SCI INSTITUTE

TECHNICAL REPORT

Data Bounded Polynomials and Preserving Positivity in High Order ENO and WENO Methods

M. Berzins

UUSCI-2009-003

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

July 24, 2009

Abstract:

The positivity and accuracy properties of the widely used ENO and WENO methods are considered by undertaking an analysis based upon data-bounded polynomial methods. The positivity preserving approach of Berzins based upon data-bounded polynomial interpolants is generalized to arbitrary meshes. This makes it possible to prove positivity conditions for ENO Methods by using a derivation based on such bounded polynomial approximations. Numerical examples are used to show that although high order methods may be used in a way that preserves positivity, care must be taken in terms of resolving shock-like features with a fine enough mesh for high-order approximations to be effective.

THE UNIVERSITY OF UTAH

Data Bounded Polynomials and Preserving Positivity in High Order ENO and WENO Methods

M. Berzins*

July 24, 2009

Abstract

The positivity and accuracy properties of the widely used ENO and WENO methods are considered by undertaking an analysis based upon data-bounded polynomial methods. The positivity preserving approach of Berzins based upon data-bounded polynomial interpolants is generalized to arbitrary meshes. This makes it possible to prove positivity conditions for ENO Methods by using a derivation based on such bounded polynomial approximations. Numerical examples are used to show that although high order methods may be used in a way that preserves positivity, care must be taken in terms of resolving shock-like features with a fine enough mesh for high-order approximations to be effective.

Key words: Hyperbolic equations, ENO, positivity, data-bounded polynomials
[1]

1 Introduction

This report considers positivity preserving methods for hyperbolic equations, by combining two distinct areas of work. The first is the substantial and influential body of work on ENO and WENO methods. The second area is work on data-bounded polynomial interpolants. The overall aim is to make it possible to derive methods for many physical problems such as the solution to hyperbolic equations in which the computed solution values should, on physical grounds, remain non-negative e.g. the advection equation with non-negative initial data as given by

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{1}$$

with appropriate boundary conditions on a spatial interval $[A,B]$, and an exact solution $u(x,t)$.

In spite of the influential and substantial body of work on ENO and WENO methods [10, 11, 16, 21, 18], there are still a few unresolved theoretical questions with regard to these methods. An early modification to the ENO approach to enable the method to be TVD is provided in the preprint of Shu [19]. A slightly different approach so as to keep the ENO stencil closer to a linearly stable stencil is also described by Shu [22]. Balsara and Shu [2] use the method of Suresh and Huynh [23], to construct sixth order schemes. Shu points out in a recent survey of WENO methods it is difficult to genralize analysis of some of the methods beyond third order, [20].

The approach taken here is different in that it involves the use of polynomials whose higher divided differences may be written as bounded multiples of lower divided differences. This will be seen to be important in deriving schemes with positivity preserving properties. The algorithm used will limit the signs and growth of divided difference terms to arrive at bounded monotone polynomial approximations of potentially arbitrarily high degree within an interval. While this limiting process may be used with any divided difference polynomial, its use in conjunction with ENO and WENO schemes is natural in that both approaches seek to control the size of the differences used in the schemes. The use of polynomials whose higher divided differences may be written as bounded multiples of lower divided differences will be seen to be important in deriving schemes with positivity preserving properties. The overall intention is to derive conditions under which ENO and WENO methods are positive with regard to the

[1] *SCI Institute, School of Computing, University of Utah, Salt Lake City, Utah

standard definition used here for a positivity preserving scheme for the advection equation. This definition requires (see Laney [15]) that the numerical solution at time t_{n+1} be written in terms of the numerical solution at time t_n in the form

$$U_i(t_{n+1}) = \sum_j a_j U_j(t_n) \text{ where } \sum_j a_j = 1, \text{ and } a_j \geq 0 . \tag{2}$$

or in semi-discrete form as:

$$\dot{U}_i(t_{n+1}) = \sum_j b_j (U_j(t_n) - U_i(t_n) \text{ where } \sum_j b_j = B, \text{ and } b_j \geq 0 . \tag{3}$$

The key observation with regard to preserving positivity is due to Godunov [6] who proved that any scheme of better than first order which preserves positivity for the advection equation must be nonlinear. For example, the coefficients a_j in (2) above must depend on the numerical solution to the p.d.e. For a recent discussion of this topic see [3]. In obtaining such results for the methods considered here the first step is to prove that the data-bounded polynomial approximation derived by Berzins [1] is also data-bounded on non-uniform meshes. This is done in Section 2 of this paper. These results then make it possible to prove results about positivity in Section 3. Numerical results on three test problems in Section 4 show that it is possible to use higher orders than is often done, but that it is important to resolve features such as steep waves with enough mesh points for high-order ENO methods to be effective.

2 ENO Divided Difference Polynomial Interpolation

In common with the standard treatments of ENO and WENO methods e.g. see [18, 20], the divided difference form of polynomial interpolation is used here as it enables the unified treatment of polynomial approximations based on any set of spatial points. In this paper we will use divided differences as defined by the usual notation where $U[x_i] = U(x_i)$ and

$$U[x_i, x_{i+1}] = \frac{U[x_{i+1}] - U[x_i]}{x_{i+1} - x_i}, \tag{4}$$

and subsequent differences are defined recursively by

$$U[x_i, x_{i+1}, ..., x_{i+k}] = \frac{U[x_{i+1}, x_{i+2} ..., x_{i+k}] - U[x_i, x_{i+1}, ..., x_{i+k-1}]}{x_{i+k} - x_i}. \tag{5}$$

Suppose that a set of mesh points are given by $x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4} ... x_{i+N}$ with associated solution values $U[x_i], ...,$ $U[x_{i+N}]$, then the standard Newton divided difference form of the interpolating polynomial $U(x)$is given by

$$\begin{aligned} U(x) = U[x_i] + \pi_{1,i}(x)\, U[x_i, x_{i+1}] + \pi_{2,i}(x)\, U[x_i, x_{i+1}, x_{i+2}] \\ + \pi_{3,i}(x)\, U[x_i, x_{i+1}, x_{i+2}, x_{i+3}] + ... + \pi_{N,i}(x)\, U[x_i, ..., x_{i+N}], \end{aligned} \tag{6}$$

where

$$\begin{aligned} \pi_{1,i}(x) = (x - x_i), \quad \pi_{2,i}(x) = (x - x_i)(x - x_{i+1}), \\ \pi_{3,i}(x) = (x - x_i)(x - x_{i+1})(x - x_{i+2}), \ etc. \end{aligned} \tag{7}$$

In this case each additional term in the series makes use of the next mesh point and associated solution value to the right of the previous ones. An alternative polynomial could have been constructed by starting at the point x_j, $j > 0$ and then adding successive points to the left or right of x_j, [13]. As the divided difference, $U[x_i, x_{i+1}, ..., x_{i+k}]$, is invariant under permutations of the points $x_i, x_{i+1}, ..., x_{i+k}$, the convention adopted here is that the points will be ordered as an increasing sequence when the difference is evaluated. The denominator in equation (5) will also then be the width of the stencil of points used to evaluate the difference.

