solution is smooth. Two types of redundant non-orthogonal wavelet basis functions are considered. One is the B-spline wavelet (Mallat and Zhong [14]) used by Gerritsen and Olsson [3] in an adaptive mesh refinement method, to determine regions where refinement should be done. The other is the modification of the multiresolution method of Harten [5] by converting it to a new, redundant, non-orthogonal wavelet. The wavelet sensor is then obtained by computing the estimated Lipschitz exponent of a chosen physical quantity (or vector) to be sensed on a chosen wavelet basis function. Both wavelet sensors can be viewed as dual purpose adaptive methods leading to dynamic numerical dissipation control and improved grid adaptation indicators. Consequently, they are useful not only for shock-turbulence computations but also for computational aeroacoustics and numerical combustion. In addition, these

[1] Department of Numerical Analysis and Computer Sciences, KTH, 100 44 Stockholm, Sweden. E-mail: bjorns@nada.kth.se

[2] NASA Ames Research Center, Moffett Field, California 94035.

0885-7474/04/0400-0211/0 © 2004 Plenum Publishing Corporation

sensors are scheme independent and can be stand-alone options for numerical algorithms other than the Yee *et al.* scheme.

**KEY WORDS:** Low dissipation; high order finite difference method; shock waves; multi resolution; wavelets; TVD schemes.

## 1. INTRODUCTION

Efficient and accurate numerical simulation of fluid flows containing both sharp layers and turbulence are computationally very challenging. Resolving a wide range of length scales, including shock layers, and high shear mixings is time consuming and costly. Numerical methods of the total variation diminishing (TVD) type for shock capturing are too dissipative to be useful when turbulence is present. Higher order difference methods in conjunction with scalar or characteristics type nonlinear numerical dissipations, without appropriate adaptive sensor control, behave similarly. Accurate methods such as spectral, spectral elements and high order spectral-like compact schemes for computing turbulence, break down when shocks are present. Although CPU intensive high order essentially nonoscillatory (ENO), weighted ENO (WENO) and discrete Galerkin schemes, generally exhibit less numerical dissipation than TVD schemes, nevertheless, their built in numerical dissipation still prevents the accurate capturing of fine scale turbulent structures without resorting to very fine grids. On the other hand, the switching mechanisms for multi-dimensional complex flow structures in hybrid schemes (e.g., switch between spectral element and ENO schemes) are extremely complicated and frequent activations of the ENO schemes are expected. As a remedy for this situation, the artificial compression method *(ACM) based filter scheme* was proposed in Yee *et al.* [25]. In this method one time step consists of one step with a fourth-order or higher accurate non-dissipative spatial base scheme. Often an entropy split form of the inviscid flux derivative (Yee *et al.* [27] and Gerritsen and Olsson [3]) is used along with a post processing step, where regions of oscillation are detected and filtered by adding the numerical dissipation portion of a shock capturing scheme at these parts of the solution. The entropy splitting of the inviscid flux derivative is considered as a conditioned form of the governing equations. The idea of the scheme is to have the spatially fourth-order or higher non-dissipative scheme activated at all times and to add the full strength, efficient and accurate numerical dissipation only at the shock layers. Thus, it is necessary to have good detectors which flag the layers, and not the oscillatory turbulent parts of the flow field. It was shown in Yee *et al.* [25, 27, 28] that the ACM sensor, while minimizing the use of numerical dissipation away from discontinuities, consists of tuning parameters and is physical problem dependent. The objective of the present work is to develop adaptive numerical dissipation sensors that are improvements over the ACM sensor.

Wavelets were originally developed for feature extraction in image processing and for data compression. It is well known that the regularity of a function can be determined from its wavelet coefficients [1, 14, 9] far better than from its Fourier coefficients. By computing wavelet coefficients (of a suitable set of wavelet basis functions), we obtain very precise information about the regularity of the function in question. This information is obtained just by analyzing a given grid function. No information about the particular problem which is solved is used. Thus, wavelet detectors are general, problem independent, and rest on a solid mathematical foundation.

As of the 1990's, wavelets have been a new class of basis functions that are finding use in analyzing and interpreting turbulence data from experiments. They are also used for analyzing the structure of turbulence from numerical data obtained from direct numerical simulation (DNS) or large eddy simulation (LES). See Farge [2], and Perrier *et al.* [16]. There are several ways to introduce wavelets. One standard way is through the continuous wavelet transform and another is through multiresolution analysis, hereafter, referred to as wavelet based multiresolution analysis. Mallat and collaborators [9–14] established important wavelet theory through multiresolution analysis. See references [21, 22] for an introduction to the concept of multiresolution analysis. Recently, wavelet based multiresolution analysis has been used for grid adaptation (Gerritsen and Olsson [3]) and to replace existing basis functions in constructing more accurate finite element methods. Here we utilize wavelet based multiresolution analysis to adaptively control the amount of numerical dissipation that is inherent in standard high-resolution shock-capturing schemes. With a proper choice of a set of the wavelet basis functions, we have a better control on the proper distribution and the amount of numerical dissipation leading to a more accurate simulation than the ACM sensor, especially than the numerical dissipations that are built-in in the TVD, ENO, and WENO schemes. The resulting wavelet sensors are readily available as more desirable grid adaptation indicators than those commonly used. It is well known that numerical dissipation is the main cause of wrong speed of propagation of discontinuous data in numerical combustion (LeVeque and Yee [8]) unless an order of magnitude adaptive grid refinement and time step reduction are used. Numerical dissipation is also a major stumbling block in efficient and accurate simulation of aeroacoustic problems. Consequently, the proposed wavelet based adaptive numerical dissipation control and grid adaptation indicator can be valuable to numerical combustion and computational aeroacoustics. In addition, this dual purpose adaptive method is scheme independent and can be a stand-alone option for a variety of schemes other than what is discussed here. A procedure which adapts the appropriate type of filters for the particular flow structures for long time integrations is proposed and studied in Yee and Sjögreen [26] and Sjögreen and Yee [20]. This adaptive detection/filter approach is able to distinguish shock/shear, turbulent fluctuation and spurious high frequency oscillation, and is valuable for high speed turbulence and combustion simulations. For numerical experiments in [20], it appears that more than one type of filter is needed for this type of applications.

Section 2 reviews the Yee *et al.* [25] high order scheme employing ACM as a numerical dissipation sensor. Section 3 derives two multiresolution wavelet numerical dissipation sensors accompanied with scalar examples. Section 4 discusses the options in applying the wavelet sensor for highly coupled nonlinear systems of conservation laws. Numerical experiments for 2-D compressible Euler and Navier–Stokes equations are given in Sec. 5.

## 2. HIGH ORDER ACM BASED FILTER SCHEME

In vector notation the 2-D compressible time-dependent Euler equations in conservation form for a perfect gas can be written, in Cartesian coordinates, as

$$U_t + F_x + G_y = 0, \tag{2.1}$$

where $U_t = \frac{\partial U}{\partial t}$, $F_x = \frac{\partial F}{\partial x}$, and $G_y = \frac{\partial G}{\partial y}$ and the $U$, $F$, $G$, are vectors given by

$$\begin{aligned} &U = (\rho, \rho u, \rho v, e)^T, \\ &F = (\rho u, \rho u^2 + p, \rho uv, eu + pu)^T, \qquad G = (\rho v, \rho uv, \rho v^2 + p, ev + pv)^T. \end{aligned} \tag{2.2}$$

The dependent variable $U$ is the vector of conservative variables, and $(\rho, u, v, p)^T$ is the vector of primitive variables. Here $\rho$ is the density, $u$ and $v$ are the $x$- and $y$-velocity components, $\rho u$ and $\rho v$ are the $x$- and $y$-components of the momentum per unit volume, $p$ is the pressure, $e = \rho[\varepsilon + (u^2 + v^2)/2]$ is the total energy per unit volume, and $\varepsilon$ is the specific internal energy. For constant specific heats (calorically perfect gas) $\varepsilon = c_v T$, where $c_v$ is the specific heat at constant volume, $T = p/R\rho$ is the temperature and $R$ is the perfect gas constant.

The eigenvalues associated with the flux Jacobian matrices of $F$ and $G$ are $(u, u, u \pm c)$ and $(v, v, v \pm c)$, where $c$ is the sound speed. The two $u, u$ and $v, v$ characteristics are linearly degenerate. Hereafter, we refer to the fields associated with the $u \pm c$ and $v \pm c$ characteristics as the *nonlinear fields* and the fields associated with the $u, u$ and $v, v$ characteristics as the *linear fields*.

*Spatial Discretizations.* The spatial discretizations of the ACM based filter scheme of Yee *et al.* [25] consist of two parts, namely, a *base scheme* and a *filter*. When numerical dissipations or filters are not used, the scheme consists of only the base scheme. Entropy splitting, or skew-symmetric splitting, (Yee *et al.* [27] and Gerritsen and Olsson [3]) can be used to improve the nonlinear stability of the scheme. It is especially important for long-time integrations, and enables a norm estimate of the solution. It has been reported [20] that the onset of non-linear instability can be delayed for an order of magnitude in the time scale, by using entropy splitting. If entropy splitting is used, the base scheme is applied to the split form of the inviscid flux derivatives. It is noted that the entropy splitting form of the inviscid flux derivatives can be viewed as a more conditioned form of the Euler equations for stability considerations. See Yee *et al.* [18, 27, 28] for details. Possible non-dissipative high order base schemes for $F_x$ and $G_y$ and the viscous terms (if present) are the standard *fourth and sixth-order compact and non-compact central schemes* for the interior grid points.

*The ACM Filter.* For efficiency and ease of numerical boundary treatment, Yee *et al.* [25] proposed using filter operators whose grid stencils have a width similar to that of the base scheme. The filter operator consists of the *product* of a *sensor* and a *nonlinear dissipation*. Denote $F_{j,k}$ as the discrete approximation of the inviscid flux $F$ at $(j\,\Delta x, k\,\Delta y)$, where $\Delta x$ and $\Delta y$ are the grid spacings in the $x$- and $y$-directions and $j$ and $k$ are the corresponding spatial indices. Let the filter vector in the $x$-direction be of the form

$$\tilde{F}^{*}_{j+\frac{1}{2},k} = \tfrac{1}{2} R_{j+\frac{1}{2}} \Phi^{*}_{j+\frac{1}{2}}. \tag{2.3}$$

$\tilde{F}^{*}_{j+\frac{1}{2},k}$ is the *modified* form of the *nonlinear dissipation portion* of the standard numerical flux. For characteristic based methods, the quantity $R_{j+\frac{1}{2}}$ (with the $k$ index suppressed) is the right eigenvector matrix of $\frac{\partial F}{\partial U}$ using Roe's average state (Roe's approximate Riemann solver). $\tilde{G}^{*}_{j,k+\frac{1}{2}}$ is defined in the same manner. The elements of $\Phi^{*}_{j+\frac{1}{2}}$ (with the $k$ index suppressed), denoted by $(\phi^{l}_{j+\frac{1}{2}})^{*}$, are

$$(\phi^{l}_{j+\frac{1}{2}})^{*} = \mathscr{S}^{l}_{j+\frac{1}{2}} \tilde{\phi}^{l}_{j+\frac{1}{2}}. \tag{2.4}$$

$\tilde{\phi}^{l}_{j+\frac{1}{2}}$ is the dissipative portion of the high resolution scheme resulting from using a TVD, MUSCL with slope limiters, ENO or WENO scheme. Formulae for $\tilde{\phi}^{l}_{j+\frac{1}{2}}$ are well known and can be found in the literature. See

Yee *et al.* [25] for details and for a discussion of other possible $\tilde{\phi}^l_{j+\frac{1}{2}}$. Here $\mathscr{S}^l_{j+\frac{1}{2}}$ is the sensor and is a mechanism for controlling excess numerical dissipation that is inherent in the dissipative portion of standard high-resolution shock-capturing schemes.

For the numerical experiments to be presented later, we use the Harten and Yee upwind TVD numerical dissipation (Yee [23, 24])

$$\tilde{\phi}^l_{j+\frac{1}{2}} = \tfrac{1}{2}\,\tilde{\psi}(a^l_{j+\frac{1}{2}})(g^l_{j+1}+g^l_j) - \tilde{\psi}(a^l_{j+\frac{1}{2}}+\gamma^l_{j+\frac{1}{2}})\,\tilde{\alpha}^l_{j+\frac{1}{2}}, \tag{2.5}$$

$$\gamma^l_{j+\frac{1}{2}} = \tfrac{1}{2}\,\tilde{\psi}(a^l_{j+\frac{1}{2}}) \begin{cases} (g^l_{j+1}-g^l_j)/\tilde{\alpha}^l_{j+\frac{1}{2}} & \tilde{\alpha}^l_{j+\frac{1}{2}} \neq 0 \\ 0 & \tilde{\alpha}^l_{j+\frac{1}{2}} = 0. \end{cases} \tag{2.6}$$

The $a^l_{j+\frac{1}{2}}$, $l = 1, 2, 3, 4$, are the characteristic speeds of $\frac{\partial F}{\partial U}$ evaluated at the Roe's average state. The $\tilde{\alpha}^l_{j+\frac{1}{2}}$ are elements of $R^{-1}_{j+\frac{1}{2}}(U_{j+1,k}-U_{j,k})$. The corresponding $\tilde{\alpha}^l_{j+\frac{1}{2}}$, $\tilde{\phi}^l_{j+\frac{1}{2}}$, and $R_{j+\frac{1}{2}}$ using the MUSCL formulation are instead functions of the left and right states of $U$. The function $\tilde{\psi}$ is an entropy correction to $|a^l_{j+\frac{1}{2}}|$. One possible form is given by Harten and Hyman [6].

