Numerical Algorithms manuscript No.
(will be inserted by the editor)

Nonlinear Data-Bounded Polynomial Approximations and their Applications in ENO Methods.

Martin Berzins

Received: date / Accepted: date

Abstract A class of high-order data-bounded polynomials on general meshes are derived and analyzed in the context of numerical solutions of hyperbolic equations. Such polynomials make it possible to circumvent the problem of Runge-type oscillations by adaptively varying the stencil and order used, but at the cost of only enforcing C^0 solution continuity at data points. It is shown that the use of these polynomials, based on extending the work of [1] to nonuniform meshes, provides a way to develop positivity preserving polynomial approximations of potentially high order for hyperbolic equations. The central idea is to use ENO (Essentially Non Oscillatory) type approximations but to enforce additional restrictions on how the polynomial order is increased. The question of how high a polynomial order should be used will be considered, with respect to typical numerical examples. The results show that this approach is successful but that it is necessary to provide sufficient resolution inside a front if high-order methods of this type are to be used, thus emphasizing the need to consider nonuniform meshes.
Mathematics subject classification 35L03 65M08 65D05.

Keywords Data Bounded Polynomials · Essentially Non-Oscillatory Methods

Mathematics Subject Classification (2000) MSC 35L03 · MSC 65M08 · 65D05

1 Introduction

This paper considers positivity preserving methods for hyperbolic equations, by combining two distinct approaches. The first approach is the substantial and influential body of work on ENO and WENO methods [12]. The second approach is recent work on data-bounded polynomial interpolants, [1]. 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

M. Berzins
SCI Institute, University of Utah, Salt lake City, Utah, USA
Tel.: 001-801-585-1545
E-mail: mb@sci.utah.edu

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 $[A, B]$.
In spite of the influential and substantial body of work on ENO and WENO methods See [5, 12], there are still a few unresolved 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 [11]. A slightly different approach so as to keep the ENO stencil closer to a linearly stable stencil is also described by Shu [13]. Balsara and Shu [3] construct specific high-order schemes. Shu points out in a recent survey of WENO methods it is difficult to generalize analysis of some of the methods beyond third order, [12]. Recently Zhang and Shu [15] have extended TVD WENO schemes to sixth order [15], by using a novel approach.
The approach taken here 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 divided differences used in the schemes. The overall intention is to derive conditions under which ENO and WENO methods are positive with regard to the standard definition used here for a positivity preserving scheme for the advection equation. This definition requires (see [4]) that the numerical solution at time t_{n+1} be written in terms of the numerical solution at time t_n in the form

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

The constant C should ideally be one, [4]. The key observation with regard to preserving positivity is due to Godunov see [4] 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 [4]. In obtaining such results for the methods considered here the first step is to prove that the data-bounded polynomial approximation on uniform meshes 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-preserving schemes of potentially high order in space in Section 3. Numerical results on three test problems in Section 4 show that it is possible to use polynomial of a higher order 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 [5,12], 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

$$U[x_i, x_{i+1}] = \frac{U[x_{i+1}] - U[x_i]}{x_{i+1} - x_i}, \tag{3}$$

and where subsequent differences are defined recursively by

$$U[x_i, x_{i+1}, ..., x_{i+k}] = \frac{U[x_{i+1}, x_{i+2} ..., x_{i+k}] - U[x_i, x_{i+1}, ..., x_{i+k-1}]}{(x_{i+k} - x_i)}. \tag{4}$$

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

$$\begin{aligned} U(x) = {} & U[x_i] + \pi_{1,i}(x)\, U[x_i, x_{i+1}] + \pi_{2,i}(x)\, U[x_i, x_{i+1}, x_{i+2}] \\ & + \pi_{3,i}(x)\, U[x_i, x_{i+1}, x_{i+2}, x_{i+3}] + ... + \pi_{N,i}(x)\, U[x_i, ..., x_{i+N}], \end{aligned} \tag{5}$$

where

$$\begin{aligned} \pi_{1,i}(x) = (x - x_i),\ \ \pi_{2,i}(x) = (x - x_i)(x - x_{i+1}), \\ \pi_{3,i}(x) = (x - x_i)(x - x_{i+1})(x - x_{i+2}),\ etc. \end{aligned} \tag{6}$$