The idea behind ENO type interpolants is to vary the difference stencil to consistently pick the best polynomial. For example suppose that $i > 1$ then one valid quadratic polynomial for interpolation on the interval $[x_i, x_{i+1}]$ is given by the first three terms of the sum on the right side of equation (6) which uses the three data points $U(x_i), U(x_{i+1})$ and $U(x_{i+2})$. An alternative polynomial using the points $U(x_i), U(x_{i+1})$ and $U(x_{i-1})$ is given by

$$U(x) = U[x_i] + \pi_{1,i}(x)\, U[x_i, x_{i+1}] + \pi_{2,i}(x)\, U[x_{i-1}, x_i, x_{i+1}] \tag{8}$$

with the same values of the functions $\pi_{1,i}(x)$ and $\pi_{2,i}(x)$. The central idea in ENO methods [10] is to pick the polynomial with the smallest divided differences in order to potentially reduce oscillations. In the above case if

$$|U[x_{i-1}, x_i, x_{i+1}]| < |U[x_i, x_{i+1}, x_{i+2}]| \tag{9}$$

then the polynomial defined by equation (8) is used rather than the polynomial defined by the first three terms on the right side of equation (6). WENO methods use a combination of both of these polynomials, see [20], to achieve a higher degree of accuracy.

2.1 A Recursive Formulation of ENO Interpolants

A key step in constructing a provably data-bounded interpolant is to write the divided difference interpolation scheme in recursive form. This is important as it enables techniques used in in the finite volume solution of hyperbolic equations to generate data-bounded low-order polynomials to be extended to high order polynomials. In order to do this it is helpful to define the ratios of divided differences, for example, by

$$r^{[i,i+1]}_{[i-1,i]} = \frac{U[x_i, x_{i+1}]}{U[x_{i-1}, x_i]}, \tag{10}$$

with obvious extensions to higher differences and other indices. As an example, when a divided difference approximation incorporates a new point from the left x_{i-1}, is

$$U[x_{i-1}, x_i, \ldots, x_{i+k}, x_{i+k}] \;=\; \frac{\left(1 - r^{[x_{i-1},\ldots,x_{i+k-1}]}_{[x_i,\ldots,x_{i+k}]}\right)}{x_{i+k} - x_{i-1}} U[x_i, x_{i+1}, \ldots, x_{i+k}]. \tag{11}$$

An alternative divided difference computed from $U[x_i, x_{i+1}, \ldots, x_{i+k}]$ is

$$U[x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k+1}] \;=\; \frac{\left(r^{[x_{i+1},\ldots,x_{i+k+1}]}_{[x_i,\ldots,x_{i+k}]} - 1\right)}{x_{i+k+1} - x_i} U[x_i, x_{i+1}, \ldots, x_{i+k}]. \tag{12}$$

In this case the ENO scheme picks the next difference to be $U[x_{i-1}, x_i, \ldots, x_{i+k}, x_{i+k}]$ if

$$\frac{\left(|1 - r^{[x_{i-1},\ldots,x_{i+k-1}]}_{[x_i,\ldots,x_{i+k}]}|\right)}{|x_{i+k} - x_{i-1}|} < \frac{\left(|r^{[x_{i+1},\ldots,x_{i+k+1}]}_{[x_i,\ldots,x_{i+k}]} - 1|\right)}{|x_{i+k+1} - x_i|}, \tag{13}$$

or picks $U[x_i, x_{i+1}, x_{i+2}, \ldots, x_{i+k+1}]$ otherwise. In the approach of [1], providing that the values of $r^{[\ldots]}_{[\ldots]}$ satisfy the restriction

$$0 \le r^{[\ldots]}_{[\ldots]} \le 1. \tag{14}$$

then, if equation (13) holds we pick the next stencil point to be to the "left" i.e. x_{i-1} as in equation (11) and

$$1 \ge \lambda_{k+1} \;=\; \left(1 - r^{[x_{i-1},\ldots,x_{i+k-1}]}_{[x_i,\ldots,x_{i+k}]}\right) \ge 0 \tag{15}$$

Alternatively if equation (13) does not hold, then the next stencil point is picked to the "right", as in equation (12), x_{i+1} and

$$-1 \le \lambda_{k+1} \;=\; \left(r^{[x_{i+1},\ldots,x_{i+k+1}]}_{[x_i,\ldots,x_{i+k}]} - 1\right) \le 0. \tag{16}$$

In both cases it follows that

$$|\lambda_k| \le 1. \tag{17}$$

although the sign of λ_k is positive in one case and negative in the other. Berzins [1] proved for uniform meshes that this polynomial was data-bounded i.e.

$$Min(U(x_i), U(x_{i+1})) \le U(x) \le Max(U(x_i), U(x_{i+1})).$$

In section 2.3 this proof will be extended to non-uniform meshes.

2.2 Harten's Example

In order to illustrate the difference between the standard ENO approximation and the new approach consider the example of Harten [8]. The function being approximated is defined by:

$$\begin{aligned} U(x) &= (x+b)^3/6.0 + (x+b), x > -b \\ U(x) &= -(x+b), x \le -b \end{aligned} \tag{18}$$

where $b = 0.1e-3-0.5e-12$. the results in Figure 1 for polynomials of degree 4, 8 16 and 32 (with N the number of points defining the polynomial being one larger) show that the new scheme does not introduce oscillations but the original ENO scheme does oscillate for the different N evenly spaced mesh points. The L1 error in the standard ENO case for polynomials of degree 32 is $O(10^3)$.

