SCI INSTITUTE
TECHNICAL REPORT

Adaptive Polynomial Interpolation on Evenly Spaced Meshes

Martin Berzins

UUSCI-2006-033

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

November 1, 2006

Abstract:

The problem of oscillatory polynomial interpolants arising from equally spaced mesh points is considered. It is shown that by making use of adaptive approaches that the oscillations may be contained and that the resulting polynomials are data bounded and monotone on each interval. This is achieved at the cost of using a different polynomial on each sub-interval. Computational results for a number of challenging functions including a number of problems similar to Runge's function with as many as 511 points per interval are shown.
keywords Adaptive polynomial interpolation, data bounded polynomials, Runge's function
This report is to appear in SIAM Review 2007.

THE UNIVERSITY OF UTAH

ADAPTIVE POLYNOMIAL INTERPOLATION ON EVENLY SPACED MESHES

M. BERZINS [‡]

Abstract. The problem of oscillatory polynomial interpolants arising from equally spaced mesh points is considered. It is shown that by making use of adaptive approaches that the oscillations may be contained and that the resulting polynomials are data bounded and monotone on each interval. This is achieved at the cost of using a different polynomial on each sub-interval. Computational results for a number of challenging functions including a number of problems similar to Runge's function with as many as 511 points per interval are shown.

Key words. Adaptive polynomial interpolation, data bounded polynomials, Runge's function

AMS subject classifications. 65D05, 65D25

1. Introduction. Two central themes in scientific computing are the inadequacy of polynomial interpolation using evenly spaced meshes and the development of numerous algorithms to model the solution of ordinary and partial differential equations on the very same meshes, often in the case of hyperbolic equations using techniques to suppress oscillations. The resolution of these different points occurs mostly through the use of relatively low order polynomials in many differential equations algorithms based on evenly spaced meshes. At the same time the natural trend in the development of new algorithms for differential equations is towards the use of higher order methods. Good examples of this trend are ENO and WENO schemes, [2, 18] and spectral methods [17]. One way of resolving the issue of potentially poor properties of equally spaced meshes is to use spectral methods based on Chebyshev or Legendre polynomials and associated meshes. Nevertheless there are many situations where it is desirable to use equally spaced meshes because of algorithmic, code or data constraints. There are also many situations in which it is important to use an interpolant that preserves the positivity of the data. Chemical concentrations, for example, must be positive; but there are many other cases. This in turn suggests that it is worth revisiting the properties of interpolation on equally spaced grids with an eye to the developments in nonlinear interpolation methods so successfully used to control oscillations in the solution of hyperbolic equations.

This paper follows earlier work of Berzins [5–7] which is concerned with the development of positivity preserving finite element methods for the solution of hyperbolic equations in one space variable. This paper extends the approach of Berzins by exploring the approximation properties of an adaptive bounded polynomial approximation method. This will be achieved by building on key ideas of Harten [19] and of ENO methods [18]. 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 algorithm used will limit the signs and growth of divided difference terms to arrive at bounded monotone polynomial approximations of possibly arbitrarily high degree within an interval. While this limiting process may be used with any divided difference polynomial, its use in conjunction with ENO schemes is natural in that both approaches seek to control the size of the differences used in divide difference polynomial schemes. The price that will be paid for this is that the resulting polynomials will only be C^0 continuous at data points.

2. Divided Difference Polynomial Interpolation. The approximation space used here is polynomials on $[a,b]$. 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

[‡]SCI Institute, University of Utah, Salt Lake City UT 84112 , USA

points.

Consider the case in which we have an approximating polynomial $U^L(x)$ based on a set of nodal values $U(x_i)$ which may be evenly spaced when $a=-1$ and $b=1$:

$$x_i = -1 + 2i/N, \;\; i = 0, ..., N \tag{2.1}$$

Interpolation using these values may be performed using a number of techniques such as Lagrange interpolation as defined by

$$U^L(x) = \sum_{i=0}^{N} l_i(x) U(x_i) \tag{2.2}$$

where $l_i(x)$ is the polynomial of degree N with $l_i(x_j) = \delta_{ij}$, and where δ_{ij} is the Kronecker delta. An alternative is to use Newton divided difference interpolation. The pros and cons of these approaches are discussed by [3]. In this paper we will use divided differences as defined by the usual notation where $U[x_i] = U(x_i)$ and

$$U[x_i, x_{i+1}] = \frac{U[x_{i+1}] - U[x_i]}{x_{i+1} - x_i}, \tag{2.3}$$

and subsequent differences are defined recursively

$$U[x_i, x_{i+1}, ..., x_{i+k}] = \frac{U[x_{i+1}, x_{i+2} ..., x_{i+k}] - U[x_i, x_{i+1}, ..., x_{i+k-1}]}{x_{i+k} - x_i}. \tag{2.4}$$