The "limiter" function $g^l_j$ used for the numerical experiment can be chosen as one of the following expressions

$$g^l_j = \{\tilde{\alpha}^l_{j-\frac{1}{2}}[(\tilde{\alpha}^l_{j+\frac{1}{2}})^2+\delta_2]+\tilde{\alpha}^l_{j+\frac{1}{2}}[(\tilde{\alpha}^l_{j-\frac{1}{2}})^2+\delta_2]\}/[(\tilde{\alpha}^l_{j+\frac{1}{2}})^2+(\tilde{\alpha}^l_{j-\frac{1}{2}})^2+2\delta_2\}], \tag{2.7a}$$

$$g^l_j = \text{min mod}(2\tilde{\alpha}^l_{j-\frac{1}{2}}, 2\tilde{\alpha}^l_{j+\frac{1}{2}}, \tfrac{1}{2}\,(\tilde{\alpha}^l_{j+\frac{1}{2}}+\tilde{\alpha}^l_{j-\frac{1}{2}})), \tag{2.7b}$$

$$g^l_j = S\cdot\max\,[0, \min(2\,|\tilde{\alpha}^l_{j+\frac{1}{2}}|, S\cdot\tilde{\alpha}^l_{j-\frac{1}{2}}), \min(|\tilde{\alpha}^l_{j+\frac{1}{2}}|, 2S\cdot\tilde{\alpha}^l_{j-\frac{1}{2}})]. \tag{2.7c}$$

Here $\delta_2$ is a small dimensionless parameter to prevent division by zero and $S = \text{sign}(\tilde{\alpha}^l_{j+\frac{1}{2}})$. The min mod function of a list of arguments is equal to the smallest number in absolute value if the list of arguments is of the same sign, or is equal to zero if any arguments are of opposite sign.

*The ACM Sensor.* An artificial compression method (ACM) originally proposed by Harten [4] makes use of the gradient sensor to switch between a higher order scheme and a first-order scheme. Yee *et al.* [25] use this gradient sensor as part of $\mathscr{S}^l_{j+\frac{1}{2}}$. In contrast to hybrid schemes that switch between spectral or spectral-like non-shock-capturing schemes and high order ENO schemes, the high order non-dissipative base scheme is always activated. For the ACM sensor, $\mathscr{S}^l_{j+\frac{1}{2}} = \kappa\theta^l_{j+\frac{1}{2}}$. The parameter $\kappa$ is problem-dependent. The function $\theta^l_{j+\frac{1}{2}}$ is the Harten ACM gradient sensor. For a general $(2m+1)$-point base scheme,

$$\theta^l_{j+\frac{1}{2}} = \max(\hat{\theta}^l_{j-m+1}, \ldots, \hat{\theta}^l_{j+m}), \tag{2.8}$$

$$\hat{\theta}^l_j = \left| \frac{|\tilde{\alpha}^l_{j+\frac{1}{2}}| - |\tilde{\alpha}^l_{j-\frac{1}{2}}|}{|\tilde{\alpha}^l_{j+\frac{1}{2}}| + |\tilde{\alpha}^l_{j-\frac{1}{2}}|} \right|^p. \tag{2.9}$$

Here the parameter $p$ is an exponent $\geqslant 1$ and is not the "pressure $p$" in (2.1) and (2.2).

For smooth flows in the absence of high shears, $\kappa$ can be very small. It is used to minimize spurious high frequency oscillation producing nonlinear instability associated with pure central schemes, especially for long time integration problems. Different physical problems require different values of $\kappa$ because of the large variation in flow properties. The $\kappa$ value may vary from one characteristic wave to another, and from one region of the flow field to another region with different flow structure. Instead of varying $\kappa$ for the particular flow problem, one can vary $p$. For larger $p$, less numerical dissipation is added. Note that by varying $p \geqslant 1$ in the ACM sensor, one can essentially increase the order of accuracy of the filter.

For a chosen grid spacing without grid adaptation, we would like to point out that neither the ACM sensor nor the wavelet sensor (to be discussed next) would be able to improve the accuracy at the shock and shear locations over the inherent shock-capturing capability of the nonlinear dissipation. The accuracy of the shock and shear is dictated by the chosen flux limiter of the nonlinear dissipation. The role of the sensors is to allow the full amount of numerical dissipation in shock and shear regions, and to limit the amount of numerical dissipation in regions immediately away from shock and shear locations and the rest of the flow field. Therefore, with a suitable sensor and flux limiter, one does not have to use CPU-intensive high order high-resolution shock-capturing numerical dissipation. Only when sensors are not employed the higher-order nonlinear numerical dissipation gives a slightly more accurate solution away from discontinuities but exhibits similar shock and shear resolution as second- or third-order high-resolution numerical dissipation.

*Full Discretizations.* If a multistage time discretization such as the Runge–Kutta method is desired, the high order non-dissipative spatial differencing base scheme is applied at every stage of the Runge–Kutta method. If viscous terms are present, they use the same order and type of base scheme as for the inviscid terms. There are two methods for applying the characteristic filter. Method 1 is to apply the filter at every stage of the

Runge–Kutta step. Method 2 is to apply the filter at the end of the full Runge–Kutta step. For inviscid and strong shock interactions, method 1 might be more stable.

If one desires a time discretization that belongs to the class of linear multistep methods (LMMs), e.g., trapezoidal rule or three-point backward differentiation, then the filter can be applied as a numerical dissipation vector in conjunction with the base scheme. The filter in this case is evaluated at $U^n$ for explicit LMMs. For implicit LMMs additional similar filters evaluated at the $n+1$ time level might be involved. Alternatively, method 2 can be applied to LMMs as well. In this case, we apply the filter after the completion of the implicit time step.

As an example, we illustrate the complete form of the schemes for Runge–Kutta methods with the filters applied at the completion of a full Runge–Kutta time step. Let $\hat{U}^{n+1}$ be the solution after one full Runge–Kutta time step using a non-dissipative spatial base scheme. Then the solution at the next time level $U^{n+1}$ is

$$U^{n+1}_{j,k} = \hat{U}^{n+1}_{j,k} + \frac{\Delta t}{\Delta x}\left[ \tilde{F}^*_{j+\frac{1}{2},k} - \tilde{F}^*_{j-\frac{1}{2},k} \right] + \frac{\Delta t}{\Delta y}\left[ \tilde{G}^*_{j,k+\frac{1}{2}} - \tilde{G}^*_{j,k-\frac{1}{2}} \right]. \qquad (2.10)$$

Here, the numerical filters $\tilde{F}^*_{j\pm\frac{1}{2},k}$ and $\tilde{G}^*_{j,k\pm\frac{1}{2}}$ are evaluated at $\hat{U}^{n+1}$.

Thus, the filter is applied after each full time step with the Runge–Kutta method. For problems with very strong shocks, this might not give enough suppression of oscillations. For such problems, it is possible to apply the filter after each stage in the Runge–Kutta method instead. For all computations made in this article, the filter was applied only after each complete time step by the Runge–Kutta method.

## 3. WAVELET DETECTION ALGORITHMS

The ACM sensor function is not entirely satisfactory since tuning parameters $\kappa$ and $p$ are involved. We will develop here a wavelet based sensor function to replace $\kappa\theta^l_{j+\frac{1}{2}}$, which has the advantages of relying on a solid theoretical foundation, and minimizing the number of problem dependent parameters. Below we first describe the method in standard wavelet framework. Next we show how the Harten multiresolution method [5] can be used as a starting point for the derivation of a detection algorithm, on the same form. This modified Harten multiresolution description has the advantage of being more intuitive.

There exists today a large body of results on determining regularity of a function from its wavelet representation. See for example [1, 13, 14]. The wavelet technique has been especially useful in simulation of turbulence, where it can be used as a data analyzing tool, extracting structures from the numerical solution, [2, 15, 16]. In [3] a wavelet based multiresolution analysis from [14] is used to determine points where mesh adaption should be done. We will, to some extent, follow the description in [3] but rather utilize the wavelet technique for numerical dissipation control.

The wavelet decomposition of a function $f(x)$ is obtained by taking the inner product with wavelet functions $\psi_{m,n}(x)$. This leads to the wavelet coefficients,

$$w_{m,j} = \langle f, \psi_{m,j} \rangle = \int_{-\infty}^{\infty} f(x)\, \psi_{m,j}(x)\, dx,$$

$$m = \ldots, -1, 0, 1, \ldots, \quad j = \ldots, -1, 0, 1, \ldots \tag{3.1}$$

Here $m$ is an index representing scale, and $j$ is an index representing position. The set of basis functions $\psi_{m,j}(x)$ is obtained by scalings of a single "mother wavelet" function $\psi(x)$ and is not to be confused with the $\tilde{\psi}$ in (2.6). The construction of $\psi(x)$ depends on the types of application. Among the many rich mathematical wavelet developments, one of the key elements of the present interest is the scaling of the mother wavelet. The other is the mathematical characterization of multiresolution scales by the so called Lipschitz exponents [1, 13, 14].

There exist two major scalings leading to orthogonal and non-orthogonal wavelets. There also exist different scaling factors giving rise to different normalizations. The scaling we will use is

$$\psi_{m,j}(x) = 2^{-m} \psi((x-j)/2^m). \tag{3.2}$$

With this scaling the resulting wavelets do not form an orthonormal basis, and this is sometimes referred to as a redundant wavelet decomposition. In addition, the mother wavelet $\psi(x)$ should have compact support, or rapidly decrease towards infinity, so that the wavelet coefficient $w_{m,j}$ contains information about how much of scale $2^m$ is present in $f(x)$ at the point $x = j$. Additional technical conditions on $\psi(x)$, for example that its integral is equal to zero, should be imposed. Under these conditions it is possible to compute $f(x)$ from given coefficients $w_{m,j}$.

It is even possible to make the functions $\psi_{m,j}(x)$ an orthonormal basis for $L^2$. In fact, more known results and applications are based on orthonormal wavelet basis functions. In order to construct an orthonormal basis, the scaling of the wavelet function (3.2) is replaced by

$$\psi_{m,j}(x) = 2^{-m/2} \psi(x/2^m - j) \tag{3.3}$$

so that the number of coefficients on the coarser scale is not as densely distributed. In practice, the scaling is restricted by the chosen grid size in numerical computations and consequently, cannot be very small. If $m$ corresponds to grid levels, this scaling (3.3) would restrict us to fewer coefficients on the coarser grids, which is not unnatural. For our case, however, since we will estimate the regularity of a function at all grid points using several levels of wavelet coefficients, we keep all level of coefficients at all grid points. We can then estimate the regularity of the function $f(x)$ with the same accuracy at all grid points. With orthogonal wavelets we would have had a poorer estimate in between the coarse grid points. For a redundant wavelet decomposition, the scaling (3.2) is more natural and meets our requirement. Thus we will not use orthogonal wavelets.

We are here mainly interested in computing the wavelet coefficients (3.1) and the information which we can obtain from them. In the wavelet based multiresolution analysis, there exist several theorems about the relation between the regularity of a function $f(x)$ and its wavelet coefficients (3.1), see [1, 2, 9, 13, 16]. For example Theorem 9.2.2 in [1], which states that if $\psi$ is in $C^1$ and has compact support, and if the wavelet coefficients satisfy

$$\max_{j \in S(x_0, m, \epsilon)} |\langle f, \psi_{m,j} \rangle| \leqslant C 2^{m\alpha} \tag{3.4}$$

for all $m$ smaller than some limit, and for some $\epsilon > 0$, and some $\alpha$, $0 < \alpha < 1$, then the function $f$ has Lipschitz exponent $\alpha$ at $x = x_0$,

$$|f(x) - f(x_0)| \leqslant C\,|x - x_0|^{\alpha} \tag{3.5}$$

for all $x$ in a neighborhood of $x_0$. $S(x_0, m, \epsilon)$ is the extended domain of dependence of the wavelet function,

$$S(x_0, m, \epsilon) = \{j\colon \psi_{m,j}(x) \neq 0 \text{ for some } x \in (x_0 - \epsilon, x_0 + \epsilon)\}. \tag{3.6}$$

The theorem was originally formulated for non-redundant wavelets. An advantage of using a redundant decomposition is that the same theorem holds, but without the regularity requirement that $\psi(x) \in C^1$, see [1], p. 302.

The assumptions on $f$ and $\psi$ for similar results to hold vary between different references. Often it is required that $f$ is in $L^2$, bounded and continuous, and that $\psi$ is $C^1$ and compactly supported. Furthermore, the number of vanishing moments of $\psi(x)$ determines an upper limit on $\alpha$. For example, if the first two moments of $\psi(x)$ vanish, the result is valid for

$0 < \alpha < 2$. A $k$th vanishing moment means that $\int_{-\infty}^{\infty} x^k \psi(x)\, dx = 0$. The result in [13, 14], which is used in [3], instead gives the condition that $\psi(x)$ should be the $k$-derivative of a smooth function $\eta(x)$ with the property

$$\eta(x) > 0, \qquad \int \eta(x)\, dx = 1, \qquad \lim_{x \to \pm\infty} \eta^{(k)}(x) = 0. \tag{3.7}$$

Then the result is valid for $0 < \alpha < k$. A continuous function has a Lipschitz exponent $\alpha > 0$. A bounded discontinuity (shock) is $\alpha = 0$, and a Dirac function (local oscillation) has $\alpha = -1$. Large values of $k$ can be used in turbulent flow so that large vortices or vortex sheets can be detected. Although the theorem above does not hold for $\alpha$ negative, a useful upper bound on $\alpha$ can be obtained from the wavelet coefficient estimate. The works [9, 13] are good references for more detailed results on regularity estimates from wavelets. Before the discussion of the numerical computation of the wavelet coefficients we would like to emphasize that the Lipschitz exponent $\alpha$ in (3.4) and (3.5), which measures the regularity of a function $f(x)$, holds the key to our wavelet based adaptive numerical dissipation control.

### 3.1. Numerical Computation of the Wavelet Coefficients

For practical computations, we cannot make the scale $2^m$ infinitely small. The smallest scale is given by the grid, which we normalize to $m = 0$. For the discrete case, the index $m$ denotes the *grid level*, and the index $j$ is the *grid point* index of the wavelet coefficients. The grid points are $x = j$, $j = 1, 2,..., N$. For increasing $m$ the support of $\psi_{m,j}(x)$ increases, which means that the scale becomes larger. For the present application, the regularity of the function that we want to analyze consists of numerical data obtained from numerically solving a system of nonlinear partial differential equations (PDEs) with a chosen numerical scheme (e.g. (2.10)).

For a given grid function $f_j$, $j = 1, 2,..., N$ that we want to analyze, we now describe how to compute numerically the coefficients $w_{m,j}$ for $m = 1, 2,..., m_0$ and $j = 1, 2,..., N$. $m_0$ is the coarsest scale that we want to use. Of course, we could evaluate the coefficients directly from the definition $\langle f, \psi_{m,j} \rangle$, by numerical quadrature. This would be very expensive because the support of $\psi_{m,j}$ increases when $m$ increases, and the quadrature formulas would then involve sums where the number of terms approaches $N$ on the coarsest scales.