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, [9]. 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 methods is to vary the difference stencil to consistently pick the best polynomial for each interval, rather than to use just one polynomial over the spatial range. 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 (5) 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

$$U(x) = U[x_i] + \pi_{1,i}(x)\, U[x_i, x_{i+1}] + \pi_{2,i}(x)\, U[x_{i-1}, x_i, x_{i+1}] \tag{7}$$

with the same values of the functions $\pi_{1,i}(x)$ and $\pi_{2,i}(x)$. ENO methods [5] pick the polynomial with the smallest divided differences in order to potentially reduce oscillations. In the above case if

$$|U[x_{i-1}, x_i, x_{i+1}]| < |U[x_i, x_{i+1}, x_{i+2}]| \tag{8}$$

then the polynomial defined by equation (7) is used rather than the polynomial defined by the first three terms on the right side of equation (5). WENO methods use a combination of both of these polynomials, see [12], 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

$$r_{[i-1,i]}^{[i,i+1]} = \frac{U[x_i, x_{i+1}]}{U[x_{i-1}, x_i]}, \tag{9}$$

with obvious extensions to higher differences and other indices. Such ratios are used as part of many very widely used positivity methods for solving compressible flow problems, see [14]. The main idea here is to use ratios of divided differences in constructing polynomials that may form part of positivity-preserving discretization methods of possibly arbitrarily high order. As an example, when the next divided difference approximation to be computed incorporates a new point from the left x_{i-1}, it may be written in the form

$$U[x_{i-1},x_i,...,x_{i+k},x_{i+k}] \;=\; \frac{\left(1-r^{[x_{i-1},...,x_{i+k-1}]}_{[x_i,...,x_{i+k}]}\right)}{x_{i+k}-x_{i-1}}U[x_i,x_{i+1},...,x_{i+k}]. \qquad (10)$$

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

$$U[x_i,x_{i+1},x_{i+2},...,x_{i+k+1}] \;=\; \frac{\left(r^{[x_{i+1},...,x_{i+k+1}]}_{[x_i,...,x_{i+k}]}-1\right)}{x_{i+k+1}-x_i}U[x_i,x_{i+1},...,x_{i+k}]. \qquad (11)$$

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

$$\frac{\left(|1-r^{[x_{i-1},...,x_{i+k-1}]}_{[x_i,...,x_{i+k}]}|\right)}{|x_{i+k}-x_{i-1}|} < \frac{\left(|r^{[x_{i+1},...,x_{i+k+1}]}_{[x_i,...,x_{i+k}]}-1|\right)}{|x_{i+k+1}-x_i|}, \qquad (12)$$

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

$$0 \le r^{[...]}_{[...]} \le 1, \qquad (13)$$

then, if equation (12) holds we pick the next stencil point to be to the ”left” i.e. x_{i-1} as in equation (10) and define λ_{k+1} (a term used in the next sub-section) by

$$1 \ge \lambda_{k+1} \;=\; \left(1-r^{[x_{i-1},...,x_{i+k-1}]}_{[x_i,...,x_{i+k}]}\right) \ge 0 \qquad (14)$$

Alternatively if equation (12) does not hold, then the next stencil point is picked to the ”right”, as in equation (11), x_{i+1} and define λ_{k+1} by

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

It is worth remarking that the restriction defined by equation (13) is an extension to higher order differences of the well-known minmod limiter for hyperbolic equations, see [14]. In the case when both components of $r^{[...]}_{[...]}$ are zero the value is set to zero as a safety strategy. While the restriction on $r^{[...]}_{[...]}$ is a potentially severe one it does not seem easy to circumvent it for arbitrary degree polynomials. Berzins [1] proved in the case of uniform meshes that the polynomial defined by the approach of equations (5,...,15) is 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 2.2, Theorem 1, a much simplified proof will be given that extend the approach to non-uniform meshes. It is also worth remarking that there are potentially many other approaches to limiting polynomials, but this approach is, to the best of the authors knowledge, perhaps the only one that works for polynomials of arbitrarily high degree.

2.2 A General Data-Bounded ENO Polynomial

In order to extend the proof of Berzins [1] to non-uniform meshes, it is helpful to define notation to describe the left and right edges of the stencil. 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

$$x_i^e = x_0 + e_i h, \; i \geq 1, where\ h = (x_1 - x_0), \tag{16}$$

for some value e_i and where $e_1 = 1$. In the case when $e_i > 0$ then $e_i > 1$. At the ith stage of the ENO process let the leftmost and right most parts of the stencil in use may be defined as x_i^l and x_i^l, where

$$x_i^l = min(x_i^e, x_{i-1}^l), \;\; x_0^l = x_0, \tag{17}$$

$$x_i^r = max(x_i^e, x_{i-1}^r), \;\; x_0^r = x_0. \tag{18}$$

Further, define a local co-ordinate, denoted by s, in the interval $[x_0, x_1]$ by:

$$s = \frac{x - x_0}{x_1 - x_0}. \tag{19}$$

These definitions allow the limited ENO polynomial, [1], to be written as:

$$U^I(x) = U[x_0] + [U(x_1) - U(x_0)]P_N(s) \tag{20}$$

where $P_N(s)$ is the polynomial defined by:

$$P_N(s) = s(1 + \frac{(s-1)}{D_2}\lambda_2 \; (1 + \frac{(s-e_2)}{D_3}\lambda_3 \times \; (1 + \frac{(s-e_3)}{D_4}\lambda_4 \times ...(1 + \frac{(s-e_{N-1})}{D_N}\lambda_N)...) \tag{21}$$