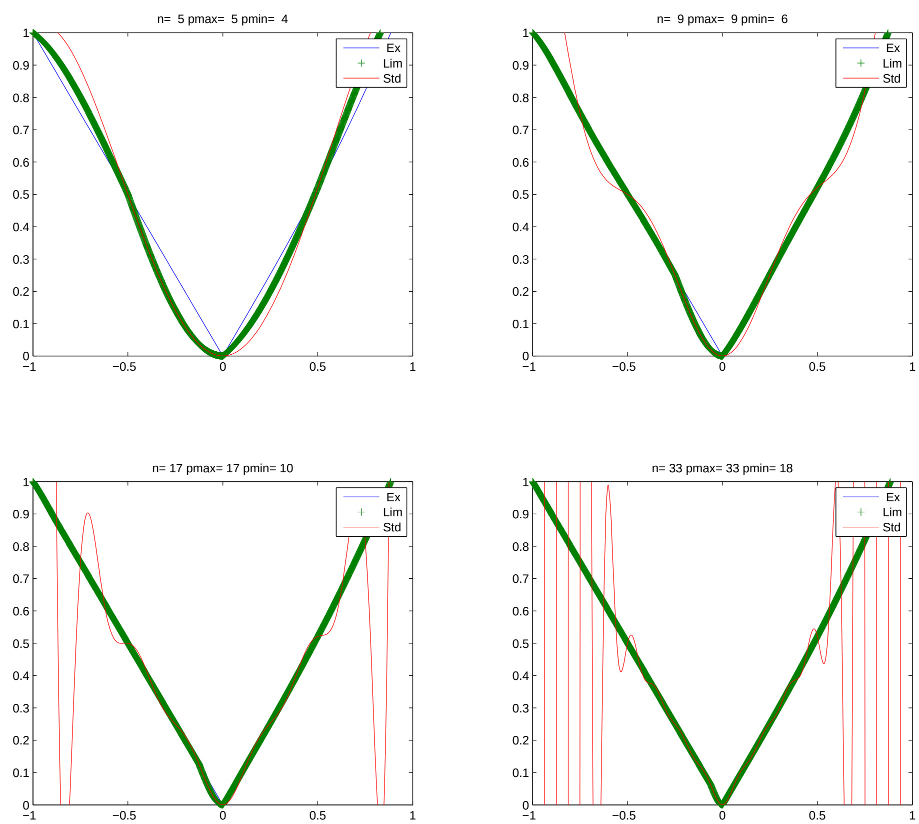

Figure 1: Harten's Example N=5,9,17,32 ENO vs Bounded ENO

2.3 A General Data-Bounded ENO Polynomial

In order to extend the proof of Berzins [1] to the more general case, it is helpful to define notation to describe the left and right edges of the stencil of points in use as this considerably simplifies the description of the ENO polynomial. Mesh points, x_i, are defined around a point x_0 by adding or subtracting multiples of an the mesh spacing h so that the mesh points chosen by the ENO approach at each stage are denoted by x_i^e as defined by

$$x_i^e = x_0 + e_i h,\ i \ge 1, h = (x_1 - x_0) \tag{19}$$

for some value e_i and where $e_1 = 1$. In the case when $e_i > 0$ then $e_i > 1$ otherwise e_i could be arbitrarily large or small. At the ith stage of the ENO process let the leftmost and right most parts of the stencil in use obviously depend on the choice made with regard to x_i^e and may be defined as

$$x_i^l = min(x_i^e, x_{i-1}^l), \;\; x_0^l = x_0, \tag{20}$$

$$x_i^r = max(x_i^e, x_{i-1}^r), \;\; x_0^r = x_0. \tag{21}$$

At the ith stage of the ENO process let the leftmost and right most parts of the stencil in use be defined by x_i^l and x_i^r. Further define a local co-ordinate in the interval $[x_0, x_1]$ by:

$$s = \frac{x - x_0}{x_1 - x_0}. \tag{22}$$

Using these definitions allows the limited form of the general ENO polynomial, as defined by equation (6) for example, to be written in the form:

$$U^l(x) = U[x_0] + [U(x_1) - U(x_0)]P_N(s) \tag{23}$$

where

$$P_N(s) = s(1 + \frac{(s-1)}{D_2}\lambda_2\ (1 + \frac{(s-e_2)}{D_3}\lambda_3 \times\ (1 + \frac{(s-e_3)}{D_4}\lambda_4 \times ....1 + \frac{(s-e_{N-1})}{D_N}(\lambda_N). \tag{24}$$

and where

$$D_i = (x_i^r - x_i^l)/(x_1 - x_0). \tag{25}$$

Equation (23) may be rewritten as

$$\begin{aligned} P_N(s) = (s + s\frac{(s-1)}{D_2}\overline{\lambda}_2\ + \frac{s(s-1)(s-e_2)}{D_2 D_3}\overline{\lambda}_3 + \\ \frac{s(s-1)(s-e_2)(s-e_3)}{D_2 D_3 D_4}\overline{\lambda}_4 + ... + \frac{s(s-1)(s-e_2)...(s-e_{N-1})}{D_2 D_3 ... D_N}\overline{\lambda}_N) \end{aligned} \tag{26}$$

where

$$\begin{aligned} \overline{\lambda}_2 = \lambda_2,\ \ \overline{\lambda}_3 = \lambda_2\ \ \lambda_3,\ \ \overline{\lambda}_4 = \lambda_2\ \ \lambda_3\ \ \lambda_4 \\ and\ \ \overline{\lambda}_N = \lambda_2\ \ \lambda_3 ... \lambda_{N-1}\ \ \lambda_N. \end{aligned} \tag{27}$$

and where

$$-1 \le \overline{\lambda}_i \le 1, i = 2, ..., N. \tag{28}$$

Theorem 1 The interpolating function constructed on an evenly spaced mesh using the ENO approach with limited ratios of divided differences is data-bounded for arbitrary order in that

$$Min(U(x_i), U(x_{i+1})) \le U^I(x) \le Max(U(x_i), U(x_{i+1})).$$

Proof. In proving this result is that we need to show that for $0 \le s \le 1$,

$$0 \le P_N(s) \le 1 \tag{29}$$

where $P_N(s)$ is defined by equation (24). for every possible consistent choice of D_i, e_j and $\overline{\lambda}_k$. The approach taken is to construct bounding polynomials such that

$$P_N^-(s) \le P_N(s) \le P_N^+(s) \tag{30}$$

Consider the polynomial defined by

$$S_N(s) = s\sum_{i=0}^{N-1}(1-s)^i \tag{31}$$

Its value is

$$S_N(s) = (1 - (1-s)^N). \tag{32}$$

We will show that one pair of choices is:

$$P_N^-(s) = s^N = 1 - S_N(1-s), \tag{33}$$

$$P_N^+(s) = S_N(s). \tag{34}$$

The proof starts with the quadratic case in which

$$P_2(s) = s(1 + \frac{(s-1)}{D_2}\lambda_2. \tag{35}$$

As $D_2 \geq 1$ and $|\lambda_2| \leq 1$ the largest possible polynomial is defined by $D_2 = 1$ and $\lambda_2 = -1$

$$P_2^+(s) = s + s(1-s). \tag{36}$$

As $\lambda_2 = -1$ this means that the cubic polynomial must have the form

$$P_3^+(s) = s(1 + (1-s) + \frac{(1-s)^2}{D_3}\lambda_3. \tag{37}$$

The closest to zero quadratic polynomial of the required form is then given by $D_2 = 1$ and $\lambda_2 = 1$ as

$$P_2^-(s) = s - s(1-s). \tag{38}$$

Continuing in the same way ensures that the largest polynomial of degree N has the form of equation (31). The largest possible polynomial of degree $N+1$ must have the form