The basis of a computational scheme in obtaining the wavelet coefficients is the introduction of a so called scaling function $\phi(x)$, which belongs to the class of smoothing functions $\eta(x)$ in (3.7) satisfying

$$\phi(x) = 2 \sum_{k=-p}^{q} d_k \phi(2x-k) \qquad \psi(x) = 2 \sum_{k=-p}^{q} c_k \phi(2x-k). \tag{3.8}$$

The scaling functions $\phi(x)$ and $\phi_{m,j}(x)$ are not to be confused with the nonlinear numerical dissipation $\tilde{\phi}^{l}_{j+\frac{1}{2}}$ in (2.4). Only a few coefficients $d_k$ and $c_k$ should be nonzero, in order to obtain an efficient computational method.

The functional equations (3.8) is closely associated with the multiresolution analysis [21, 22, 9]. In the multiresolution analysis, a sequence of closed subspaces $V_j$ of $L^2(\mathbb{R})$, $j \in \mathbb{Z}$ of different resolution is defined,

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \cdots$$

with the important property that if a function $h(x) \in V_m$, then $h(2^m x) \in V_0$, so that all spaces are scaled versions of $V_0$. Here $\mathbb{R}$ and $\mathbb{Z}$ are the spaces of reals and integers respectively. Furthermore, we require that

$$\overline{\bigcup_{m \in \mathbb{Z}} V_m} = L^2(\mathbb{R}) \qquad \text{and} \qquad \bigcap_{m \in \mathbb{Z}} V_m = \{0\}.$$

In other words, $\bigcup_{m \in \mathbb{Z}} V_m$ is dense in $L^2(\mathbb{R})$. The spaces $V_m$ can be thought of as representing all scales down to the $m$th scale. We also require that $V_0$ is invariant under integer translations, so that a basis on the form $\{\phi(x-j)\}_j$ can be found. With the definition $\phi_{m,j} = 2^{-m/2}\phi(x/2^m - j)$, we then obtain $\{\phi_{m,j}\}_j$ as a basis for $V_m$. With these definitions the scaling relation (3.8) for $\phi$ just means that we expand the function, e.g., $\phi_{0,0}$ in the basis $\phi_{-1,j}$, which can be done since $V_0 \subset V_{-1}$.

Let $W_m$ be the orthogonal complement of $V_m$ in $V_{m-1}$, so that

$$V_{m-1} = W_m \oplus V_m,$$

where the symbol $\bigoplus$ stands for direct sum. It can then be shown that a basis $\{\psi_{m,j}(x)\}$ for $W_m$ can be found, where $\psi_{m,j} = 2^{-m/2}\psi(x/2^m - j)$. The spaces $W_m$ are orthogonal, since $W_m \subset V_{m-1}$, and $V_{m-1} \perp W_{m-1}$. We can write $L^2$ as a direct sum of $W_m$ spaces,

$$L^2(\mathbb{R}) = \bigoplus_m W_m$$

so that the $\psi_{m,j}$ form an orthogonal basis for $L^2$. This is an orthogonal wavelet basis for $L^2$. The scaling relation for $\psi$, can then be thought of as an expansion of $\psi_{0,0}$ in the basis $\phi_{-1,j}$, which can be done since $W_0 \subset V_{-1}$.

If a scaling function $\phi(x)$ satisfying (3.8) can be found, we obtain a fast method for the evaluation of the wavelet coefficients, independent of whether the formula was derived from an orthogonal wavelet basis, or was found in any other way. The computational scheme becomes

$$\begin{aligned} \langle f, \phi_{m,j}\rangle &= \sum_{k=-p}^{q} d_k \langle f, \phi_{m-1, j+2^{m-1}k}\rangle \\ \langle f, \psi_{m,j}\rangle &= \sum_{k=-p}^{q} c_k \langle f, \phi_{m-1, j+2^{m-1}k}\rangle \end{aligned} \tag{3.9}$$

where we have defined $\phi_{m,j}(x) = 2^{-m}\phi((x-j)/2^m)$. To derive (3.9) from (3.8), insert the argument $(x-j)/2^m$ in the place of $x$ and multiply by $2^{-m}$ to obtain

$$\begin{aligned} \frac{1}{2^m}\phi((x-j)/2^m) &= \frac{1}{2^{m-1}} \sum_{k=-q}^{p} d_k \phi\left(2\frac{x-j}{2^m} - k\right) \\ &= \frac{1}{2^{m-1}} \sum_{k=-q}^{p} d_k \phi\left(\frac{x-j-2^{m-1}k}{2^{m-1}}\right) \end{aligned}$$

from which we can identify the relation

$$\phi_{m,j}(x) = \sum_{k=-q}^{p} d_k \phi_{m-1, j+2^{m-1}k}(x).$$

Multiplying this equation by $f(x)$ and integrating, gives (3.9) for $\phi$. The relation for $\psi$ follows from similar computations. If the coefficients on the finest grid $\langle f, \phi_{0,j}\rangle$ are given, we can compute $\langle f, \psi_{1,j}\rangle$, $\langle f, \phi_{1,j}\rangle$, by using the formulae (3.9). Repeating the procedure gives $\langle f, \psi_{m,j}\rangle$ for successively increasing $m$. The formulae (3.9) are a pair of difference operators acting on a grid function $f_j$. We can express the algorithm as follows. Introduce the grid operators

$$\begin{aligned} Af_j &= \sum_{k=-p}^{q} d_k f_{j+k} \\ Df_j &= \sum_{k=-p}^{q} c_k f_{j+k} \end{aligned} \tag{3.10}$$

and their $m$th level expanded versions

$$\begin{aligned} A_m f_j &= \sum_{k=-p}^{q} d_k f_{j+2^m k} \\ D_m f_j &= \sum_{k=-p}^{q} c_k f_{j+2^m k}, \end{aligned} \tag{3.11}$$

where the integers $p$ and $q$ are related to the chosen $\psi(x)$ and $\phi(x)$. The computation (3.9) can then be written as

$$\begin{aligned} f_j^{(m)} &= \langle f, \phi_{m,j} \rangle = A_{m-1} f_j^{(m-1)} \\ w_{m,j} &= \langle f, \psi_{m,j} \rangle = D_{m-1} f_j^{(m-1)}. \end{aligned} \tag{3.12}$$

Starting from the grid function $f_j^{(0)} = \langle f, \phi_{0,j} \rangle$, (3.12) is applied successively to obtain grid functions for $m = 1, 2, 3, \ldots, m_0$. The initial $f_j^{(0)}$ is found by numerical quadrature. The computation $f_j^{(m)} = A_{m-1} f_j^{(m-1)}$ for a three point operator $A$ is outlined in Fig. 3.1. These sequences are sometimes referred to as the impulse response of low pass and high pass filters. The computation $w_{m,j} = D_{m-1} f_j^{(m-1)}$ follows a similar pattern, but possibly with a different stencil width.

The $m$th level of wavelet coefficients can be written as

$$w_{m,j} = \langle f, \psi_{m,j} \rangle = D_{m-1} A_{m-2} A_{m-3} \cdots A_0 f_j, \quad m = 1, 2, \ldots. \tag{3.13}$$

Once the coefficients $d_k$ and $c_k$ are determined the computation is a very standard application of grid operators. In practice, we only use $m_0 = 3$ to 5. To be able to compute up to the boundary, we use one sided versions of the given operators. This seems to work well in practice, although it is not covered by the wavelet framework described above. Study on the effect of

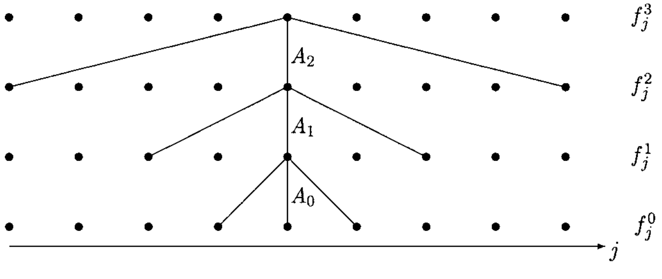


**Fig. 3.1.** The computation of $f_j^{(m)}$ (3.12), when $A$ has three point stencil.

wavelets on an interval using the appropriate boundary wavelets [9] for our application is a subject of ongoing research.

To determine the initial coefficients, $\langle f, \phi_{0,j} \rangle$ from $f_j$, we make a numerical approximation of the integral. If the support of $\phi(x)$ is small, we can set $\langle f, \phi_{0,j} \rangle \approx f_j \int \phi(x)\, dx$.

### 3.2. Approximation of the Lipschitz Exponent

After we compute the wavelet coefficients, the next step is to compute the Lipschitz exponent $\alpha$, the key for our present development. Unfortunately, it is not possible to obtain $\alpha$ exactly. There exist in the literature different methods for approximating $\alpha$. The method of approximating $\alpha$ in [3] is based on the theorems in [14], which involves tracing a maximum curve among the wavelet coefficients at points of singular behavior. We choose here, instead, to base our method on Theorem 9.2.2 of [1] as described previously. This means that we do not trace maximum lines. Instead the procedure below is applied at all grid points. Note that the results in both [1] and [14] are valid for functions of a continuous variable, so there is some freedom of interpretation when applying them to functions defined only at grid points.

After we have computed the wavelet coefficients, we first form the maximum over the domain of dependence,

$$r_{m,j} = \max_{k=-2^m p,\, 2^m q} |\langle f, \psi_{m,j+k} \rangle| \tag{3.14}$$

where the nonzero $c_k$ coefficients are enumerated from $-p$ to $q$. We estimate the Lipschitz exponent by a least squares fit of a line to the equation

$$\log_2 r_{m,j} = \alpha m + c. \tag{3.15}$$

The slope gives an estimate of $\alpha$ at the point $x = j$. A discontinuity is characterized by $\alpha = 0$. Standard numerical centered difference approximations have problems when $\alpha$ is small. Usually existence of several derivatives is required for high order difference formulas to be accurate.

### 3.3. Detectors from the B-Spline Wavelet Basis Function

Developing the best suited wavelets that can characterize all of the flow features might involve the switching or blending of more than one mother wavelet $\psi(x)$ and scaling function $\phi(x)$, especially if one needs to distinguish turbulent fluctuations from shock/shear and/or spurious high frequency oscillations. See [20, 26, 27] for some discussion. The mother wavelet function used in [3] and described in detail in [14] meets some of our requirements. It is obtained from second order B-splines, and is given by

$$\psi(x)=\begin{cases} 0 & x>1 \\ -2(x-1)^2 & 1/2<x<1 \\ -4x(1-x)+2x^2 & 0<x<1/2 \\ -4x(1+x)-2x^2 & -1/2<x<0 \\ 2(x+1)^2 & -1<x<-1/2 \\ 0 & x<-1 \end{cases}. \tag{3.16}$$

For this wavelet (3.16), there exists a scaling function, given by

$$\phi(x)=\begin{cases} 0 & x>2 \\ \frac{1}{2}(x-2)^2 & 1<x<2 \\ -(x-1/2)^2+3/4 & 0<x<1 \\ \frac{1}{2}(x+1)^2 & -1<x<0 \\ 0 & x<-1 \end{cases}. \tag{3.17}$$

If we apply the Fourier transform, $\hat{h}(\xi)=\int e^{-ix\xi}h(x)\,dx$, to the relation

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

we obtain

$$\hat{\phi}(\xi)=b_0(\xi/2)\,\hat{\phi}(\xi/2) \tag{3.18}$$

where $b_0(\xi)=\sum_k d_k e^{ik\xi}$. From a given trigonometric polynomial $b_0(\xi)$, we can find scaling functions by iterating (3.18),

$$\hat{\phi}(\xi)=\prod_{m=1}^{\infty} b_0(\xi/2^m)\,\hat{\phi}(0)$$

or conversely, for a given scaling function, we can try to find the polynomial $b_0(\xi)$ from (3.18). Taking the $\tilde{N}$ degree B-spline basis function as the scaling function gives

$$\hat{\phi}(\xi)=e^{-is\xi/2}\left(\frac{\sin \xi/2}{\xi/2}\right)^{\tilde{N}+1},$$

where $s$ is 1 for $\tilde{N}$ even, and 0 for $\tilde{N}$ odd. From this it is not hard to verify that

$$b_0(\xi) = e^{-is\xi/2} \cos^{\tilde{N}+1} \xi/2$$

and we get the coefficients $d_k$ from this trigonometric polynomial.

The normalization is such that the integral of the scaling function above is equal to one. The functions above are standard, and can be found in [1]. The scaling function differs by a shift from the scaling function used in [3], but the important relations

$$\begin{aligned} \phi(x) &= \tfrac{1}{4}\phi(2x+1) + \tfrac{3}{4}\phi(2x) + \tfrac{3}{4}\phi(2x-1) + \tfrac{1}{4}\phi(2x-2) \\ \psi(x) &= \phi(2x+1) - \phi(2x) \end{aligned} \tag{3.19}$$

hold, and give the grid operators

$$\begin{aligned} Af_j &= (f_{j-1} + 3f_j + 3f_{j+1} + f_{j+2})/8, \qquad && j = 2,\ldots, N-2 \\ Df_j &= (f_{j-1} - f_j)/2 && j = 2,\ldots, N. \end{aligned} \tag{3.20}$$

Here $p = 1$, $q = 2$, $d_{-1} = 1/8$, $d_0 = 3/8$, $d_1 = 3/8$, $d_2 = 1/8$, $c_{-1} = 0$, $c_0 = -1/2$, $c_1 = 1/2$, and $c_2 = 0$ for (3.11). For example, for $m_0 = 3$, the computation of $f_j^{(m)}$ involves a grid stencil of 13 points. Note that the wavelet stencil is not symmetric. In general, formula (3.9) shows that points from $p2^{m_0-1}$ to $-q2^{m_0-1}$ are involved in the computation, giving a stencil of totally $(p+q)\,2^{m_0-1}+1$ points. The scaling function $\phi(x)$ and the wavelet function $\psi(x)$ for the B-spline wavelet are shown in Figs. 3.2a and 3.2b respectively.