and where

$$D_i = (x_i^r - x_i^l)/(x_1 - x_0). \tag{22}$$

Equation (21) may be rewritten as

$$P_N(s) = (s + s\frac{(s-1)}{D_2}\overline{\lambda}_2 \; + \frac{s(s-1)(s-e_2)}{D_2 D_3}\overline{\lambda}_3 + \frac{s(s-1)(s-e_2)(s-e_3)}{D_2 D_3 D_4}\overline{\lambda}_4 + ... + \frac{s(s-1)(s-e_2)...(s-e_{N-1})}{D_2 D_3 ... D_N}\overline{\lambda}_N) \tag{23}$$

where $\overline{\lambda}_j = \prod_{k=2}^{j} \lambda_k$, and where $-1 \leq \overline{\lambda}_j \leq 1$. from the restrictions in equations (14) (15). It is perhaps worth remarking that the original ENO polynomial has the same form as that in equation (23), but without any restriction on the values of $\overline{\lambda}_j$. The differences between the two polynomials and hence the error introduced by introducing the data bounded approach are thus straightforward to describe, see [1].

Theorem 1 The interpolating function $U^I(x$ constructed using the ENO approach with limited ratios of divided differences is data-bounded on a nonuniform mesh in that

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

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

$$0 \leq P_N(s) \leq 1 \tag{24}$$

where $P_N(s)$ is defined by equation (23), for every possible consistent choice of D_i, e_j and λ_k. The approach taken is to construct two bounding polynomials such that

$$P_N^-(s) \leq P_N(s) \leq P_N^+(s). \tag{25}$$

Consider the polynomial defined by

$$S_N(s) = s \sum_{i=0}^{N-1} (1-s)^i. \tag{26}$$

The value of this sum is given by:

$$S_N(s) = (1-(1-s)^N). \tag{27}$$

Two polynomials that are upper and lower bounds on the interpolating polynomial are denoted by P_N^+ and P_N^- respectively and are given by:

$$P_N^-(s) = s^N = 1 - S_N(1-s), \tag{28}$$

$$P_N^+(s) = S_N(s). \tag{29}$$

The proof starts by considering supposing that the assertion is true for N. The largest possible polynomial of degree $N+1$ must have the form

$$P_{N+1}^+(s) = S_N(s) + s(1-s)^{N-1}(-1+\frac{1}{e_N}) + \frac{s(1-s)^{N-1}(s-e_N)}{e_N D_{N+1}} \lambda_{N+1}, \tag{30}$$

where

$$D_{N+1} \geq e_N \geq 1, \tag{31}$$

The largest possible value of λ_{N+1} and one that also makes the last term positive is given by

$$\lambda_{N+1} = -1. \tag{32}$$

Similarly the last term is maximized by a value of $D_{N+1} = 1$ given in equation (31). For the assertion to be true we require that

$$P_{N+1}^+(s) \leq S_{N+1}(s). \tag{33}$$

Comparing the differences between the polynomials defined by equations (23) and (26), it follows that this is true if

$$\left[(-1+\frac{2}{e_N}) - \frac{s-e_N}{e_N^2}\right] \leq 1-s,$$

and hence that

$$-1+\frac{2}{e_N} - \frac{s}{e_N^2} \leq 1-s, \tag{34}$$

as the value of the left side decreases as the value of e_N increases above one, it follows immediately that the largest polynomial for $N+1$ is given by

$$e_N = 1. \tag{35}$$

Hence if the assertion is true for N it is also true for $N+1$ The proof also applies to the base quadratic case, thus closing the induction. Similarly for the lower bound, a similar process leads to

$$P^{-}_{N+1}(s) = 1 - S_N(1-s) - (1-s)(s)^N, \tag{36}$$

as required.
Remark 1 The bounding polynomial $S_N(s)$ 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 $s=0$, $s=1$ and then $s=1+i\varepsilon$ in the polynomial as defined by

$$T_N(s) = s + s\sum_{i=1}^{N-1}\prod_{j=1}^{i}\frac{(1-s-(j-1)\varepsilon)}{1+j\varepsilon}, \varepsilon > 0. \tag{37}$$

Remark 2 With this method any monotone polynomial whose values in the interval lie outside of the region bounded by $1-S_N(1-s)$ and $S_N(s)$, for example s^{N+1}, will be approximated by a polynomial whose values are contained in this region. The error in this approximation is given by [1]. This observation may make it possible to construct limiting algorithms whose bounding curves lie closer to the edges of the box.

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}. Figure 1 plots the polynomials and shows the distribution of the mesh ratios on a logarithmic 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 $\varepsilon = 0.001$. The results show the data-bounded nature of the polynomial, even for extreme mesh ratios.

2.3 Reintroducing Extrema.

It is well-known that a key feature of schemes for hyperbolic equations is that they must not clip local extrema, [3]. 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 in that interval. The proposed condition for detecting 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 $1/(1+25x^2)$, with NPTS evenly data points spaced so as to exclude the extremal value at $x=0$, 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. The table (not surprisingly) shows that carefully allowing extrema gives much greater accuracy than truncating extrema.