$$P_{N+1}^+(s) = S_N(s) + \frac{s(1-s)^N}{D_{N+1}}\lambda_{N+1}. \tag{39}$$

Again, it is straightforward, to see that as $D_{N+1} \geq 1$ and $|\lambda_{N+1}| \leq 1$, that

$$P_{N+1}^+(s) = S_{N+1}(s). \tag{40}$$

Hence confirming the inductive step. Similarly for the lower bound a similar process leads to

$$P_{N+1}^-(s) = 1 - S_N(1-s) - (1-s)(s)^N, \tag{41}$$

as required. The bounding polynomial corresponds to a polynomial with data points at $s = 0$ and then multiple data points at $s = 1$. It is possible to get arbitrarily close to this polynomial with data points defined by $s = 0$ and $s = 1$ and then $s = 1 + i\varepsilon$. This polynomial is defined by

$$T_N(s) = s + s\sum_{i=1}^{N-1}\prod_{j=1}^{i}\frac{(1 - s - (j-1)\varepsilon)}{1 + j\varepsilon}. \tag{42}$$

In order to illustrate these results random polynomials of degree 23 were created to provide a sample of 100 polynomials in which the underlying mesh varies randomly by mesh ratios that change from one cell to the next by as much or as little as 10^5 and 10^{-5}. The left figure plots the polynomials while the righthand figue shows the distribution of the mesh ratios on a logarithnic scale, for each of the 100 cases. The bounding polynomials used in the proof are also shown. The polynomial $T_n(s)$ is evaluated with $\varepsilon = 0.1e-2$. The results show the data-bounded nature of the polynomial, even for extreme mesh ratios samd also the bounding case.

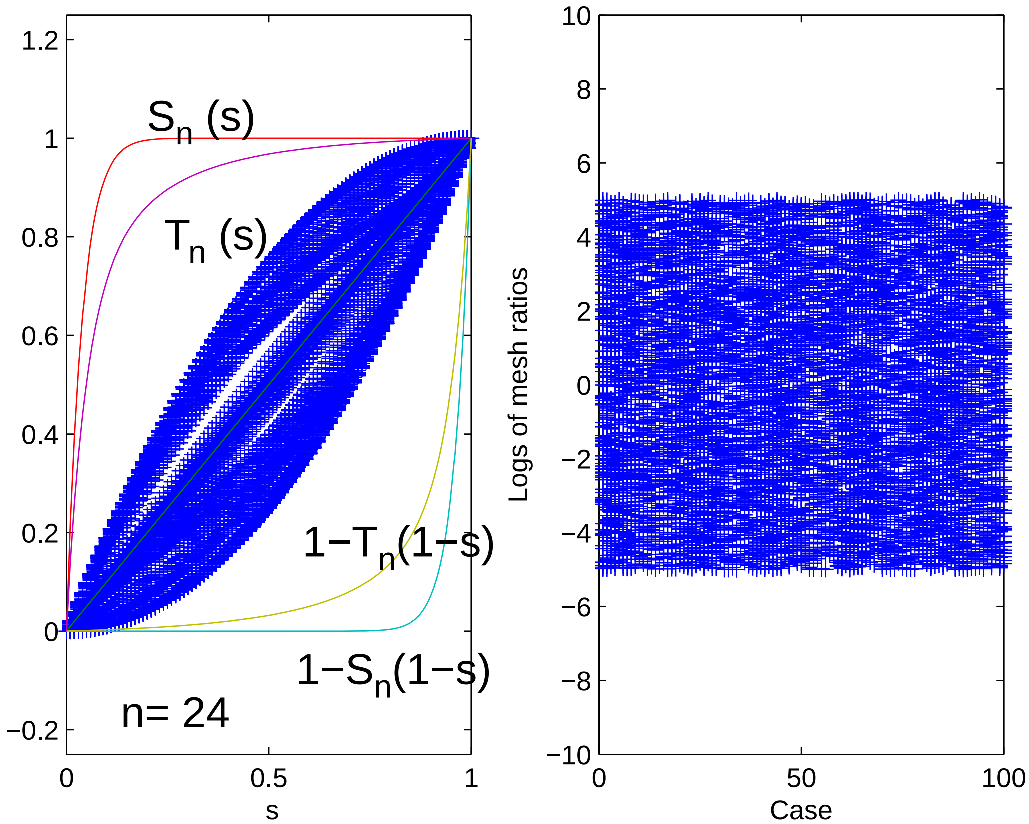

Figure 2: Random Polynomial coefficient results to illustrate Theorem 1.

2.4 Reintroducing Extrema.

It is well-known that a key feature of schemes for hyperbolic equations is that they must not clip local extrema, [2]. One possible problem with the proposed approach of bounding the polynomial by the values at either end of the interval is that if the true solution has an extremal value in between the data points then this value will be truncated. One solution to this is to detect possible extrema in an interval and switch off limiting. The proposed condition for deetecting possible extrema is given by requiring that the cells on either side of the ”flat” cell have opposite and significant slopes. In other words the following two conditions must hold for extrema to be assumed to exist:
(i) $U[x_{i+1},x_{i+2}]/U[x_{i-1},x_i] \leq 0$,
(ii) $U[x_{i+1},x_{i+2}]/U[x_i,x_{i+1}] \geq 1$ $U[x_i,x_{i+1}]/U[x_{i-1},x_i] \leq 1$.
In the case when a possible extremal value is detected then limiting is switched off in that interval and a standard ENO polynomial used. The effectiveness of this approach on Runge's function with NPTS data points spaced so as to exclude the extremal value is shown by the numerical results in Table 1. In Table 1, NP is the number of points used to define the polynomial, or the order plus one. In the case when extrema are introduced the only difference is that in the central section of the plots in Figure 3 the new method has exactly the same profile as the original ENO method.

2.5 Derivative Approximations in ENO Schemes

In order to use the above approximation results in the context of numerical schemes fro hyperbolic equations it is important to understand the behavior of the polynomial derivatives at the the spatial mesh points. This behavior is described by the following theorem.

Method	NPTS	L2 Error	L∞ Error	Max NP	Min NP	Avg NP
No New Extrema Allowed	6	3.4e-3	5.0e-1	4	3	3
	14	5.7e-4	1.3e-1	8	3	7
	30	8.6e-5	2.9e-2	17	3	15
	60	1.5e-5	7.1e-3	34	3	32
	120	2.6e-6	1.3e-4	57	3	53
New Extrema Allowed	6	2.9e-3	4.3e-1	6	3	4
	14	1.9e-4	4.3e-2	14	5	8
	30	2.3e-6	4.7e-4	30	15	16
	60	5.7e-9	2.3e-6	60	18	33
	120	5.1e-8	1.1e-8	120	38	54

Table 1: Approximation of Runge's Function With and Without Extrema Creation

Theorem 2: The interpolating function constructed on an evenly spaced mesh using the modified ENO algorithm is monotone in that

$$\frac{dU^I(x)}{dx} = (U(x_1) - U(x_0))\ f(x)$$

where $f(x) \geq 0$ for $x = x_0$ and $x = x_1]$.