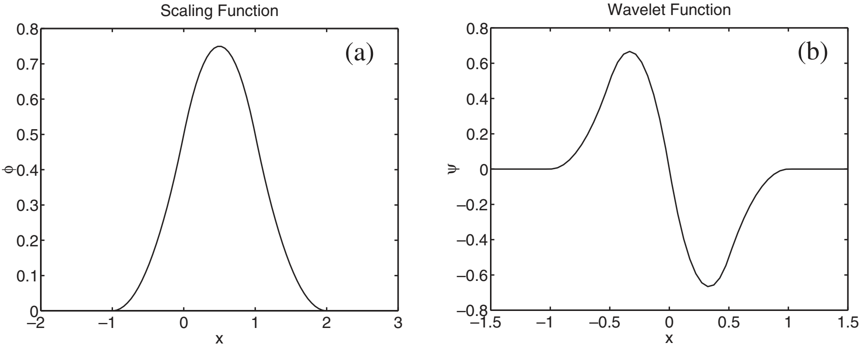


**Fig. 3.2.** (a) scaling function, (b) wavelet function.

For boundary operators we use

$$\begin{aligned} Df_1 &= (f_3-3f_2+2f_1)/2 \\ Af_1 &= (7f_1-3f_2+5f_3-f_4)/8 \\ Af_{N-1} &= (7f_N-3f_{N-1}+5f_{N-2}-f_{N-3})/8 \\ Af_N &= (25f_N-37f_{N-1}+27f_{N-2}-7f_{N-3})/8. \end{aligned} \tag{3.21}$$

The coefficients are determined so that the action on smooth functions is close to the interior operators. The simpler formulae

$$\begin{aligned} Df_1 &= (f_1-f_2)/2 \\ Af_1 &= (7f_1-3f_2+5f_3-f_4)/8 \\ Af_{N-1} &= (f_N+3f_{N-1}+3f_{N-2}+f_{N-3})/8 \\ Af_N &= (7f_N-3f_{N-1}+5f_{N-2}-f_{N-3})/8 \end{aligned} \tag{3.22}$$

have also turned out to work well in practice. We will later show some experiments with these operators. The computation consists of the following steps. Use (3.20) with (3.21) or (3.22) as the operators (3.10). Compute $w_{m,n}$ from (3.12), and estimate the regularity by (3.14) and (3.15).

### 3.4. Detectors from Converting Harten's Multiresolution Scheme to Redundant Wavelets

This section describes a procedure to modify Harten's multiresolution method [5] into redundant non-orthogonal wavelets and derive an alternate wavelet detector which in some sense is slightly simpler than the B-spline wavelet sensor. The multiresolution method by Harten is a way to speed up computations by using a wavelet decomposition. See Sjögreen [19] for a study of Harten's multiresolution scheme. We next give a brief description of the method.

Consider the approximation of a PDE in one space dimension, on the uniform grid $x_j = j\,\Delta x$, $j = 0, 1,..., N$. The numerical solution is given by $(f_0, f_1,..., f_N)$, with $f_j$ an approximation of the solution at $x_j$. Introduce $L$ levels of successively coarser grids,

$$G_m = (x_0, x_{2^m},..., x_N) \qquad m = 0,..., L.$$

Let $x_j^m$ denote grid point $j$ on grid $G_m$. Then $x_j^0 = j\,\Delta x$, and $x_j^m = x_{j2^m}^0 = j2^m\Delta x$. Let $N_m$ denote the number of points in $G_m$. Then

$N_m = 2^{L-m}N_L$. We let $f_j^m$ denote the numerical solution of a PDE at the point $x_j^m$.

Assume that the solution is given on grid $G_m$, and that we want to find it on the finer grid $G_{m-1}$. For the even numbered grid points we have

$$f_{2j}^{m-1} = f_j^m, \qquad j = 0, 1,..., N_m.$$

To find the solution at the odd grid points, we let $I(x, f^m)$ interpolate $f_j^m$ on $G_m$, such that $I(x_j^m, f^m) = f_j^m$. We then have the approximation $\hat{f}_{2j-1}^{m-1} = I(x_{2j-1}^{m-1}, f^m)$ of $f_{2j-1}^{m-1}$. The interpolation error is $d_j^m = f_{2j-1}^{m-1} - \hat{f}_{2j-1}^{m-1}$. Thus with knowledge of $f^m$ and $d^m$ we can reconstruct the solution on $G^{m-1}$. We call $(f^m, d^m)$ the multiresolution representation of $f^{m-1}$. Note that the vectors $f^m, d^m$ together contain the same number of elements as does $f^{m-1}$. In summary, we switch between the representation $f^{m-1}$, and $(d^m, f^m)$ by the forward transformation

$$\begin{aligned} f_j^m &:= f_{2j}^{m-1} \\ d_j^m &:= f_{2j-1}^{m-1} - I(x_{2j-1}^{m-1}, f^m), \end{aligned} \tag{3.23}$$

and the backward transformation

$$\begin{aligned} f_{2j}^{m-1} &:= f_j^m \\ f_{2j-1}^{m-1} &:= d_j^m + I(x_{2j-1}^{m-1}, f^m). \end{aligned}$$

This is inexpensive if $I(x_{2j-1}^{m-1}, f^m)$ is a straightforward linear interpolation operator.

We transform consecutively on all grids

$$f^0 \to (d^1, f^1) \to (d^1, d^2, f^2) \to \cdots (d^1, d^2,..., d^L, f^L).$$

The vectors $(d^1, d^2,..., d^L, f^L)$ are the multiresolution representation of $f^0$. The interpolation errors $d^m$ contain information about the smoothness of the solution. If $d_j^m$ is large at some grid points, this indicates that the solution is non-smooth there.

The multiresolution method does not increase the number of coefficients, and is therefore not redundant. Here, we propose instead to use a redundant form, obtained from computing coefficients $d_j^m$ at all points on the fine grid. To do this, we treat the fine grid as consisting of two coarse grids. The even points are interpolated from the odd points, and vice versa. The interpolation errors are used for wavelet coefficients. For this redundant method, there is only one fine grid, with points $x_j$, $j = 1, 2,..., N$, where we assume that $N$ is odd, $N = 2s - 1$. All grid functions $f^m$, $d^m$ are defined on this fine grid. Denote $I(x, \{f_k\}_{k=a}^b)$ a function which interpolates the values $f_a, f_{a+1},..., f_b$. Let the function values $f_j^{m-1}$, $j = 1, 2,..., N$ be given. The following formula will then replace (3.23),

$$\begin{aligned} f_{2j}^m &:= I(x_{2j}, \{f_{2k-1}^{m-1}\}_{k=1}^s), && j = 1, 2,..., s-1 \\ f_{2j-1}^m &:= I(x_{2j-1}, f_{2k}^{m-1}\}_{k=1}^{s-1}), && j = 1, 2,..., s \\ d_j^m &:= f_j^{m-1} - f_j^m && j = 1, 2,..., N. \end{aligned}$$

By expressing the wavelet coefficients as interpolation errors, it becomes intuitively clear why they should be small when the function is regular.

We express the computation of the level $m$ coefficients from the level $m-1$ coefficients in operator form as

$$\begin{aligned} f_j^m &= A_{m-1} f_j^{m-1} \\ d_j^m &= f_j^{m-1} - A_{m-1} f_j^{m-1} \end{aligned}$$

where $Af_j$ is an averaging operator, coming from the interpolation formula. We assume that the same $I(x, f)$ is used for the odd and the even points. The computation is done for all $j$ on the fine grid. At the boundary points, we use one sided versions of the operator $A$. Note that (3.23) is exactly the same form as (3.12), if we define $D = I - A$, with $d_j^m$ as $w_{m,j}$, and $f_j^m$ as $f_j^{(m)}$ in (3.12).

There is more than one choice for the interpolation function. See Sjögreen [19] for a discussion. The exact form of the method for the computations in this article will be

$$\begin{aligned} Af_j &= (f_{j-1} + f_{j+1})/2 && j = 2,..., N-1 \\ Df_j &= f_j - Af_j && j = 2,..., N-1. \end{aligned} \tag{3.24}$$

The above choice was made in order to have a simple and efficient method. The stencil is narrower than for the B-spline formulas that were given previously. With the formula above, we also get a symmetric stencil, which is more natural if the other parts of the computation, such as difference approximations of PDEs are done by symmetric formulas. Furthermore, symmetry makes periodic boundary conditions (BCs) somewhat easier to implement. Note that the absence of symmetry for either the scaling function or the wavelet can lead to phase distortion. This can be shown to be important for signal processing applications. Another advantage of (3.24) is that the wavelet has two vanishing moments, one more than the B-spline wavelet used in (3.20).

We now have a method in the same form as the wavelet detector from the previous subsection, but derived in an intuitive way and using a more narrow grid stencil. For example, for $m_0 = 3$, the computation of $f_j^{(m)}$ involves a grid stencil of 9 instead of 13. Using the above formula, we proceed with the Lipschitz exponent computations as before.

### 3.5. Two Space Dimensions

Wavelet functions can be defined in different ways in two dimensions. One way is to use the product basis functions,

$$\psi_{m,n,j,k}(x, y) = \psi_{m,j}(x)\,\psi_{n,k}(y)$$

giving two scales $m$ and $n$ at each point $(j, k)$ at the plane. It is possible to use instead

$$\psi_{m,j,k}(x, y) = \frac{1}{2^m}\,\psi((x-j)/2^m, (y-k)/2^m)$$

where only one scale is present at each grid point. The function $\psi(x, y)$ is a two dimensional version of $\psi(x)$.

The computation of multi-dimensional wavelets is quite expensive, especially in 3-D. A simple minded efficient way is to evaluate the wavelet coefficients dimension by dimension. This means that we get two sets of wavelet coefficients $w^x_{m,j}(y)$ and $w^y_{m,k}(x)$, where now $(j, k)$ is the position and $m$ is the scale. The precise definition is

$$w^x_{m,j}(y) = \int f(x, y)\,\psi_{m,j}(x)\,dx$$

$$w^y_{m,k}(x) = \int f(x, y)\,\psi_{m,k}(y)\,dy.$$

In the sequel, our numerical method will have some terms evaluated as finite differences in the $x$-direction and some which are evaluated in the $y$-direction. We then use the $w^x_{m,j}(y)$ coefficients for the $x$-direction computation, and the $y$-coefficients for the $y$-direction computation. When the coefficients are evaluated along grid lines, the notation

$$w^x_{m,j,k} = w^x_{m,j}(y_k) \qquad w^y_{m,j,k} = w^y_{m,k}(x_j)$$

is used.

The implementation of the algorithm is made memory efficient by computing the wavelet coefficients line by line. Only one dimensional extra memory is needed. An outline of the computational procedure (for three space dimensions) is given in Algorithm 3.1. We see that the routine requires 6 additional one dimensional arrays. Three levels of wavelet coefficients are hard coded into the algorithm. However, it would be straightforward to use an arbitrary number of levels, by looping over the levels instead. Algorithm 3.1 computes the $I$-sweep, and the operators $A, D$ are acting in the $I$-coordinate direction. In our implementation the same code is used for all coordinate sweeps. This is possible to achieve by programming the indexation of multi dimensional arrays in such a way that the innermost index is always the sweep direction. The input is the solution array $w_{i,j,k}$ defined for $1 \leqslant i \leqslant N_i$, $1 \leqslant j \leqslant N_j$, $1 \leqslant k \leqslant N_k$. The output is the adding of the filter fluxes, computed at $w_{i,j,k}$, to a residual $r_{i,j,k}$.

**Algorithm 3.1.**

```
for $k := 1$ to $N_k$
for $j := 1$ to $N_j$
  for $i := 1$ to $N_i$
     $d_i^1 := D_0 w_{i,j,k}$
     $f_i^1 := A_0 w_{i,j,k}$
  endfor
  for $i := 1$ to $N_i$
     $d_i^2 := D_1 f_i^1$
     $f_i^2 := A_1 f_i^1$
  endfor
  for $i := 1$ to $N_i$
     $d_i^3 := D_2 f_i^2$
  endfor
  for $i := 1$ to $N_i$
     Lipschitz exponent $\alpha$ computed from $d^1, d^2, d^3$.
     Compute sensor $s_{i+1/2} := \tau(\alpha)$.
  endfor
  for $i := 1$ to $N_i - 1$
    if $s_{i+1/2} \neq 0$ then
      Compute filter flux $h_{i+1/2}$
       add to residual:
      $r_{i,j,k} := r_{i,j,k} - s_{i+1/2} h_{i+1/2}$
      $r_{i+1,j,k} := r_{i+1,j,k} + s_{i+1/2} h_{i+1/2}$
    endif
  endfor
endfor
endfor
```

### 3.6. Numerical Experiments with 1-D Test Cases

We test here the detector algorithms on one dimensional examples, where the regularity of the functions is known. We will compare the redundant wavelet of Harten's multiresolution method (3.24), referred to as RH-wavelet, and the B-spline wavelet (3.20) referred to as BS-wavelet. A typical example from the wavelet literature is the function given in Fig. 3.3. This function has a steep layer, a jump ($\alpha = 0$), a sharp peak which on the grids that we use is under resolved and could be understood as a Dirac pulse. A smooth bump is also present. It has the form

$$f(x) = \begin{cases} 0 & x < -1 \\ 1/(e^{-15(x+3/4)} + 1) & -1 \leqslant x < -1/2 \\ 1 & -1/2 < x < 1/2 \\ 0 & 1/2 < x < 1.9 \\ 1 - 20\,|x - 1.95| & 1.9 < x < 2 \\ 0 & 2 < x < 3 \\ e^{-15(x-7/2)^2} & 3 < x < 4 \\ 0 & x > 4. \end{cases} \tag{3.25}$$

A good detector should flag the discontinuity, and the possible spike.

Figure 3.4 shows results from using the ACM sensor $\kappa\theta^l_{j+\frac{1}{2}}$ in (2.8) and (2.9) with $p = 1$ on the function in Fig. 3.3. We define $\tilde{\alpha}^l_{j+\frac{1}{2}}$ as $f_{j+1} - f_j$. The cases $\kappa = 1$ and $\kappa = 0.5$ are shown. Note that $\kappa$ is more relevant if it is a function of the dependent variables, and/or when more than one wave is

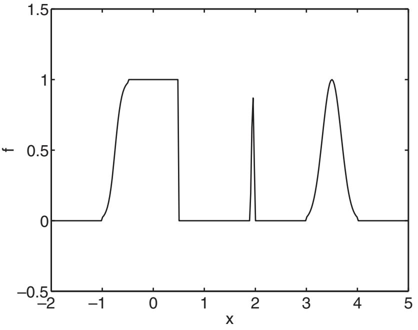


**Fig. 3.3.** Function used for testing.

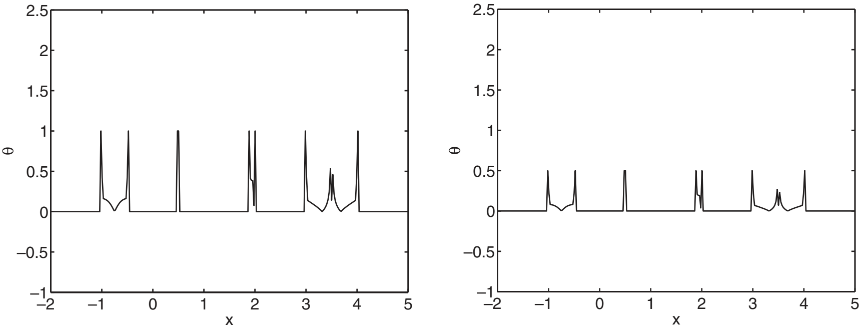


**Fig. 3.4.** ACM sensor, $\kappa = 1$ (left) and $\kappa = 0.5$ (right).

involved as in systems of hyperbolic PDEs. Here $\kappa$ is just a scaling parameter. The value 1 (0.5) corresponds to non smoothness, and the value 0 corresponds to a smooth linear solution. The function was evaluated at 300 grid points. We observe that the detector is unable to distinguish a corner ($\alpha = 1$ for the wavelet detectors) from a discontinuity ($\alpha = 0$ for the wavelet detectors) since the ACM sensor is a single scale detector. Furthermore, the smooth maximum also triggers the detector. Figure 3.4 indicates that numerical dissipation with the indicated amount will be used for all of these nonzero regions.