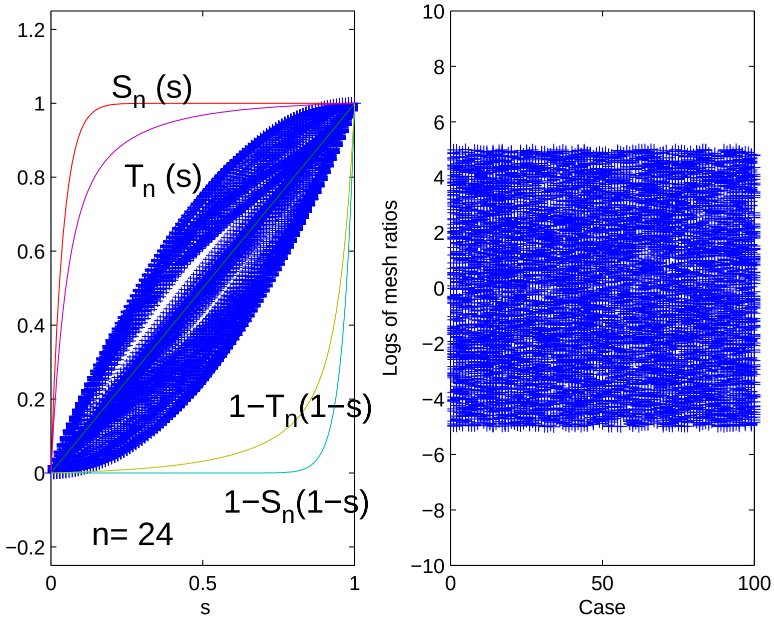

Fig. 1 Random Polynomial coefficient results to illustrate Theorem 1. The left figure shows the values of the polynomials. The right figure shows the mesh point spacing ratios for the 100 randomly chosen cases.

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

2.4 Derivative Approximations in ENO Schemes

In order to use the above approximation results in the context of numerical schemes for 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.

Theorem 2: The interpolating function, $U^I(x)$, constructed using the modified ENO algorithm satisfies the equation:

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

where $f(x) \geq 0$ for $x = x_0$ and $x = x_1$.
Proof: From equation (20), the interpolating polynomial on an interval $[x_0, x_1]$ may be written as

$$U^I(x) = U[x_0] + \frac{[U(x_1) - U(x_0)]}{(x_1 - x_0)}(x - x_0)(1 + (x - x_1)P^*(s)) \tag{38}$$

where $P^*(s)$ is defined by rearranging the terms in the polynomial expansion defined by equation (23) as

$$P^*(s) = \frac{\lambda_2}{D_2(x_1 - x_0)}(1 + \frac{(s - e_2)}{D_3}\lambda_3 \times (1 + \frac{(s - e_3)}{D_4}\lambda_4 \times ...1 + \frac{(s - e_{N-1})}{D_N}(\lambda_N), \tag{39}$$

with s defined as in equation (19). Differentiating equation (38) gives

$$\frac{dU^I(x)}{dx} = \frac{U(x_1) - U(x_0)}{(x_1 - x_0)} \left[(x - x_0)(x - x_1)\frac{dP^*(s)}{dx} + (1 + (2x - x_0 - x_1))P^*(s))\right]. \tag{40}$$

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

$$\frac{dU^I(x_1)}{dx} = \frac{(U(x_1) - U(x_0)}{(x_1 - x_0)}[1 + (x_1 - x_0)P^*(1)] \tag{41}$$

and again evaluating this expression at the grid point $x = x_0$ gives

$$\frac{dU^I(x_0)}{dx} = \frac{(U(x_1) - U(x_0))}{(x_1 - x_0)}[1 - (x_1 - x_0)P^*(0)]. \tag{42}$$

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 the quantities $[1 + (x_1 - x_0)P^*(1)]$ and $[1 - (x_1 - x_0)P^*(0)]$ must be positive. These quantities may be zero in the cases when

$$P^*(s) = \frac{\pm 1}{(x_1 - x_0)}.$$

More precise upper bounds for these expressions may be obtained by substituting for $P_N(s)$ using equation (38) into equation (25) and using equations (28) and (29) to get

$$s^N \leq s + s(s - 1)(x_1 - x_0)P^*(s) \leq s\sum_{i=0}^{N-1}(1 - s)^i. \tag{43}$$

Subtracting s from all the terms and dividing by $s(1 - s)$ gives.

$$\frac{(s^{N-1} - 1)}{(s - 1)} \leq -(x_1 - x_0)P^*(s) \leq \sum_{i=0}^{N-2}(1 - s)^i. \tag{44}$$

Consequently at $s = 1$, after using L'Hôpital's rule and rearranging gives

$$0 \leq 1 + (x_1 - x_0)P^*(1) \leq N, \tag{45}$$

and at $s = 0$

$$0 \leq 1 - (x_1 - x_0)P^*(0) \leq N. \tag{46}$$

Thus giving bounds on the right sides of equations (41) and (42).