Proof: The interpolating polynomial on an interval $[x_0, x_1]$ may be written as

$$U^I(x) = U[x_0] + \frac{[U(x_1) - U(x_0)]}{(x_1 - x_0)}(x - x_0)(1 + (x - x_1)P^*(x))$$

where $P^*(x)$ is defined by collecting together the terms in the polynomial expansion defined by equation (24) as

$$P^*(x) = \frac{\lambda_2}{D_2(x_1 - x_0)^2}\ (1 + \frac{(s - e_2)}{D_3}\lambda_3 \times\ (1 + \frac{(s - e_3)}{D_4}\lambda_4 \times ....1 + \frac{(s - e_{N-1})}{D_N}(\lambda_N).$$

with s defined as in equation (22). Differentiating this equation gives

$$\frac{dU^I(x)}{dx} = \frac{U(x_1) - U(x_0)}{(x_1 - x_0)}\left[(x - x_0)(x - x_1)\frac{dP^*(x)}{dx} + (1 + (2x - x_i - x_{i+1}))P^*(x))\right]$$

Evaluating this at the grid point $x = x_1$ gives

$$\frac{dU^I(x_1)}{dx} = \frac{(U(x_1) - U(x_0)}{(x_1 - x_0)}\,[1 + (x_1 - x_0)P^*(x_1)]$$

and at the grid point $x = x_0$ gives

$$\frac{dU^I(x_0)}{dx} = \frac{(U(x_1) - U(x_0))}{(x_1 - x_0)}\,[1 - (x_1 - x_0)P^*(x_0)]\,.$$

As the polynomial $U^I(x)$ is data-bounded on the interval it follows that the derivatives at the end points must have the same sign as the first divided difference of $U(x)$ and so that

$$[1 + (x_1 - x_0)P^*(x_1)] \geq 0$$

and that

$$[1 - (x_1 - x_0)P^*(x_0)] \geq 0.$$

Thus ensuring that

$$P^*(x_1) \geq \frac{-1}{(x_1 - x_0)} \tag{43}$$

and that

$$P^*(x_0) \geq \frac{1}{(x_1 - x_0)}. \tag{44}$$

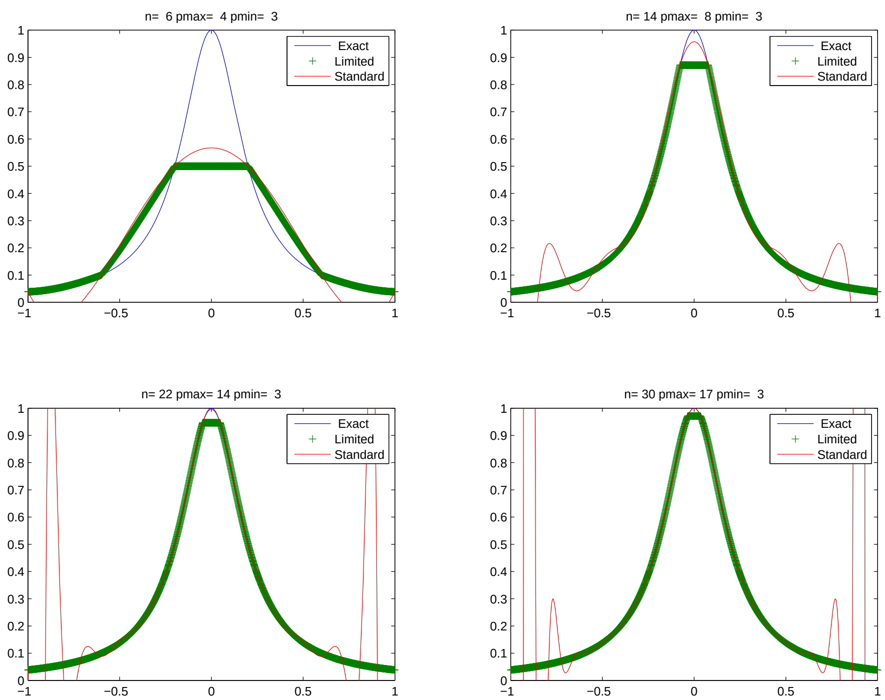

Figure 3: Runge's function approximated with standard and limited polynomials

2.6 Rounding Error Analysis

The rounding error analysis of Newton polynomials and the Horner's scheme often used to evaluate them is as old as modern numerical analysis; Higham [9], pp.109-115, gives an excellent survey of work going back to Wilkinson. Some of the more recent results show that severe difficulties may be encountered at very high orders. In the approach here a stencil of points is defined for each interval. Once the points are chosen the polynomial may be evaluated with any suitable method. An important part of this evaluation for the differential equations considered here is to evaluate the derivatives of the polynomial at the mesh points. In order to consider the rounding error in this the approach of [9] may be applied as the polynomial $P^*(x)$ is simply calculated in the same way as applying Horner's scheme to $P_N(x)$ and then truncating the summation two steps early and dividing by $(x_1 - x_0)^2$. As the summation takes place at the mesh points Higham's analysis is immediately applicable.

It is also worth noting that recent work on the compensated Horner scheme substantially improves the accuracy, [7]. One possible error with the approach described here is that rounding errors in the individual divided differences with introduce errors in parameters $\bar{\lambda}_j$, in equations (26-28), and hence possibly in the choice of stencil used. In the worst case using equation (24) directly to evaluate the polynomial will result in a bounded polynomial when this should not be the case, due to rounding error.

3 Positivity Preserving ENO Schemes

Once the polynomial approximation is defined as above it is straight forward to use the results of Theorem 2 to prove results about ENO and WENO schemes. These schemes integrate equation (1) over the interval $[x_{i-1}, x_i]$:

$$\frac{\partial \bar{u}_{i+1/2}}{\partial t} + a\frac{[u(x_i,t) - u(x_{i-1},t)]}{(x_i - x_{i-1})} = 0 \tag{45}$$

where

$$\bar{u}_{i+1/2}(t) = \frac{1}{(x_i - x_{i-1})} \int_{x_{i-1}}^{x_i} u(x,t)dx. \tag{46}$$

Defining the ENO reconstruction function w(x) by:

$$w_i(x,t) = \int_{\bar{x}_{i-1}}^{x} u(x^*,t)dx^*, x \in [x_{i-1}, x_i], \tag{47}$$

where $\bar{x}_{i-1}$ is an arbitrary lower limit, immediately provides the relationship

$$w_i(x_i,t) - w_i(x_{i-1},t) = \bar{u}_{i+1/2}(t)(x_i - x_{i-1}). \tag{48}$$

From differentiating equation (47) it follows that

$$\frac{dw_i}{dx}(x_i) - \frac{dw_i}{dx}(x_{i-1}) = u(x_i,t) - u(x_{i-1},t). \tag{49}$$