Next, the BS-wavelet operators (3.20) are used to compute four levels of wavelet coefficients for the function given in Fig. 3.3, evaluated at 300 grid points. A least squares fit to the line (3.15) is done. The resulting local regularity exponent $\alpha$ is plotted in Fig. 3.5. If some wavelet coefficients are equal to zero, $\alpha$ is set equal to one, since this corresponds to maximum regularity and one is the highest $\alpha$ this particular wavelet is able to predict. As seen in Fig. 3.5, a value of alpha higher than one is sometimes

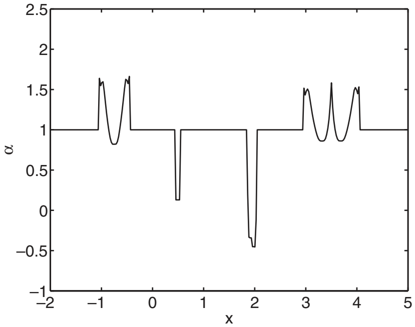


**Fig. 3.5.** BS-wavelet $\alpha$ estimate.

obtained from the numerical computation. Because of the theory, these values must not be interpreted as the real regularity being higher than one. The initial coefficients $\langle f, \phi_{0,n} \rangle$ were evaluated by numerical quadrature from function values at the grid points. In this computation $\alpha$ is equal to 0.13 at the jump, and $-0.46$ at the spike, and is close to or above one for the rest of the function.

Figure 3.6 shows the same computation, but with the RH-wavelet decomposition using operators (3.24) instead. The same function as above is decomposed into four levels of coefficients, and a least squares fit is done to find $\alpha$. The predictions are of similar type as for the BS-wavelet, but give a somewhat lower value of $\alpha$. The RH-wavelet gives the $\alpha$ value of 0.0 for the discontinuity, and $-0.65$ for the spike. We used a numerical quadrature formula for the initial coefficients, $\langle f, \phi_{0,n} \rangle = (f_j + f_{j+1})/2$. This was necessary to do in order to have an exactly correct result for step functions. In [3, 14] scaling factors, which change with the scale level, are used on the wavelet coefficients in order to renormalize for step functions.

The number of vanishing moments for the RH-wavelet is two, one more than for the B-spline wavelet. This is seen in Figs. 3.5 and 3.6, where, for the smooth regions, the RH-wavelet is reaching higher values of $\alpha$ than the B-spline wavelet.

Readers are cautioned not to compare Fig. 3.4 with Figs. 3.5 and 3.6 in a straight sense. The ACM sensor $\kappa\theta^{l}_{j+\frac{1}{2}}$ has no one-to-one correspondence with the Lipschitz exponent $\alpha$, even in the actual implementation of the wavelet sensor in the numerical scheme. Figure 3.4 shows the amount of numerical dissipation needed for the Yee *et al.* scheme to capture a solution like (3.25), whereas Figs. 3.5 and 3.6 show the regularity of (3.25). The amount of numerical dissipation determined by the wavelet sensors does not come into play until later. In a loose sense, Figs. 3.4–3.6 illustrate the

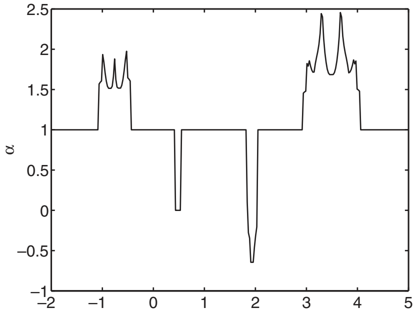


**Fig. 3.6.** RH-wavelet α estimate.

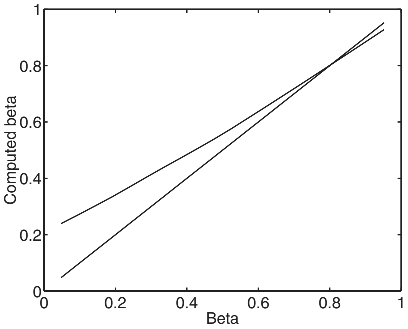


**Fig. 3.7.** BS-wavelet estimate.

fact that the ACM sensor is a single scale detector as opposed to the chosen multiple scale multiresolution wavelet sensors which are capable of detecting all of the four features of the function (3.25).

As a second example, we investigate the capability of the detectors to predict the exponent of the function $|x|^{\beta}$, where $0 < \beta < 1$. The Hölder (Lipschitz) exponent $\alpha$ obtained from the BS-wavelet decomposition (3.20) with four wavelet levels is shown in Fig. 3.7 as a function of the exact exponent $\beta$. 300 grid points were used in this example. The same quantity computed by the RH-wavelet decomposition (3.24) is shown in Fig. 3.8. In the figures the exact exponent is also plotted as a line with slope one. We note that the prediction is fairly good in the interval $0.5 < \beta < 1$, but that the computations tend to overestimate the regularity at the lower end of the interval. This is probably due to poor resolution of the singularity.

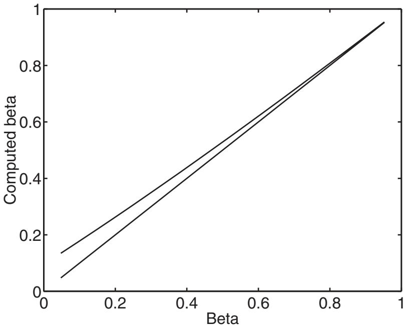


**Fig. 3.8.** RH-wavelet estimate.

## 4. WAVELET DETECTORS IN THE HIGH ORDER ACM FILTER SCHEME

In the previous section we described how to estimate the regularity (Lipschitz) exponent $\alpha$ numerically. However, we have not yet discussed how to use the information contained in the Lipschitz exponent. One possibility would be to let the order of accuracy of the numerical scheme adapt to the regularity of the function. Other possibilities are either to use an approximation adapted to the regularity of the solution similar to the $(h, p)$ finite element methods, or to integrate the wavelet sensor into a difference scheme, by using the Lipschitz exponent $\alpha$ in the same way as the switching quantity used in the high resolution shock-capturing schemes. These are subjects of ongoing investigations. In this paper, we discuss how to use the Lipschitz exponent to dynamically control the numerical dissipation. We concentrate on the simplest method of improving the Yee *et al.* ACM based filter scheme. The straightforward procedure is to insert the new estimator into the ACM based filter scheme to replace the ACM sensor. This boils down to how to adaptively switch on or gradually transition to the numerical dissipation $\tilde{\phi}^l_{j+\frac{1}{2}}$ in (2.4).

For the numerical experiments presented in Sec. 5, the wavelet sensor is obtained by computing a vector of the approximated Lipschitz exponent of a *chosen vector function to be sensed* with a suitable multiresolution non-orthogonal wavelet basis function. Here, "vectors or variables to be sensed" means the represented vectors or variables that are suitable for the extraction of the desired flow physics. The variables to be sensed can be the density and/or pressure, the characteristic variables, the jumps in the characteristic variables $\tilde{\alpha}^l_{j+\frac{1}{2}}$, or the entropy variable vector $W$ [3, 27]. The choice on the type of wavelet basis functions and their scaling functions depends on the types of features that we want to extract or detect. For the test problems to be presented, we would like the wavelet basis function and its scaling function to be capable of detecting shocks, shears, spurious oscillations and turbulence. The BS and RH wavelets are two possible choices. Study on the optimal choice of wavelet basis functions and their scaling functions is a subject of ongoing research.

For example, if the characteristic variables are the chosen vector to be sensed by the wavelet, the sensor $\mathcal{S}^l_{j+1/2}$ in (2.4) can be defined as

$$\mathcal{S}^l_{j+1/2} = \tau(\alpha^l_{j+1/2}) \tag{4.1}$$

where $\alpha^l_{j+1/2}$ is the estimated Lipschitz exponent of the $l$th characteristic component with $l = 1, 2, 3, 4$, the four characteristic waves. $\tau(\alpha)$ is a sensing function which decreases from $\tau(0) = 1$ to $\tau(1) = 0$ (for the aforementioned type of wavelet). Note that the $l$th component of the estimated

Lipschitz exponent $\alpha^l_{j+\frac{1}{2}}$ is not to be confused with the jump in the $l$th characteristic variables $\tilde{\alpha}^l_{j+\frac{1}{2}}$ in Sec. 2.

If we base instead the exponent estimate on point centered quantities, we will use the sensor function

$$\mathscr{S}^l_{j+1/2} = \max(\tau(\alpha^l_j), \tau(\alpha^l_{j+1})) \tag{4.2}$$

and if the exponent estimate is based on other quantities than the characteristic, e.g., density and pressure, we use the switch

$$\mathscr{S}_{j+1/2} = \max_l \mathscr{S}^l_{j+1/2} \tag{4.3}$$

where the maximum is taken over all components of the waves used in the estimate, and which is thus the same for all characteristic fields.

The function $\tau(\alpha)$ should be such that $\tau(0) = 1$, and $\tau(1) = 0$ or a smooth transition between 1 and 0. Three options are considered.

$$\begin{aligned} &\tau(\alpha) = \begin{cases} 1 & \alpha < \alpha_0 \\ 0 & \alpha \geqslant \alpha_0 \end{cases} \\ &\tau(\alpha) = \tfrac{1}{2} + \tfrac{1}{\pi} \operatorname{arc\,tan} K(\alpha_0 - \alpha) \\ &\tau(\alpha) = \max\{0, \min[1, (\alpha - 1)/(\alpha_0 - 1)]\}. \end{aligned} \tag{4.4}$$

Here, $\alpha_0$ is a cut off exponent to be chosen. For the arctan function the values 0 and 1 are not attained, but we take the constant $K$ large enough, so that the function is close to zero for $\alpha > 1$, and close to one for $\alpha < 0$. We have tried values for $K$ in the interval [200, 500].

The sensors obtained from (4.4), all lead to almost the same amount of dissipation when $\alpha$ is below some limit. Alternatively, one can integrate the actual $\alpha$ value into the sensor function to have a more complicated dependence, especially when the sensor is used as grid adaption sensor as well.

After some experimentation, we have found that switching on the dissipation at the grid points where $\alpha < 0.5$ works well, i.e., taking

$$\tau(\alpha) = \begin{cases} 1 & \alpha < 0.5 \\ 0 & \alpha \geqslant 0.5. \end{cases} \tag{4.5}$$

In fact the method does not seem to be very sensitive to the exact value of cut off $\alpha_0$, (for $0.4 \leqslant \alpha_0 \leqslant 0.6$) for all the test cases considered. Furthermore, the same cut off value, 0.5, works well for all problems we have tried in Sec. 5 (except for the vortex convection case, where $\alpha_0 = 0$ should be used in conjunction with entropy splitting [27]). Experiments with smoothed step functions do not give very different results. To distinguish the high order ACM based filter scheme, we referred the scheme discussed in Sec. 2 using the wavelet sensor (4.1)–(4.2) as the *wavelet based filter scheme*.

We would like to point out that the simple minded wavelet sensor (4.4) or (4.5) does not make full use of the Lipschitz exponent. In practice, a better strategy is to have distinct sensors for distinct flow structures. This might involve the use of more than one type of wavelet and filters (cf. Yee *et al.*, Yee and Sjögreen, and Sjögreen and Yee [20, 26, 27] for some discussion). Regardless of the amount of information on the Lipschitz exponent used to design the wavelet sensor, the sensor is not really parameter free. Unlike the ACM sensor, the parameter involved in the wavelet sensor is, however, not arbitrary. There are theorems and guidelines on what values of the Lipschitz exponent are to be expected for various features of the function to be sensed.

## 5. NUMERICAL EXPERIMENTS FOR 2-D COMPRESSIBLE EULER AND NAVIER–STOKES EQUATIONS

To illustrate the performance of the wavelet sensor, the same three perfect gas test cases with distinct flow properties as in Yee *et al.* [25, 27] are used. The first is inviscid and the last two are compressible full Navier–Stokes computations. The three test cases are: (1) a 2-D inviscid horizontally convecting vortex with periodic BCs, (2) a 2-D vortex pairing in a time-developing mixing layer with shock waves formed around the vortices, and (3) a 2-D shock wave impinging on a spatially evolving mixing layer where the evolving vortices must pass through a shock wave, which in turn is deformed by the vortex passage. Figures 5.1–5.3 (taken from [25, 27]) show the schematic, flow conditions and the computational domains of the three test cases.