2.5 Rounding Error Analysis

The rounding error analysis of Newton polynomials and Horner's scheme is as old as modern numerical analysis. Higham [7], pp.109-115, gives an excellent survey of work going back to Wilkinson. Some of the more recent results show that severe rounding error 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 [7] may be applied as the polynomial $P^*(x)$, as defined by (38), 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, [6]. One possible problem 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, with rounding error, using equation (24) directly to evaluate the polynomial may result in a bounded polynomial when this should not be the case.

3 Positivity Preserving ENO Schemes

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

$$\frac{\partial \bar{u}_{i+1/2}}{\partial t} + a\frac{[u(x_i,t) - u(x_{i-1},t)]}{(x_i - x_{i-1})} = 0 \tag{47}$$

where

$$\bar{u}_{i+1/2}(t) = \frac{1}{(x_i - x_{i-1})}\int_{x_{i-1}}^{x_i} u(x,t)dx. \tag{48}$$

Defining the ENO reconstruction function $w_i(x,t)$ by:

$$w_i(x,t) = \int_{\bar{x}_{i-1}}^{x} u(x^*,t)dx^*, x \in [x_{i-1}, x_i], \tag{49}$$

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

$$w_i(x_i,t) - w_i(x_{i-1},t) = \bar{u}_{i+1/2}(t)h_i. \tag{50}$$

From differentiating equation (49) it follows that

$$\frac{dw_i}{dx}(x_i,t) - \frac{dw_i}{dx}(x_{i-1},t) = u(x_i,t) - u(x_{i-1},t). \tag{51}$$

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 (47) and integrating in time using the forward Euler method gives.

$$\bar{u}_{i+1/2}(t_{n+1}) = \bar{u}_{i+1/2}(t_n) - \frac{a\delta t}{h_i}\left[\frac{dw_i}{dx}(x_i,t_n) - \frac{dw_i}{dx}(x_{i-1},t_n)\right]. \tag{52}$$

In calculating the values of these derivatives of $\frac{dw_i}{dx}(x,t)$, it is necessary to take into account upwind directions, see [13]. For a more general p.d.e. we would have to evaluate flux function values using Riemann solvers etc. The essence of the ENO algorithm for the advection equation is to take the following steps:
(i) On each interval create initial values of $\bar{u}_{i+1/2}(t)$ by using exact or high-order quadrature based on the values $u(x,t)$.
(ii) Use equation (50) 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 $w_i(x,t)$; we denote this polynomial by $w_i^*(x,t)$.
(iv) Calculate $\frac{dw_i^*}{dx}(x_i,t)$ and $\frac{dw_i^*}{dx}(x_{i-1},t)$ using the algorithm described in Section 2.
(v) Advance the solution in time using equation (52) with a sufficiently small timestep, δt.
From the analysis of Section 2 and using the data bounded polynomial approximation of that section, it follows that

$$\frac{dw_i^*}{dx}(x_i,t) = \frac{w(x_i,t)-w(x_{i-1},t)}{h_i}(1+h_iP_i^\diamond(x_i,t)) \tag{53}$$

where $P_i^\diamond(x,t)$ is the polynomial $P^*(s)$ evaluated on the interval $[x_i,x_{i+1}]$ at time t, and consequently, from equation (50) and using an upwind approach on $[x_{i-1},x_i]$ that

$$\frac{dw_i^*}{dx}(x_i,t) = \bar{u}_{i+1/2}(t)(1+h_iP_i^\diamond(x_i,t)). \tag{54}$$

In similar vein, using an upwind approach on $[x_{i-2},x_{i-1}]$, it follows that

$$\frac{dw_{i-1}^*}{dx}(x_{i-1},t) = \frac{w(x_{i-1},t)-w(x_{i-2},t)}{h_{i-1}}(1+h_{i-1}P_{i-1}^\diamond(x_{i-1},t)) \tag{55}$$

and

$$\frac{dw_{i-1}^*}{dx}(x_{i-1},t) = \bar{u}_{i-1/2}(t)(1+h_{i-1}P_{i-1}^\diamond(x_{i-1},t)). \tag{56}$$

Hence equation (52) may be written as

$$\bar{u}_{i+1/2}(t_{n+1}) = \bar{u}_{i+1/2}(t_n)$$
$$-\frac{a\delta t}{h_i}\left[\bar{u}_{i+1/2}(t)(1+h_iP_i^\diamond(x_i,t_n)) - \bar{u}_{i-1/2}(t)(1+h_{i-1}P_{i-1}^\diamond(x_{i-1},t_{n-1}))\right]. \tag{57}$$

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

$$0 \le \frac{a\delta t}{h_i}(1+h_iP_i^\diamond(x_i,t_n)) \le 1, \tag{58}$$

$$0 \le \frac{a\delta t}{h_i}(1+h_{i-1}P_{i-1}^\diamond(x_{i-1},t_n)) \le 1. \tag{59}$$