At the boundary $x = 0$ the appropriate solution value $U_0(t)$ is substituted for $u(x_{i-1},t)$. Using this relation in equation (45) and integrating in time using the forward Euler method gives.

$$\bar{u}_{i+1/2}(t_{n+1}) = \bar{u}_{i+1/2}(t_n) - \frac{a\delta t}{(x_i - x_{i-1})}\left[\frac{dw_i}{dx}(x_i,t_n) - \frac{dw_i}{dx}(x_{i-1},t_n)\right]. \tag{50}$$

In choosing the values of these derivatives it is necessary to take into account upwind directions, see [22]. The essence of the ENO algorithm is to take the following steps:
(i) On each interval create initial values of $\bar{u}_{i+1/2}(t)$ by using high-order quadrature based on the values $u(x,t)$.
(ii) Use equation (48) to create the first differences of the function $w_i(x,t)$.
(iii) Use these differences and subsequent differences to create a high order polynomial approximation on each interval to $w_i(x,t)$ denote this by $w_i^*(x,t)$.
(iv) Calculate $\frac{dw_i^*}{dx}(x_i)$ and $\frac{dw_i^*}{dx}(x_{i-1})$ using the algorithm described in Section 2.
(v) Advance the solution in time using equation (50) with a sufficiently small timestep.
From the analysis of Section 2 it follows that

$$\frac{dw_i^*}{dx}(x_i,t) = \frac{w(x_i,t) - w(x_{i-1},t)}{(x_i - x_{i-1})}(1 + h_i P_i^*(x_i)) + O(h^{k_i}) \tag{51}$$

where $P_i^*(x_i)$ is the polynomial $P^*(x)$evaluated on the interval $[x_i, x_{i+1}]$, and consequently that

$$\frac{dw_i^*}{dx}(x_i,t) = \bar{u}_{i+1/2}(t)(1 + h_i P_i^*(x_i)) + O(h^{k_i}). \tag{52}$$

In similar vein

$$\frac{dw_{i-1}^*}{dx}(x_{i-1}) = \frac{w(x_{i-1},t) - w(x_{i-2},t)}{(x_i - x_{i-1})}(1 + h_{i-1}P_{i-1}^*(x_{i-1})) + O(h^{k_{i-1}}) \tag{53}$$

and

$$\frac{dw_{i-1}^*}{dx}(x_{i-1}) = \bar{u}_{i-1/2}(t)(1 + h_{i-1}P_{i-1}^*(x_{i-1})) + O(h^{k_{i-1}}). \tag{54}$$

Hence equation (50) may be written as

$$\bar{u}_{i+1/2}(t_{n+1} = \bar{u}_{i+1/2}(t_n) - \frac{a\delta t}{(x_i - x_{i-1})}\left[\bar{u}_{i+1/2}(t)(1 + h_i P_i^*(x_i)) - \bar{u}_{i-1/2}(t)(1 + h_{i-1}P_{i-1}^*(x_{i-1}))\right].$$

Positivity of the $\bar{u}_{1+1/2}$ values then requires

$$0 \leq \frac{\delta t}{(x_i - x_{i-1})}(1 + h_i P_i^*(x_i)) \leq 1, \tag{55}$$

$$0 \leq \frac{\delta t}{(x_i - x_{i-1})}(1 + h_{i-1}P_{i-1}^*(x_{i-1})) \leq 1. \tag{56}$$

For a sufficiently small Courant number this follows from Theorem 2. Positivity of the solution values $U_i(t)$ requires a further step. From equation (50) it then follows that: the numerical solution values satisfy:

$$\frac{dw_i^*}{dx}(x_i) - \frac{dw_{i-1}^*}{dx}(x_{i-1}) = u(x_i,t) - u(x_{i-1},t), \tag{57}$$

and hence that

$$u(x_i,t) = \sum_{j=1}^{i} \left[\frac{dw_j^*}{dx}(x_j) - \frac{dw_{j-1}^*}{dx}(x_{j-1}) \right] + u_0(t). \tag{58}$$

From equation (50) it follows that

$$u(x_i,t) = \bar{u}_{i+1/2}(t)(1 + h_i P_i^*(x_i)), \tag{59}$$

and that positivity of the numerical averaged values $\bar{u}_{i+1/2}(t)$ implies positivity of the numerical solution values $U_i(t)$.

3.1 A Simple Alternative ENO Positivity Preservation Algorithm

The positivity condition based upon data-bounded polynomials is sufficient for positivity but not necessary in that positivity is still possible if the polynomials $P_i^*(x_i)$ and $P_{i-1}^*(x_{i-1})$ satisfy equations (55) and (56). Hence an alternative approach to seeking positivity is to simply require that the order of the ENO method be chosen so that

$$-1 \le h_i P_i^*(x_i)) \le \frac{1}{CFL} - 1, \tag{60}$$

$$-1 \le h_i P_i^*(x_{i-1})) \le \frac{1}{CFL} - 1. \tag{61}$$

When using this approach it is possible to get results that are as accurate as the original ENO approach by switching positivity preservation off when

$$|u(x_i,t) - u(x_{i-1},t)| \le TOL \tag{62}$$

This algorithm has performed well with $TOL = 0.0001$ in the experiments described below.

4 Investigation of the Order in ENO Methods

In order to illustrate and investigate the effect of using the positivity preserving methods described above three test problems will be used. In order to perform these experiments in a time-independent way, the spatial truncation error of the different approaches will be calculated and compared. In these experiments the original ENO method will be compared against the new approaches as indicated in by Section 3. The original ENO method gives almost identical results to the method described in Section 3.1.

4.1 ENO truncation Error

The classical spatial truncation error for ENO methods may be calculated from the exact solution $u(x,t)$ by first calculating $\bar{u}_{i+1/2}(t)$ and then forming $\frac{dw_i^*}{dx}(x_i,t)$ by using the polynomial approximation procedure to arrive at approximations $\frac{dw_i^*}{dx}(x_i,t)$. The truncation error is then denoted by $TE_{eno}(x,t)$, where

$$TE_{eno}(x,t) = \frac{u(x_i,t) - u(x_{i-1},t)}{x_i - x_{i-1}}. - \frac{1}{x_i - x_{i-1}}. \left[\frac{dw_i^*}{dx}(x_i) - \frac{dw_{i-1}^*}{dx}(x_{i-1}) \right]. \tag{63}$$

This truncation error provides a measure of how accurate an ENO method with these hig -order polynomial expressions can be.