In order to compare results with Yee *et al.* [25, 27], we use the same time and spatial base scheme as in [25, 27]. The classical fourth-order Runge–Kutta time discretization, and the non-compact central spatial schemes with the same order of accuracy and type of base scheme for the inviscid and viscous terms (if viscosities are present) are employed. The filters are applied at the end of the full Runge–Kutta time step. Roe's [17] average states are used in (2.3), along with the Harten and Yee ([23, 24]) second-order upwind TVD dissipation portion (2.5)–(2.7) for $\tilde{\phi}^l_{j+\frac{1}{2}}$ in (2.4). From numerical experiment, limiter (2.7a) produces the best result for test case 1 and limiter (2.7c) for test cases 2 and 3. Results shown in Sec. 5.2 reflect these choices. For the ACM sensor the parameters $p$ and $m$ in

## Isentropic Vortex Evolution

*(Horizontally Convecting Vortex, vortex strength $\hat{\beta}$=5)*

**Freestream:**

$$(u_\infty, v_\infty) = (1,0);\ p_\infty = \rho_\infty = 1$$

**IC:** Perturbations are added to the freestream (not in entropy)

$$(\delta u, \delta v) = \frac{\hat{\beta}}{2\pi} e^{\frac{1-r^2}{2}} (-(y-y_0), (x-x_0))$$

$$\delta T = -\frac{(\gamma-1)\hat{\beta}^2}{8\gamma\pi^2} e^{1-r^2}$$

$$r^2 = (x-x_0)^2 + (y-y_0)^2$$

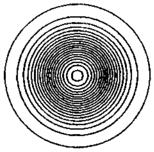
**Computational Domain & Grid Size:**

$$0 \le x \le 10\ \&\ -5 \le y \le 5$$

$80 \times 79$ Uniform grid

Periodic BC in $x$ & $y$

Initial Vortex, Density Contours



$(x_{v_0}, y_{v_0})$
center of vortex

Fig. 5.1. Test problem 1, isentropic vortex convection.

## Vortex Pairing in a Time-Developing Mixing Layer

*($M_c$=0.8, Re=1000, $T_R$=300K, Prandtl #=0.72)*

Normalized with vorticity thickness:

$$\delta_\omega = \frac{u_1 - u_2}{(du/dy)_{max}}$$

T & c: determined by assuming constant stagnation enthalpy

**IC:**

$u_1$=1, $u_2$=-1, $T_1$=$T_2$,

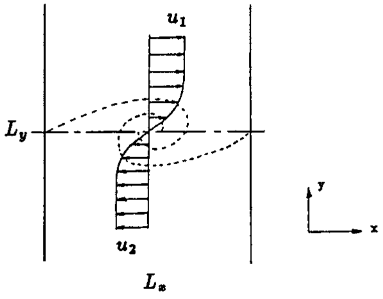
Inital shear profile: u = 0.5 tanh (2y)

Crocco-Busemann:

$$c^2 = c_1^2 + \frac{\gamma-1}{2}(u_1^2 - u^2)$$

Initial perturbations:

$$v' = \sum_{k=1}^{2} a_k \cos(2\pi k x/L_x + \phi_k) e^{(-y^2/b)}$$



$$L_x = 30, b = 10$$
$$a_2 = 0.05, \phi_2 = -\pi/2$$
$$a_1 = 0.01, \phi_1 = -\pi/2$$

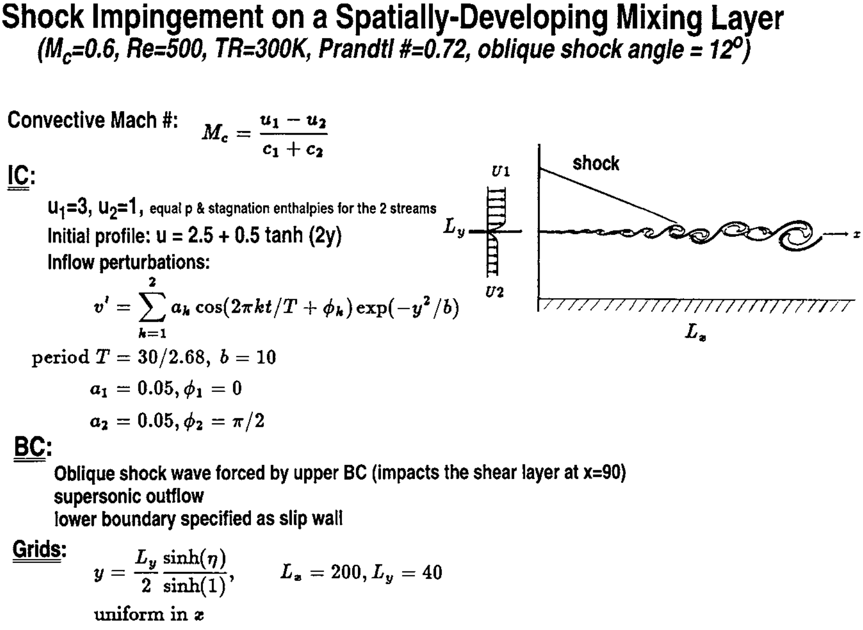
**BC:** Periodic in x, slip walls in y

**Grids:**

$$y = \frac{L_y}{2}\frac{\sinh(b_y \eta)}{\sinh(b_y)},\quad L_y = 100,\ b_y = 3.4$$

uniform in $x$

Fig. 5.2. Test problem 2, vortex pairing.



**Shock Impingement on a Spatially-Developing Mixing Layer**

*($M_c$=0.6, Re=500, TR=300K, Prandtl #=0.72, oblique shock angle = 12°)*

Convective Mach #: $M_c = \frac{u_1 - u_2}{c_1 + c_2}$

**IC:**

$u_1=3$, $u_2=1$, equal p & stagnation enthalpies for the 2 streams

Initial profile: u = 2.5 + 0.5 tanh (2y)

Inflow perturbations:

$$v' = \sum_{k=1}^{2} a_k \cos(2\pi kt/T + \phi_k)\exp(-y^2/b)$$

period $T = 30/2.68$, $b = 10$

$a_1 = 0.05, \phi_1 = 0$

$a_2 = 0.05, \phi_2 = \pi/2$

**BC:**

Oblique shock wave forced by upper BC (impacts the shear layer at x=90)
supersonic outflow
lower boundary specified as slip wall

**Grids:** $y = \frac{L_y}{2}\frac{\sinh(\eta)}{\sinh(1)}$, $L_x = 200, L_y = 40$

uniform in $x$

**Fig. 5.3.** Test problem 3, shock impingement on a mixing layer.

(2.8) and (2.9) are set to 1 and a small value of $10^{-6}$ is added to the denominator of (2.9) to avoid an extra logical statement for the ACM sensor. For the wavelet sensor, the cut off Lipschitz exponent $\alpha_0$ is 0.5 for test cases 2 and 3 and 0.0 for test case 1. The reason for using $\alpha_0 = 0.0$ for test case 1 is that there are no shock waves present. The sensor is used merely to suppressed high frequency producing nonlinear instability associated with the central base scheme in this long time wave propagation phenomena. These various numerical methods will be denoted as ACM or WAV (depending on whether an ACM or wavelet is used as the sensor) with the following numbers indicating the order of the spatial interior base scheme for the inviscid and viscous terms. For example, ACM66 (WAV66) means the use of sixth-order central as the base scheme for both the inviscid and viscous terms, and ACM as sensor (wavelet as sensor). These wavelet sensors are computed using the dimension by dimension method as discussed in Sec. 3.5. In order not to introduce additional notation, inviscid flow simulations are designated by the same notation, with the viscous terms not activated. Computations using the B-spline and the redundant form of Harten's multiresolution method will be denoted as "BS" and "RH" as in WAV66-BS and WAV66-RH. Computations using entropy splitting are indicated by adding the letters ENT as in

ACM66-ENT, WAV66-ENT-BS or WAV66-ENT-RH. Computation using $\mathscr{S}^l_{j+\frac{1}{2}} = 1$, i.e., the sensor is turned off and the full amount from the upwind TVD dissipation portion is used as the filter, will be denoted by TVD as in TVD66. The fifth-order WENO scheme of Jiang and Shu [7] will be denoted as WENO5.

For the second and third test cases, we lower the order of the base scheme near the boundary points for the boundary scheme. For example, for the sixth-order base scheme, we use a fourth-order central scheme two points away from the boundary point and second-order central scheme one point away from the boundary point. For the third test case, for simplicity, slip wall BC is used for the lower wall, and the upper $y$-direction physical BC is over specified and nonreflecting BC is not used. A uniform Cartesian grid of $80 \times 79$ is used for test case 1. For test cases 2 and 3, a uniform Cartesian grid is used in the $x$-direction and a mildly stretched Cartesian grid is used in the $y$-direction with the grid size of $101 \times 101$ and $321 \times 81$ respectively.

## 5.1. Comparison of the Wavelet Sensor with the ACM Sensor

Before showing the comparison of the high order wavelet based filter scheme with the high order ACM based filter scheme, we first show the performance of the wavelet sensor using the dimension-by-dimension approach for a 2-D complex flow structure and then compare loosely with the ACM sensor. The initial illustration does not involve the numerical scheme and only the performance of the wavelet sensor is demonstrated. Fig. 5.4 shows the computed density and pressure contours at $t = 120$ by

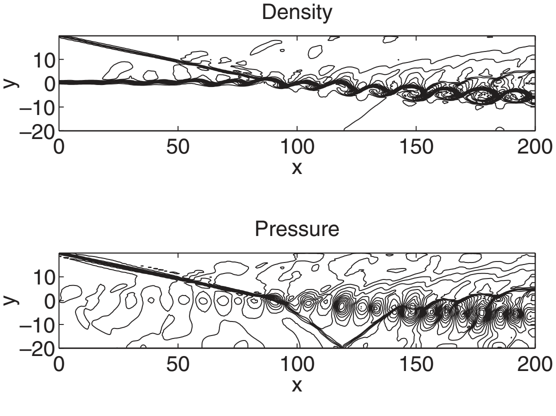


**Fig. 5.4.** 2-D Testing discrete function, (density and pressure contours at $t = 120$ from test case 3).

WAV66-RH with $\Delta t = 0.12$ for test case 3. Here we only consider these numerical data as a given two-dimensional discrete function to be analyzed by the wavelet algorithm. The function represents a shock from the upper left corner, impinging on a horizontal shear layer in the middle of the domain (see Fig. 5.3). The shock is reflected from the lower wall boundary. For more details about the problem, see Yee *et al.* [25, 27].

Figure 5.5 shows contours of the estimated Lipschitz exponent $\alpha$ for the function in Fig. 5.4. The value $\alpha$ was computed here from three levels ($m_0 = 3$) of the wavelet algorithm, using the wavelet coefficient

$$w_{m,j,k} = \sqrt{(w^x_{m,j,k})^2 + (w^y_{m,j,k})^2}$$

where the one dimensional coefficients were computed by the multiresolution operators (3.24) in each coordinate direction. The coefficients were computed for the pressure. The top figure in Fig. 5.5 shows $\alpha$ contours on levels from 0.5 to 0.9. The lower figure shows the corresponding sensor, a function which is one for $\alpha < 0.5$ and zero otherwise. The wavelet sensor clearly captures the shock and the shear layer. The low $\alpha$ at the upper boundary to the right is probably due to mildly unstable boundary conditions at the upper boundary.

We want to emphasize that Fig. 5.5 shows the sensor when applied to a precomputed solution at a fixed time. No dynamic behavior was involved (i.e., the numerical scheme is not part of the analysis). Since the ACM sensor has no one-to-one correspondence counterparts of Fig. 5.5, no results are shown for the ACM sensor. Next we show in Figs. 5.6 and 5.7

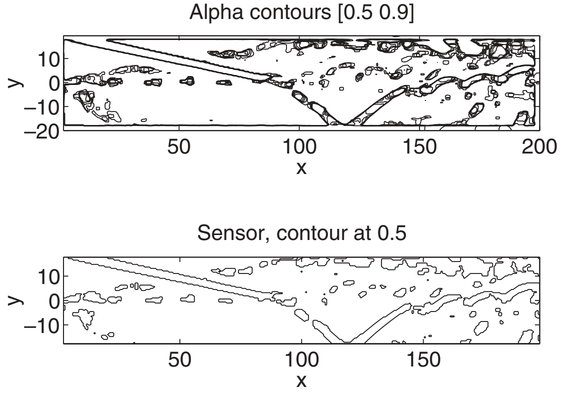


Fig. 5.5. Top: $\alpha$ contours $0.5 \leqslant \alpha \leqslant 0.9$; Bottom: sensor contour at $\alpha = 0.5$. by the RH-wavelet.

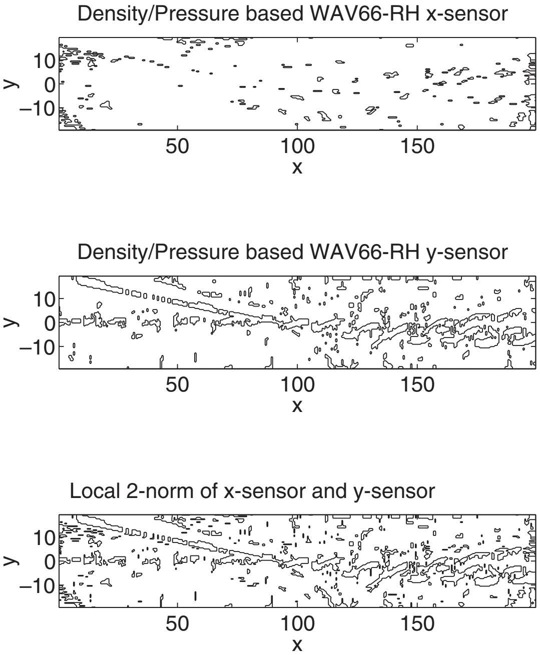


**Fig. 5.6.** One contour at $\alpha = 0.5$ of the sensors used by WAV66-RH applied to the density and pressure of test case 3.

results from actually computing the flow using the respective wavelet based (WAV66-RH) and ACM based (ACM66) filter schemes for $\Delta t = 0.12$. Figure 5.6 shows the wavelet sensor applied to the density and pressure at $t = 120$ in the $x$- and $y$-directions, and the square root of the sum of these quantities (local 2-norm) in the $x$- and $y$-directions.

Figure 5.7 shows the corresponding contours using the ACM sensor with $\kappa = 0.35$. There is only one contour level plotted. The level value is in the middle of the range, i.e., at the average of the maximum and the minimum sensor values. The wavelet sensor was able to extract the full features of the flow structure far better than the ACM sensor. Although this is the case, as we can see later, the wavelet sensor exhibits accuracy similar to the best tuned ACM sensor. This is due partly to the fact that in actual implementation formulas (4.4)–(4.5) are used. We are not making full use of the Lipschitz exponent.