For a sufficiently small Courant number this follows from Theorem 2 and from the bounds provided by equations (45) and (46). Harten style positivity, see Borisov, [4]. requires that when all solution values are constant that $\bar{u}_{i+1/2}(t_{n+1}) = \bar{u}_{i+1/2}(t_n)$. This follows immediately as all divided differences are zero in this case. It is worth remarking that positivity of the averaged values is different from the scalar case as the averaged exact advection equation solution values (denoted with the superscript e) satisfy:

$$\bar{u}^e_{i+1/2}(t_{n+1}) = \lambda_1\bar{u}^e_{i+1/2}(t_n) + \lambda_2\,\bar{u}^e_{i-1/2}(t_n) \tag{60}$$

where

$$\lambda_1 = \frac{\int_{x_i}^{x_{i+1}-a\delta t} u(x,t_n)dx.}{\int_{x_i}^{x_{i+1}} u(x,t_n)dx.} \text{ and } \lambda_2 = \frac{\int_{x_i-a\delta t}^{x_i} u(x,t_n)dx.}{\int_{x_{i-1}}^{x_i} u(x,t_n)dx.}.$$

Clearly $\lambda_1 + \lambda_2 \neq 1$ in general and there exist functions for which $\lambda_1 = \lambda_2 = 1$, such as the function that has value one on $[x_i - a\delta t, x_{i+1} - a\delta t]$ at time t_n and is zero elsewhere. Conservation is thus in the global sense in that

$$\sum_i \overline{u}^e_{i+1/2}(t_{n+1}) = \sum_i \overline{u}^e_{i+1/2}(t_n). \tag{61}$$

Positivity of the numerical solution value $U_i(t)$ at the mesh point x_i requires a further step. From equation (51) the numerical solution values satisfy:

$$\frac{dw_i^*}{dx}(x_i,t) - \frac{dw_{i-1}^*}{dx}(x_{i-1},t) = U_i(t) - U_{i-1}(t), \tag{62}$$

and hence that

$$U_i(t) = \sum_{j=1}^{i}\left[\frac{dw_j^*}{dx}(x_j,t) - \frac{dw_{j-1}^*}{dx}(x_{j-1},t)\right] + u_0(t). \tag{63}$$

From equation (56), and if $\frac{dw_0^*}{dx}(x_0) = u_0(t)$, it follows that

$$U_i(t) = \overline{u}_{i+1/2}(t)(1 + h_i P_i^{\diamond}(x_i,t)), \tag{64}$$

and that positivity of the numerical averaged values $\overline{u}_{i+1/2}(t)$ implies positivity of the numerical solution values $U_i(t)$. In order to show that the values $U_i(t)$ satisfy positivity, equation (57) is rewritten as:

$$U_i(t_{n+1}) = (1 + h_i P_i^{\diamond}(x_i,t_{n+1}))\left[U_i(t_n)\left[\frac{1}{(1 + h_i P_i^{\diamond}(x_i,t_n))} - \frac{a\delta t}{h_i}\right] + U_{i-1}(t_n)\frac{a\delta t}{h_i}\right].$$

This equation may be rewritten in the form of equation (2) as:

$$U_i(t_{n+1}) = \frac{(1 + h_i P_i^{\diamond}(x_i,t_{n+1}))}{(1 + h_i P_i^{\diamond}(x_i,t_n))}\left[U_i(t_n)(1-\alpha_i) + U_{i-1}(t_n)\alpha_i\right].$$

where, from equations (58) and (59), the positive term α_i is

$$0 \leq \alpha_i = (1 + h_i P_i^{\diamond}(x_i,t))\frac{a\delta t}{h_i} \leq 1. \tag{65}$$

The data bounded polynomial approach used here guarantees that this term is positive. The condition for strict positivity given by

$$(1 + h_i P_i^{\diamond}(x_i,t_{n+1})) \leq (1 + h_i P_i^{\diamond}(x_i,t_n)). \tag{66}$$

may be obtained by varying the order used to create the polynomial that is used to compute $U_i(t)$, via equation (64).

3.1 Dealing with Extrema in ENO Methods

Suppose that possible extrema in the solution are detected by tests such as the simple test in Section 2.3. The bounded polynomial approach is not appropriate in its original form but may be modified in a way that allows a bounded extremal value to be created. Assuming that the value of the difference $u[x_0,x_1]$ is small and that λ_2 as in equation (39) violates (14,15) and satisfies

$$|\lambda_2| \geq 1. \tag{67}$$

The derivatives in equations (41,42) then have opposing signs at $x = x_0$ and $x = x_1$. Rewriting equations (20,21) to define a modified interpolation polynomial $\hat{U}^l(x)$by:

$$\hat{U}^l(x) = U[x_0] + \lambda_2 U[x_0,x_1]\hat{P}_N(s) \tag{68}$$

where the term $\lambda_2 U[x_0,x_1]$ is the second difference defined as in Section 2 and where

$$\hat{P}_N(s) = s(\frac{1}{\lambda_2} + \frac{(s-1)}{D_2}\ (1 + \frac{(s-e_2)}{D_3}\lambda_3 \times\ (1 + \frac{(s-e_3)}{D_4}\lambda_4 \times \ldots 1 + \frac{(s-e_{N-1})}{D_N}(\lambda_N).$$