4.2 Computational Experiments

The following three examples have been used in the past to demonstrate the properties of hyperbolic equations solves and are used here to illustrate the properties of the approaches discussed above. In each case the L1 error norm is used as approximated by a discrete sum over the mesh point values. In the tables below NPTS is the total number of mesh points used and NP is the number of points used to define a polynomial, in otherwords the polynomial order +1. The method denoted by BENO is the bounded polynomial approach defined by Section 3. The method denoted by LENO is the limited ENO approach defined by Section 3.1 above.

4.2.1 Problem 1

The first problem is the Gaussian example was used by Rider at al [12], to illustrate the advantage of using high-order methods for problems with smooth solutions. The solution is defined by

$$U(x,t) = 0.1 + e^{(-(10.0*xtemp-5)^2/0.718125)} \tag{64}$$

In this case it is not surprising that the best results have been obtained with polynomials of degree 12 or higher.

Problem 1		NPTS					
Method	NP	15	31	63	127	255	511
BENO	3	1.1e-1	5.5e-2	1.2e-2	1.4e-3	1.9e-4	2.6e-5
LENO	3	9.9e-2	5.2e-2	3.4e-3	1.8e-4	1.3e-5	8.0e-7
BENO	6	7.2e-2	2.2e-2	6.5e-4	1.1e-5	2.2e-7	3.5e-9
LENO	6	2.7e-1	2.1e-2	2.4e-4	3.8e-6	4.0e-8	1.9e-9
BENO	12	7.2e-2	2.6e-2	2.9e-5	2.0e-8	7.4e-12	2.2e-14
LENO	12	4.8e-1	6.9e-2	2.9e-5	8.0e-9	1.8e-12	2.4e-14
BENO	24	7.4e-2	3.9e-2	9.8e-4	9.2e-10	1.1e-12	4.1e-13
LENO	24	5.5e-1	2.3e-1	6.2e-3	2.4e-10	2.1e-12	9.4e-13

Table 2: Problem 1 Comparison of Truncation Errors for Gaussian

4.2.2 Problem 2

This is the problem involving the advection of $u(x,t) = sin^4(x)$, a problem considered by Shu [22] and others. In this case too the best results are obtained with polynomials of degree 12 or higher. In this case the B. ENO method is less accurate than the L.ENO method and the original ENO method as lower-order polynomials are used at extrema.

Problem 2		NPTS					
Method	NP	15	31	63	127	255	511
BENO	3	1.9e-1	2.6e-2	3.6e-3	4.6e-4	5.7e-5	7.2e-6
LENO	3	1.3e-1	9.0e-3	5.4e-4	3.1e-5	1.8e-6	1.1e-7
BENO	6	8.5e-2	2.2e-3	5.6e-5	1.7e-6	5.0e-8	1.5e-9
LENO	6	5.3e-2	1.3e-2	2.4e-5	7.8e-8	6.5e-10	5.1e-12
BENO	12	3.4e-1	1.5e-3	5.1e-5	1.5e-6	4.7e-8	1.4e-9
LENO	12	7.8e-2	5.6e-4	1.8e-5	6.6e-13	1.3e-14	1.2e-14
BENO	24	8.9e-1	1.6e-3	1.8e-5	1.5e-6	4.7e-8	1.5e-9
LENO	24	9.0e-2	5.6e-4	1.8e-5	6.3e-12	2.8e-11	3.6e-11

Table 3: Problem 2 Comparison of Truncation Errors for $Sin^4(x)$

4.2.3 Problem 3

The results of Rider [12] show that may not be substantialy better than low-order methods when solving problems with discontinuities. In order to investigate this the second problem has a solution which is both smooth and which has a steep profile as given by Hubbard [14], the 11th order polynomial:

$$p(z) = z^6 \left[-252z^5 + 1386z^4 - 3080z^3 + 3465z^2 - 1980z + 462\right] \tag{65}$$

where

$$z = (0.5 + t + ds * 0.5 - x)/ds; \tag{66}$$

and which has a front of width ds whose centre position is at $0.5 + t$. Three sets of numerical experiments were conducted with this problem. In case (a) the front width is $ds = 0.96$ and in case (b) the front width is $ds = 0.096$. and in case (c) the front width is $ds = 0.0096$. The three solutions are shown in Figure (4). In case (c) with a mesh of 511 points there is only one mesh point inside the steep gradient part of the solution. Table 4 shows that with large values of $ds > 0.1$, say, using high order leads to an improvement. In the case when $ds = 0.0096$ and there is only one mesh-point in the front then the numerical evidence shows that there is little point using more than quadratic approximations, $NP = 3$.

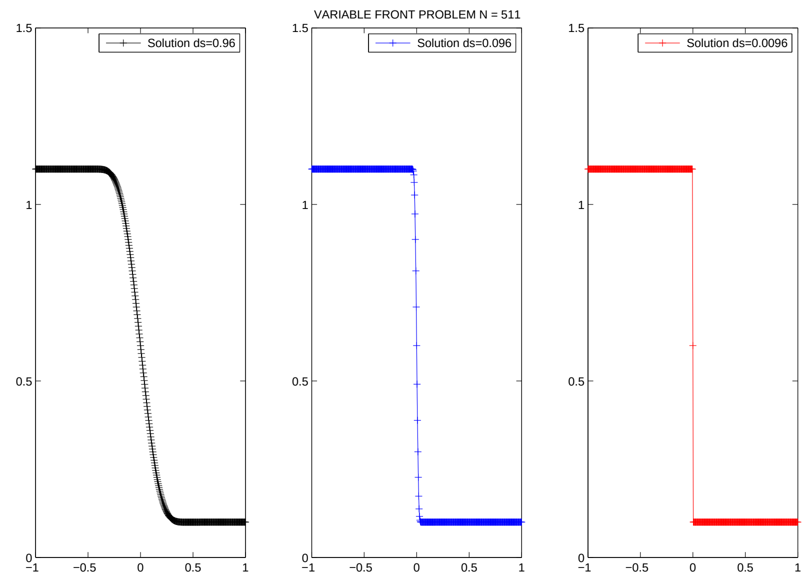

Figure 4: Problem 3:Steep Front Example Solutions, ds=0.96,ds=0.096,ds=0.0096

The conclusions from these experiments are that for smoopth solutions where there is enough mesh resolution there are advantages in using high order polynomials. The effective polynomial order does tend to be limited by the number of mesh points in a front, [1].

5 Summary

In this paper a novel approach to preserving positivity for variable-order ENO methods has been extended in a general way using the idea of bounded polynomial approximations. Positivity conditions have been proved and numerical experiments have shown that it is possible to use much higher order methods than is often done with ENO methods. Achieving the appropriate spatial order is somewhat more problematicali and on steep fronts it is important to have a mesh that ensures that multiple points are present in the front. One issue that still remains to be resolved is how to treat extrema in an accurate way without introducing new extrema elsewhere.