Again one cannot compare the ACM sensor and the wavelet sensor directly on these figures. The wavelet has more flexibility and choices whereas the ACM sensor only compares the strength of gradients ($m = p = 1$ in (2.8) and (2.9)) between neighboring grid points of a chosen

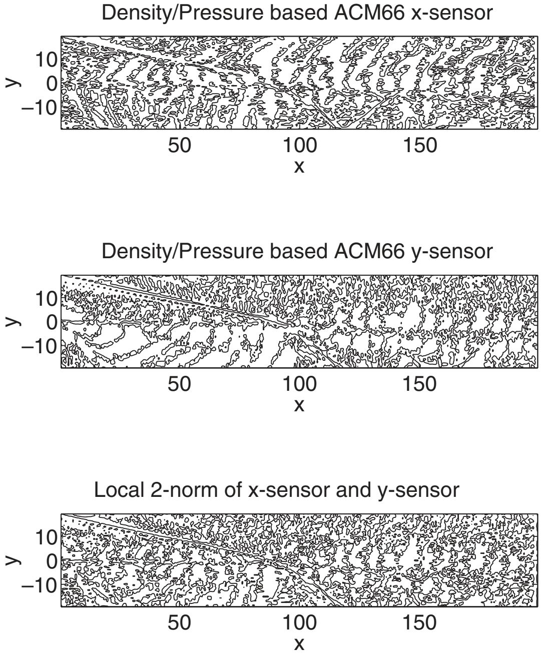


**Fig. 5.7.** One contour of the sensor, $\mathcal{S}^l_{j+\frac{1}{2}}$ used by ACM66 applied to the density and pressure of test case 3.

physical quantity (or vector). Perhaps the comparisons would be more relevant if we were to compare the ACM sensor using different $p$ and $m$ values in (2.8) and (2.9) for the different flow features. This involves additional switching parameters and is not pursued here.

## 5.2. Comparison Among TVD66, WENO5, ACM66, and WAV66

Sample computations using the high order wavelet based filter scheme WAV66 compared with the high order ACM based filter scheme ACM66 for test cases 1–3 are shown in Figs. 5.8–5.11. The accuracy of the two wavelet sensors, B-spline wavelet (WAV66-BS) or the redundant form of Harten wavelet (WAV66-RH) for test cases 1–3 (results not shown) is very similar and the effect on accuracy of the choice of the physical vector to be sensed (density and/or pressure, characteristic variables, $\tilde{\alpha}^l_{j+\frac{1}{2}}$, or entropy variables $W$) is not pronounced. In all cases, no physical problem-dependent parameter has to be tuned. The accuracy compared very well with that of the corresponding best tuned $\kappa$ for ACM66 for the individual test

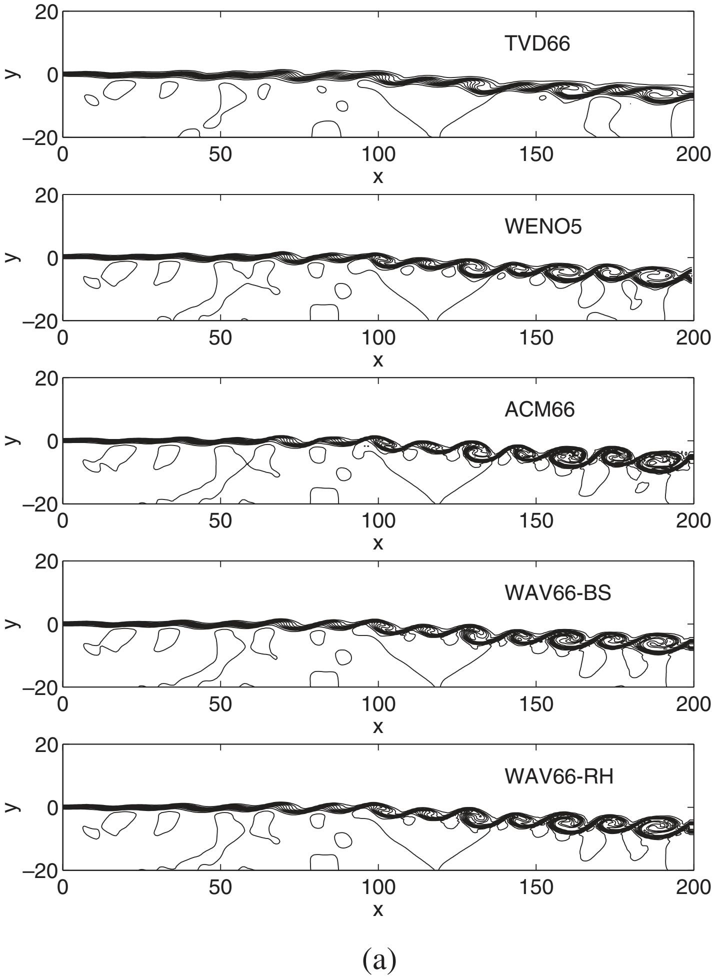


**Fig. 5.8.** (a) Normalized temperature contours for test case 3 at $t = 113.16$ using TVD66, WENO5, ACM66, WAV66-BS, and WAV66-RH on a $321 \times 81$ grid.

cases 1–3. In particular, similar accuracy was sustained using the redundant form of Harten wavelet sensor and entropy splitting (WAV66-ENT-RH) for long time integrations of the vortex convection problems as ACM66-ENT using $\kappa = 0.01$ and $\Delta t = 0.01$.

Figure 5.8 shows the comparison among TVD66 ($\mathscr{S}^{l}_{j+\frac{1}{2}} = 1$), WENO5, ACM66, and WAV66-RH for test case 3, illustrating the normalized temperature and pressure contours at $t = 113.16$ with $\kappa = 0.35$ for the nonlinear

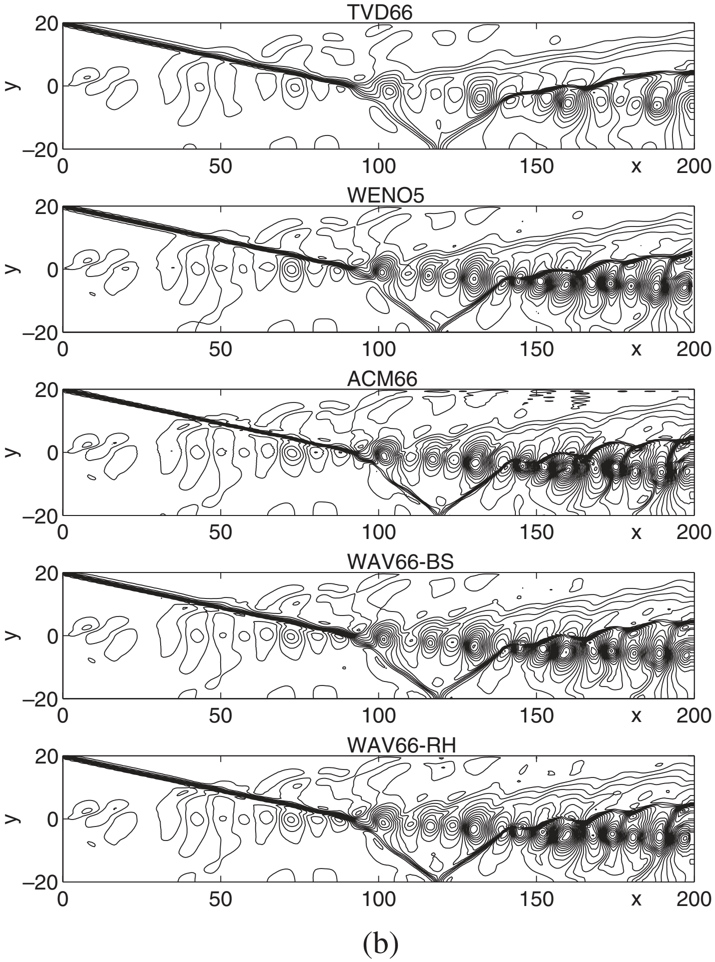


**Fig. 5.8.** (b) Pressure contours for test case 3 at $t = 113.16$ using TVD66, WENO5, ACM66, WAV66-BS, and WAV66-RH on a $321 \times 81$ grid.

fields and $\kappa = 0.175$ for the linear fields for the ACM66. For this set of tuned $\kappa$'s, the solution obtained is very accurate and without visible instability. The solution is comparable with the WAV66-RH. For test cases 2 and 3, only 50% of the wavelet sensor is applied to the linear fields (i.e., 50% numerical dissipation). The resolution of the WAV66-RH is more diffusive if full strength is applied to the linear fields. Observe that the wavelet sensor was able to remove the noise generated on the upper boundary due

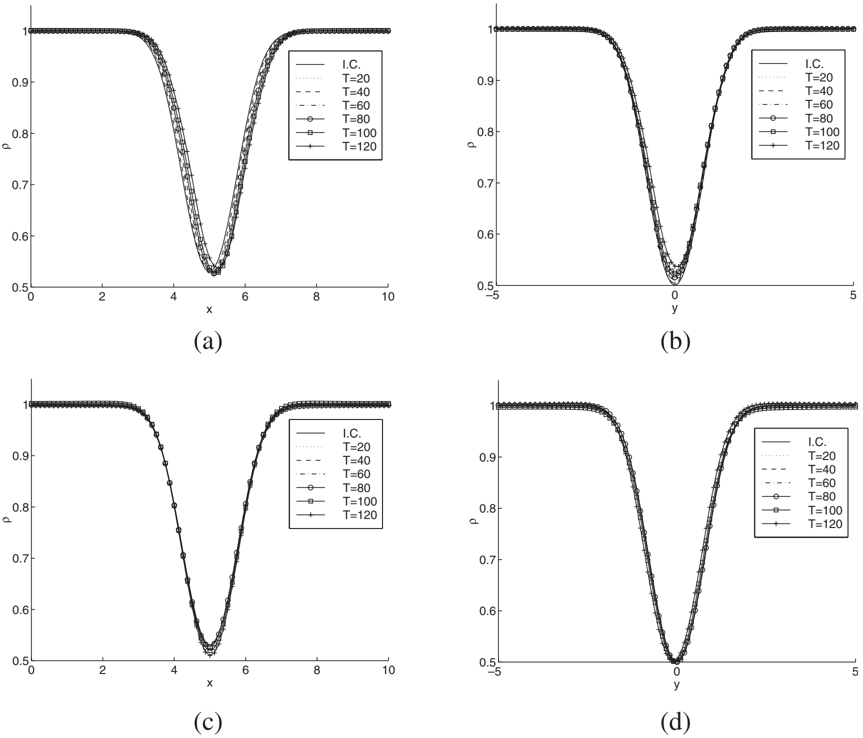


**Fig. 5.9.** (a) density at the line $y = 0$ at $T = 20$ period increment for test case 1, ACM66-ENT on a $80 \times 79$ grid, (b) density at the line $x = 5$ at $T = 20$ period increment for test case 1, ACM66-ENT on a $80 \times 79$ grid, (c) density at the line $y = 0$ at $T = 20$ period increment for test case 1, WAV66-ENT-RH on a $80 \times 79$ grid, (d) density at the line $x = 5$ at $T = 20$ period increment for test case 1, WAV66-ENT-RH on a $80 \times 79$ grid.

to the over specified BC. Note that the normalized temperature is the most sensitive value to examine for accuracy of the schemes [25]. By examining temperature contours, we note that the vortices are more diffusive in the WENO5 computations. There is a minor difference between the WENO5 computation and the other simulations. The WENO5 code has a built-in nonreflecting BC on the upper $y$-direction. The WENO5 also requires more arithmetic operations than the ACM66 or WAV66.

The long time wave propagation of the inviscid vortex convection problem in test case 1 poses a different challenge to the numerical method. For long time wave propagation of this nature, non-dissipative or low

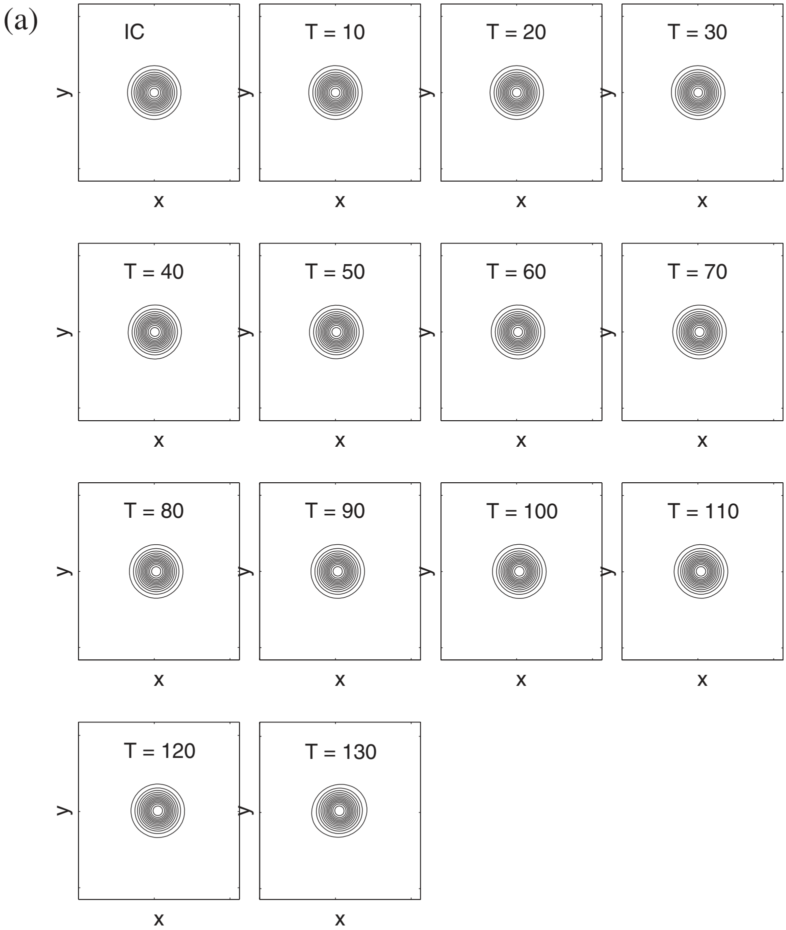


**Fig. 5.10.** (a) Density contours at $T = 10$ period increment for test case 1, ACM66-ENT on a $80 \times 79$ grid.

dissipative schemes usually exhibit high frequency oscillation nonlinear instability at quite early stages of the wave propagation process. Although numerical dissipation can suppress the high frequency oscillation, if applied incorrectly, the vortex becomes very diffuse at a longer time integration and eventually nonlinear instability sets in. Figures. 5.9 and 5.10 show the long time wave propagation comparison between ACM66-ENT and WAV66-ENT-RH for test case 1. Figures 5.9 shows the density profiles at the centerlines $y = 0$ and at $x = 5$, cutting through the center of the initial