Assuming that the signs of λ_2 and $u[x_0,x_1]$ are the same it follows from Theorem 2 that a bounds on any new maximum value of $\hat{U}^l(x)$ are given by:

$$U[x_0] \leq \hat{U}^l(x) \leq U[x_0] + [U(x_1) - U(x_0)]\lambda_2, \forall x \in [x_0,x_1] \tag{69}$$

A similar argument may be made when the signs of λ_2 and $u[x_0,x_1]$ are different and a minimum is created.

3.2 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 $P_i^{\diamond}(x_i,t)$ and $P_{i-1}^{\diamond}(x_{i-1},t)$ satisfy equations (58) and (59). Hence an alternative approach to seeking positivity is to simply require that the order of the ENO method be chosen so that this is the case. 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

$$|u(x_i,t) - u(x_{i-1},t)| \leq TOL. \tag{70}$$

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 derived in Section 3.

4.1 ENO truncation Error

The classical spatial truncation error for ENO methods may be calculated from the exact solution $u_e(x,t)$ by first calculating $\overline{u}_{i+1/2}(t)$ and then 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

$$TE_{eno}(x,t) = \frac{u_e(x_i,t) - u_e(x_{i-1},t)}{h_i} - \frac{1}{x_i - x_{i-1}} . \left[\frac{dw_i^*}{dx}(x_i) - \frac{dw_{i-1}^*}{dx}(x_{i-1}) \right] . \qquad (71)$$

This truncation error provides a measure of how accuracy of an ENO method based upon these high-order polynomial expressions can be for the problem being solved and so will be used to investigate the performance of the different ENO-based methods.

4.2 Computational Experiments

The following three examples have been used in the past to demonstrate the properties of hyperbolic equation solvers 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, or the polynomial order plus one. The method denoted by BENO is the bounded polynomial approach defined by Section 3, without any method for re-inserting extrema. The method denoted by LENO is the extension of the limited BENO approach defined by Section 3.1 to switch off limiting where there are extrema. This method gives almost identical results to the unlimited ENO method. For the sake of brevity we refer the reader to [2] where the Gaussian example was used by Rider at al [8], to illustrate the advantage of using high-order methods for problems with smooth solutions shows similar results for both methods. In this case, given the smoothness of the solution, it is not surprising that the best results were obtained with polynomials of degree 12 or higher. With the problem involving the advection of $u(x,t) = sin^4(x)$, a problem considered by Shu [13] and others, the best results are obtained with polynomials of degree 12 or higher. In this case the BENO method is less accurate than the LENO method and the original ENO method as lower-order polynomials are used at extrema, see [2].

4.2.1 Steep Front Problem

The results of Rider [8] show that high-order methods may not be substantially 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 [10], the 11th order polynomial:

$$u_e(x,t) = z^6 \left[-252z^5 + 1386z^4 - 3080z^3 + 3465z^2 - 1980z + 462 \right] \qquad (72)$$

where $z = (0.5 + t + ds * 0.5 - x)/ds$; and which has a front of width ds whose center position is at $0.5 + t$. Three sets of numerical experiments were conducted with this problem in order to examine the performance of the high-order methods in the presence of a progressively steeper front. 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 $ds = 0.0096$. In cases (b) and (c) the number of mesh points inside the front is low as shown by Table 2.

Steep Front	NPTS					
ds	15	31	63	127	255	511
0.0096	1	1	1	1	3	5
0.096	1	3	5	11	23	53

Table 2 Number of Points Inside Steep Front.

Figure 2 shows the solution profiles in all three cases. The experiments were conducted with equally spaced meshes and allowing the polynomial degree to vary to 23. The L1 norms of the truncation errors in Table 3 show that with large values of $ds > 0.1$, say, using high order polynomials leads to an improvement in accuracy. 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, $NP = 3$. The conclusions from these experiments are that for smooth solutions where there is enough mesh resolution there are advantages in using high order polynomials of order 6-12. The effective polynomial order does tend to be limited by the number of mesh points in a front, as noted by [1]. The results from these experiments demonstrate the need to have multiple points in the region of a shock front to get high accuracy with high-order methods. One way of achieving this is to use the nonuniform meshes permitted by Theorem 1.