Problem 3	ds=0.96	NPTS					
Method	NP	15	31	63	127	255	511
B.ENO	3	1.8e-2	6.5e-3	1.3e-3	8.4e-5	3.3e-5	7.3e-6
L.ENO	3	1.1e-2	8.5e-4	6.5e-5	4.1e-6	2.5e-7	1.6e-8
B.ENO	6	5.3e-3	1.7e-4	5.3e-6	7.9e-8	1.8e-9	4.0e-11
L.ENO	6	7.8e-3	8.8e-5	1.6e-6	2.0e-8	2.0e-10	8.7e-12
B.ENO	12	5.9e-2	8.9e-5	2.8e-7	2.7e-9	2.3e-11	7.5e-12
L.ENO	12	7.1e-2	1.5e-4	2.5e-7	1.9e-9	2.0e-11	9.1e-12
B.ENO	24	5.9e-2	1.8e-2	1.6e-4	7.2e-10	8.2e-12	5.8e-12
L.ENO	24	2.6e-1	1.1e-1	2.3e-4	6.8e-10	8.0e-12	5.7e-12
Problem 3	ds=0.096						
B.ENO	3	1.3e-1	5.5e-2	1.7e-2	2.6e-3	4.6e-4	7.1e-5
L.ENO	3	1.3e-1	4.7e-2	1.2e-2	1.3e-3	1.7e-4	1.4e-5
B.ENO	6	1.2e-1	4.2e-2	7.7e-3	7.2e-4	2.5e-5	1.1e-6
L.ENO	6	1.2e-1	4.0e-2	7.7e-3	5.0e-4	1.9e-5	4.6e-7
B.ENO	12	5.7e-1	3.4e-2	6.2e-3	2.6e-4	5.5e-6	8.3e-8
L.ENO	12	1.4e-0	3.9e-2	6.1e-3	2.2e-4	5.4e-6	6.2e-8
B.ENO	24	5.7e-1	2.5e-1	6.1e-3	1.7e-4	2.8e-6	3.0e-8
L.ENO	24	2.0e-0	1.2e-0	5.2e-3	1.6e-4	2.7e-6	2.9e-8
Problem 3	ds=0.0096						
BOTH	3	1.4e-1	6.9e-2	3.2e-2	1.4e-2	6.3e-3	2.3e-3
BOTH	6	1.4e-1	6.4e-2	3.0e-2	1.4e-2	5.7e-3	1.4e-3
BOTH	12	6.9e-1	6.3e-2	2.9e-2	1.3e-2	4.7e-3	1.1e-3
BOTH	24	6.9e-2	5.4e-2	2.8e-2	1.2e-20	4.4e-32	1.3e-3

Table 4: Problem 3 Comparison of Truncation Errors for Steep Front.

References

[1] Berzins M. Adaptive Equally Spaced Polynomial Interpolation *Siam Review* **47**:604-627, December 2007.

[2] Balsara D.S. and Shiu C.W. Monotonicity Preserving Weighted Essentially Non-Oscillatory Schemes with Increasingly High Order of Accuracy. *Journal of Computational Physics* **160**:405-452 (2000).

[3] Borisov V.S. and Sorek S. On Monotonicity of Difference Schemes for Computational Physics. *SIAM J. SCI. Computing* **25**:1557-1584, 2004.

[4] Cockburn B, and Shu C-W. Nonlinearly Stable Compact Schemes for Shock Calculations. *SIAM J. Numer. Anal.* 1994; **31**:607–627.

[5] Cockburn B, Karniadakis GE, Shu C-W. (eds). *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations. Lecture Notes in Mathematics 1697* Springer Berlin Heidelberg, 2000; pp 325–418.

[6] Godunov, S. K. A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations, *Math. Sbornik*, **47**, 271-306 (1959), translated US Joint Publ. Res. Service, JPRS 7226, 1969.

[7] Graillat S. Langlois P. Louvet N. Improving the compensated Horner scheme with a fused multiply and add Proceedings of the 2006 ACM symposium on Applied Computing Dijon, France, 1323-1327, 2006 ISBN:1-59593-108-2, ACM New York, NY, USA.

[8] Harten A., Osher S., Engquist B. and Chakravarthy S.K. Some Results on Uniformly High-Order Accurate Non-oscillatory Schemes *Applied Numerical Mathematics* 1986; **2**:347–377.

[9] Higham N.J. Accuracy and Stability of Numerical Algorithms. Siam, Philadelphia 1996.

[10] Harten A. and Osher S.. Uniformly High-Order Accurate Non-oscillatory Schemes I. *SIAM J. Numer. Anal* 1987; **24**:279–309.

[11] Harten A. Engquist B. Osher S. and Chakravarthy S.. Uniformly High-Order Accurate Non-oscillatory Schemes III. *J.Computational Physics* 1987; **71**:231–303.

[12] Greenough J.A. and Rider W.J. A quatitative comparison of numerical methods for the compressible Euler equations: fifth order WENO and piecewise-linear Godunov *J.Computational Physics* 2004; **196**:259–281.

[13] Hildebrand F.B. *Introduction to Numerical Analysisa* McGraw-Hill Book Company Inc 1956;

[14] Hubbard, M E; Berzins, M. *A positivity preserving finite element method for hyperbolic partial differential equations.* in: Armfield S, Morgan P and Srinivas K (editors) Computational Fluid Dynamics 2002, pp. 205-210 Springer-Verlag. 2003.

[15] Laney C. *Computational Gasdynamics*. Cambridge University Press.

[16] Liu X-D, Osher S. and Chan T. Weighted Essentially Non-oscillatory Schemes. *J.Computational Physics* 1994; **71**:231–303.

[17] Serna S. and Marquina A. Power ENO Methods: a fifth order accurate Weighted Power ENO Method. *J.Computational Physics* 2004; **194**:632–658.

[18] Shu C-W Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schems for Hyperbolic Consewrvbation Laws *NASA CR-97-206253, ICASE Report No 97-65* 1997;

[19] C.-W. Shu, TVD properties of a class of modied ENO schemes for scalar conservation laws, IMA Preprint Series 308 (1987), University of Minnesota.

[20] Shu C-W . High Order WENO Schemes. *SIAM Review* March 2009; **51**:82–126.

[21] Shu C-W and Osher S. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J.Computational Physics* 1988; **77**:439–471.

[22] C.-W. Shu, Numerical experiments on the accuracy of ENO and modied ENO schemes, *Journal of Scientic Computing*, v5 (1990), pp.127-149.

[23] Suresh A. and Huynh H.T. Accurate Monotonicity Preserving Scheme with Runge Kutta Timestepping. *J.Computational Physics* 1997; **136**:83- 439–471.