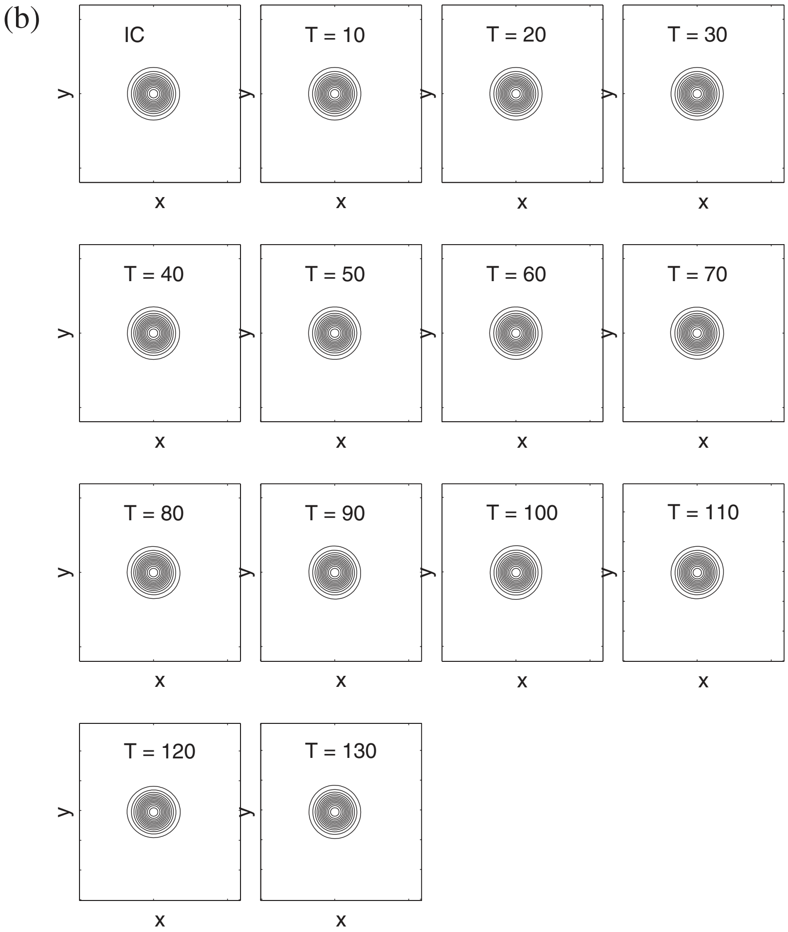
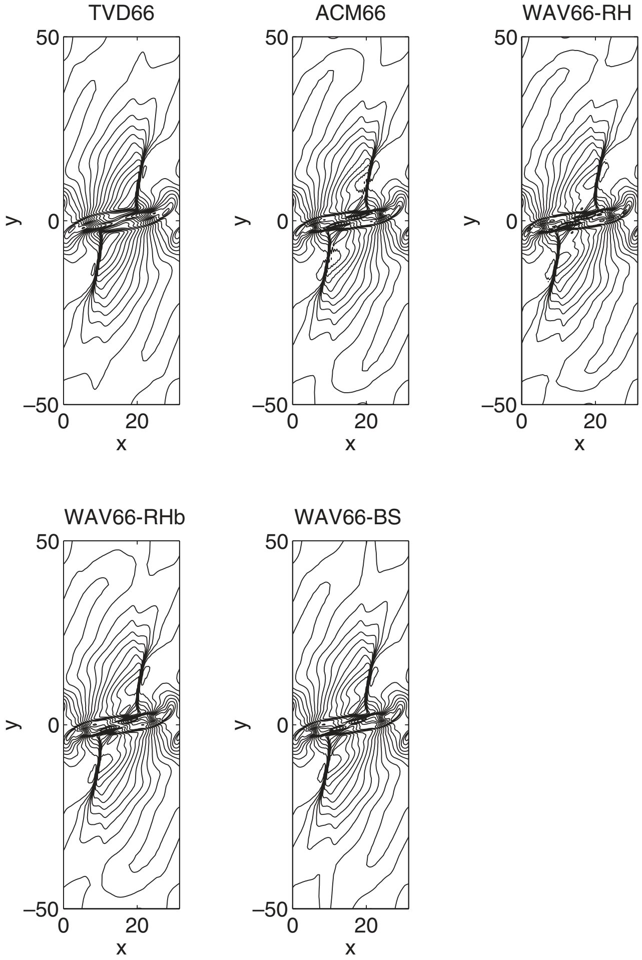


**Fig. 5.10.** (b) Density contours at $T = 10$ period increment for test case 1, WAV66-ENT-RH on a $80 \times 79$ grid.

vortex, at 20 spatial period increments. The time required for one spatial period is $T = 10$.

The time step and grid spacing are $\Delta t = 0.01$ and $80 \times 79$. Depending on the time and spatial discretizations, the grid size and time step, the vortex can drift away from the centerline after long time integrations. For the present two methods with the indicated time step and the uniform grid spacings, there is only a very slight drift of the vortex after a very long time



TVD66

ACM66

WAV66-RH

WAV66-RHb

WAV66-BS

**Fig. 5.11.** Normalized temperature contours for test case 2 using TVD66, ACM66, WAV66-RH, WAV66-RHb, and WAV66-BS on a 101 × 101 grid. WAV66-RHb is same as WAV66-RH but with 50% less filter dissipation on the linear fields.

integration (130 periods). If the computed vortex drifts away from the centerline but still preserves the vortex shape and strength, the centerline density profiles do not convey the full information and can be misleading. We complement the comparison with snap shots of density contours at different times up to 130 spatial periods in Fig. 5.10.

The vortex is convected 10 spatial periods between each plot. The result from the WAV66-ENT-RH method is at least as good as the

ACM66-ENT method. Here the base scheme is applied to the entropy split form of the inviscid flux derivative, in order to reduce effects from nonlinear instabilities. The results using the same condition and parameters, but with no entropy splitting of the inviscid flux derivatives, although very stable, exhibit smearing of the vortex and severe vertical and horizontal drifts for both ACM66 and WAV66-RH. See [27] for the ACM66 and ACM66-ENT comparison. The use of entropy splitting in conjunction with ACM66 (ACM66-ENT) or WAV66-RH (WAV66-ENT-RH) has preserved a horizontally convecting vortex with great accuracy after long time integration of 130 periods. The results use a uniform and not very fine grid. To the authors' knowledge, highly accurate numerical simulation of this problem previously reported in the literature were only carried out up to 10 periods of integration.

We would like to point out that the vertical and horizontal drifting (or rather shifting) of the vortex away from the centerline $y = 0$ and/or $x = 5$ is quite common for all schemes beyond 30 periods. Depending on the scheme, the amount of numerical dissipation and the time step, drifting can occur as early as 5 periods [27]. We believe that the drifting is due largely to the spatial numerical dissipation of the scheme provided a highly accurate low phase error time integrator is used. We also would like to point out that an eighth-order linear filter actually performs better (almost perfect vortex preservation after 200 periods [20]) than the nonlinear filter used here that is designed for shock-capturing. The main purpose here is to show that the nonlinear filter is doing quite a good job.

Figure 5.11 shows the comparison among TVD66, ACM66, WAV66-RH, and WAV66-BS for test case 2 with $\Delta t = 0.1$. Here two different versions of the WAV66-RH method are examined. The one denoted WAV66-RH in the figure has 50% reduced TVD dissipation on the linear fields. The resolution of WAV66-RH and WAV66-RHb is slightly more accurate than WAV66-BS. The result using WENO5 (not shown) is less accurate than ACM66 but more accurate than TVD66.

## 6. CONCLUDING REMARKS

Improved adaptive numerical dissipation controls over the Yee *et al.* ACM sensor have been constructed. The new sensors with improved detection properties are derived from multiresolution wavelet based analysis and require slightly more operations count than the ACM sensor. There are a variety of wavelets to choose from, depending on the flow feature. Developing the best suited adaptive numerical dissipation/filter control for a particular flow speed and flow structure might involve the switching on and off and/or blending of different filters by automatic sensors

[27, 20, 26]. This method, however, is superior to the tuning of parameter, and the use of the same filter, and especially the same amount of numerical dissipation throughout the entire domain.

We considered two types of non-orthogonal wavelet basis functions for our 2-D compressible Euler and Navier–Stokes numerical experiments. One is similar to the B-spline wavelet (Mallat and Zhong [14]) used by Gerritsen and Olsson [3] for grid adaptation and the other is a modification of the multiresolution method of Harten [5] as a redundant multiresolution wavelet. The B-spline wavelet sensor requires slightly more arithmetic operations and a wider grid stencil than the redundant form of Harten wavelet sensor. The final form of the wavelet sensor $\mathscr{S}^l_{j+\frac{1}{2}}$ involves mainly nested difference operators and least squares fits. From the numerical experiments, it appears that the RH-wavelet sensor exhibits a slightly more accurate result than the BS-wavelet sensor. The proposed wavelet sensors, unlike the ACM sensor, can detect most of the distinct flow features, including turbulence, leading to an automatic selection of the appropriate distribution of numerical dissipation. Since distinct Lipschitz exponent values represent distinct flow structures, these wavelet sensors are free of physical problem-dependent arbitrary parameters for the three test cases presented. They are also good grid adaptation indicators [3] when compared to the ones commonly used in practice. Consequently, a new dual purpose adaptive method is readily available leading to dynamic numerical dissipation controls and improved grid adaptation indicators. This dual purpose adaptive method can also serve as a stand alone option for other numerical schemes.

In the future, we will explore the full capability of the multiresolution wavelet property. This will include improved wavelet basis functions and their scaling functions for high speed compressible shock-turbulence interaction and numerical combustion, and an improved switching function other than the one proposed in Sec. 4. In other words, a better use of the Lipschitz exponent information will be implemented. Another possibility is to use an approximation adapted to the regularity of the solution similar to the $(h, p)$ finite element method.

## ACKNOWLEDGMENTS

The authors would like to thank Marcel Vinokur and Tom Coakley for their critical review of the manuscript. Special thanks to the help of A. Hadjadj for the WENO5 computations. The first author would like to acknowledge the support from the Swedish Natural Science Research Council (NFR). This paper was published as a RIACS Technical Report

01.01, October 2000, NASA Ames Research Center; Part of this work was carried out while the first author was a visiting scientist with RIACS, NASA Ames Research Center.

## REFERENCES

1. Daubechies, I. (1992). *Ten Lectures on Wavelets*, CBMS-NSF regional conference series in applied mathematics, No. 61, SIAM.
2. Farge, M. (1992). Wavelet transforms and their applications to turbulence. *Ann. Rev. of Fluid Mech.* **24**, 395–457.
3. Gerritsen, M., and Olsson, P. (1996). Designing an efficient solution strategy for fluid flows. *J. Comput. Phys.* **129**, 245–262.
4. Harten, A. (1978). The artificial compression method for computation of shocks and contact discontinuities: III self-adjusting hybrid schemes. *Math. Comp.* **32**, 363–389.
5. Harten, A. (1995). Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.* **48**, 1305–1342.
6. Harten, A., and Hyman, J. (1983). Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.* **50**, 235–269.
7. Jiang, G., and Shu, C.-W. (1996). Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**, 202–228.
8. LeVeque, R. J., and Yee, H. C. (1990). A study of numerical methods for hyperbolic conservation laws with stiff source terms. *J. Comput. Phys.* **86**, 187–210.
9. Mallat, S. G. (1999). *A Wavelet Tour of Signal Processing*, 2nd ed., Academic Press, San Diego.
10. Mallat, S. G. (1989). Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. Acoust. Speech Signal Process.* **37**, 2091–2110.
11. Mallat, S. G. (1989). Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.* **315**, 69–87.
12. Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Anal. Mach. Intell.* **11**, 674–693.
13. Mallat, S. G., and Hwang, W. L. (1992). Singularity detection and processing with wavelets. *IEEE Transactions on Information Theory* **38**, 617–643.
14. Mallat, S. G., and Zhong, S. (1992). Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 710–732.
15. Minh, D.-K., Basdevant, C., Perrier, V., and Khoa, D.-T. (1994). Wavelet analysis of 2-D turbulent fields, *Physica D* **76**, 252–277.
16. Perrier, V., Philipovitch, T., and Basdevant, C. (1999). *Wavelet Spectra Compared to Fourier Spectra*, Publication of ENS, Paris.
17. Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**, 357–372.
18. Sandham, N. D., and Yee, H. C. (2000). *Entropy Splitting for High Order Numerical Simulation of Compressible Turbulence*, RIACS Technical Report 00.10, June 2000, NASA Ames Research Center; Proceedings of the first international conference on CFD, July 10–14, 2000, Kyoto, Japan.
19. Sjögreen, B. (1995). Numerical experiments with the multiresolution scheme for the compressible euler equations. *J. Comput. Phys.* **117**, 251–261.
20. Sjögreen, B., and Yee, H. C. (2002). *Analysis of High Order Difference Methods for Multiscale Complex Compressible Flows*, Proceedings of the ninth international conference on hyperbolic problems, March 25–29, 2002, Pasadena, California.

21. Strang, G. (1989). Wavelet and dilation equations: A brief introduction. *SIAM Rev.* **31**, 614–627.
22. Strang, G. (1993). Wavelet transforms versus fourier transforms. *Bull. Amer. Math. Soc. (N.S.)* **28**, 288–305.
23. Yee, H. C. (1987). *Upwind and Symmetric Shock-Capturing Schemes*, NASA TM-89464, May 1987.
24. Yee, H. C. (1989). *A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods*, VKI lecture series 1989-04, March 6–10, 1989, also NASA TM-101088, Feb. 1989.
25. Yee, H. C., Sandham, N. D., and Djomehri, M. J. (1999). Low dissipative high order shock-capturing methods using characteristic-based filters. *J. Comput. Phys.* **150**, 199–238.
26. Yee, H. C., and Sjögreen, B. (2001). *Adaptive Numerical Dissipation Control for High Order Methods*, Proceedings of the 3rd internat. AFOSR conf. on DNS and LES, Arlington, Texas, August 5–9, 2001.
27. Yee, H. C., Vinokur, M., and Djomehri, M. J. (2000). *Entropy Splitting and Numerical Dissipation*, NASA technical memorandum 208793, August, 1999, NASA Ames Research Center; *J. Comput. Phys.* **162**, 33–81.
28. Yee, H. C., Sjögreen, B., Sandham, N. D., and Hadjadj, A. (2000). *Progress in the Development of a Class of Efficient Low Dissipative High Order Shock-Capturing Methods*, RIACS Technical Report 00.11, June, 2000; Proceedings of the CFD for the 21st Century, July 15–17, 2000 Kyoto, Japan.