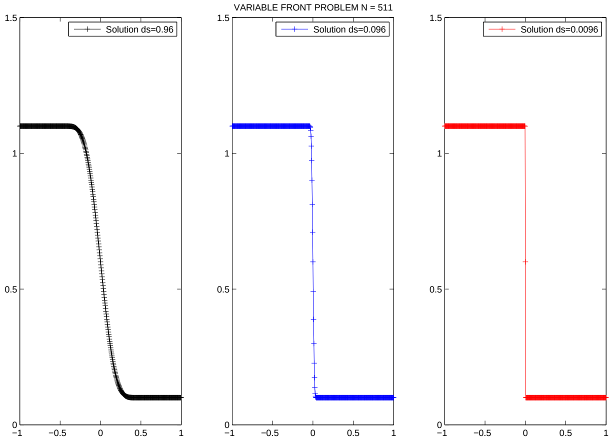

Fig. 2 Problem 3:Steep Front Example Solutions, ds=0.96,ds=0.096,ds=0.0096

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

Steep Front	ds=0.96	NPTS					
Method	NP	15	31	63	127	255	511
BENO	3	1.8e-2	6.5e-3	1.3e-3	8.4e-5	3.3e-5	7.3e-6
LENO	3	1.1e-2	8.5e-4	6.5e-5	4.1e-6	2.5e-7	1.6e-8
BENO	6	5.3e-3	1.7e-4	5.3e-6	7.9e-8	1.8e-9	4.0e-11
LENO	6	7.8e-3	8.8e-5	1.6e-6	2.0e-8	2.0e-10	8.7e-12
BENO	12	5.9e-2	8.9e-5	2.8e-7	2.7e-9	2.3e-11	7.5e-12
LENO	12	7.1e-2	1.5e-4	2.5e-7	1.9e-9	2.0e-11	9.1e-12
BENO	24	5.9e-2	1.8e-2	1.6e-4	7.2e-10	8.2e-12	5.8e-12
LENO	24	2.6e-1	1.1e-1	2.3e-4	6.8e-10	8.0e-12	5.7e-12
Problem 3	ds=0.096						
BENO	3	1.3e-1	5.5e-2	1.7e-2	2.6e-3	4.6e-4	7.1e-5
LENO	3	1.3e-1	4.7e-2	1.2e-2	1.3e-3	1.7e-4	1.4e-5
BENO	6	1.2e-1	4.2e-2	7.7e-3	7.2e-4	2.5e-5	1.1e-6
LENO	6	1.2e-1	4.0e-2	7.7e-3	5.0e-4	1.9e-5	4.6e-7
BENO	12	5.7e-1	3.4e-2	6.2e-3	2.6e-4	5.5e-6	8.3e-8
LENO	12	1.4e-0	3.9e-2	6.1e-3	2.2e-4	5.4e-6	6.2e-8
BENO	24	5.7e-1	2.5e-1	6.1e-3	1.7e-4	2.8e-6	3.0e-8
LENO	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 3 Comparison of L1 Norms of Truncation Errors for Steep Front Problem of Hubbard.

to use much higher order methods than is often done with ENO methods. Achieving the appropriate spatial order is somewhat more problematical as 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 existing extrema in an accurate way without introducing new extrema elsewhere.

References

1. M. Berzins, *Adaptive polynomial interpolation on evenly spaced meshes*, SIAM Review **1** (2007), no. 4, 624–627.
2. Berzins M. Data Bounded Polynomials and Preserving Positivity in High Order ENO and WENO Methods *SCI Report UUSCI-2009-003* (unpublished) University of Utah, July 2009, Revised March 2010.
3. Balsara D.S. and Shiu C.W. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computat. Physics* **160**:405-452 (2000).
4. Borisov V.S. and Sorek S. On monotonicity of difference schemes for computational physics. *SIAM J. Sci. Comput.* **25**:1557-1584, 2004.
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. Graillat S. Langlois P. Louvet N. Improving the compensated Horner scheme with a fused multiply and add. Proc. 2006 ACM symp. on Applied Comput., 1323-1327, 2006 ISBN:1-59593-108-2, ACM, NY, USA.
7. Higham N.J. Accuracy and Stability of Numerical Algorithms. Siam, Philadelphia 1996.
8. Greenough J.A. and Rider W.J. A quantitative comparison of numerical methods for the compressible Euler equations: mifth order WENO and piecewise-linear Godunov *J. of Computat. Physics* 2004; **196**:259–281.
9. Hildebrand F.B. *Introduction to Numerical Analysis.* McGraw-Hill Book Company Inc 1956;
10. 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) CFD 2002, pp. 205-210 Springer-Verlag. 2003.

11. C.-W. Shu, TVD properties of a class of modified ENO schemes for scalar conservation laws, IMA Preprint Series 308 (1987), University of Minnesota.
12. Shu C-W . High order WENO schemes. *SIAM Review* March 2009; **51**:82-126.
13. C.-W. Shu, Numerical experiments on the accuracy of ENO and modied ENO schemes, *Journal of Scientic Computing*, v5 (1990), pp.127-149.
14. Waterson N.P. and Deconinck H. Design Principles for bounded higher-order convection schemes - a unified approach. *J. of Computat. Physics* 2007; **224**: 182-207.
15. X. Zhang and C.-W. Shu, A genuinely high order total variation diminishing scheme for one-dimensional scalar conservation laws, Brown University preprint, to appear in SIAM Journal on Numerical Analysis.