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 $U^L(x)$is given by

$$\begin{aligned} U^L(x) &= U[x_i] + \pi_{1,i}(x)\, U[x_i, x_{i+1}] + \pi_{2,i}(x)\, U[x_i, x_{i+1}, x_{i+2}] \\ &+ \pi_{3,i}(x)\, U[x_i, x_{i+1}, x_{i+2}, x_{i+3}] + ... + + \pi_{N,i}(x)\, U[x_i, ..., x_{i+N}], \end{aligned} \tag{2.5}$$

where

$$\begin{aligned} \pi_{1,i}(x) = (x - x_i), \;\; \pi_{2,i}(x) = (x - x_i)(x - x_{i+1}), \\ \pi_{3,i}(x) = (x - x_i)(x - x_{i+1})(x - x_{i+2}), \; etc \end{aligned} \tag{2.6}$$

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, [20]. 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 (2.4) will also then be the width of the stencil of points used to evaluate the difference.

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 (2.5) 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

$$U^L(x) = U[x_i] + \pi_{1,i}(x)\, U[x_i, x_{i+1}] + \pi_{2,i}(x)\, U[x_{i-1}, x_i, x_{i+1}] \tag{2.7}$$

with the same values of the functions $\pi_{1,i}(x)$ and $\pi_{2,i}(x)$: The central idea in ENO methods [18] is to pick the polynomial with the smallest divided differences in order to potentially reduce oscillations. In the above case if

$$|U[x_{i-1},x_i,x_{i+1}]| < |U[x_i,x_{i+1},x_{i+2}]| \tag{2.8}$$

then the polynomial defined by equation (2.7) is used rather than the polynomial defined by the first three terms on the right side of equation (2.5). As an illustration, it is now straightforward to illustrate how polynomials such as these may be modified so that the value of a polynomial on the interval $[x_i,x_{i+1}]$ is bounded by $u[x_i]$ and $u[x_{i+1}]$ the polynomial defined by the first three terms on the right side of equation (2.5) may be written as

$$U^L(x) = U[x_i] + U[x_i,x_{i+1}](x-x_i)(1+\left[\frac{U[x_{i+1},x_{i+2}]}{U[x_i,x_{i+1}]}-1\right]\frac{(x-x_{i+1})}{(x_{i+2}-x_i)}) \tag{2.9}$$

which, assuming that the mesh is evenly spaced i.e. $x_{i+2}-x_{i+1} = x_{i+1}-x_i$, may be rewritten as

$$U^L(x) = U[x_i] + (U[x_{i+1}]-U[x_i])(s+[r-1]\frac{s(s-1)}{2}) \tag{2.10}$$

where $r = \frac{U[x_{i+1},x_{i+2}]}{U[x_i,x_{i+1}]}$ and $0 \le s = \frac{(x-x_i)}{(x_{i+1}-x_i)} \le 1$. For the interpolant to be bounded on the interval by $U[x_{i+1}]$ and $U[x_i]$ hence requires that

$$0 \le (s+[r-1]\frac{s(s-1)}{2}) \le 1 \tag{2.11}$$

It is straightforward to show that this holds for $|r-1| \le 2$. A common procedure in finite volume methods for hyperbolic p.d.e.s is to limit the solution by replacing r by a function $\Phi(r)$ which is known as a limiter function as it is designed to meet any required limits on r.

The polynomial defined by equation (2.7) may, in a similar way as above, be rewritten as

$$U^L(x) = U[x_i] + (U[x_{i+1}]-U[x_i)](s+[1-r]\frac{s(s-1)}{2}) \tag{2.12}$$

where now $r = \frac{U[x_{i-1},x_i]}{U[x_i,x_{i+1}]}$ and for the interpolant to be bounded on the interval by $U[x_{i+1}]$ and $U[x_i]$ requires that

$$0 \le (s+[1-r]\frac{s(s-1)}{2}) \le 1, \tag{2.13}$$

which leads to the same requirement on the modified ratio r that $|r-1| \le 2$.

The above examples illustrates the general process that is taken for arbitrary degree polynomials, once the underlying interpolation theory is discussed in the remainder of this section. Sections 3 and 4 describe the ENO type approach in general. A recursive formulation of the general polynomial similar to that in equation (2.10) is given in Section 5 and written in full in Section 6. A number of numerical experiments are described in Section 7 and the theoretical properties analyzed in Section 8.

2.1. Limitations of Evenly Spaced Data Interpolation. Numerous texts in Numerical Analysis show that evenly spaced mesh polynomial interpolation with evenly spaced meshes breaks down on Runge's function when the function $U(x)$ is defined by

$$U(x) = \frac{1}{1+25x^2} \tag{2.14}$$

as oscillations in the solution become increasingly large with increasing polynomial degree. Figure (2.1) shows these oscillations for polynomials of order 4,7,10 and 13.

There is a large body of approximation theory regarding the properties of interpolation based on equally spaced points; Revers provides an excellent summary, [26]. Many authors such as de Boor [11, 29] show that polynomial approximation based on the expanded Chebyshev points as defined by

$$x_i = -cos((2i+1)\pi/(2N+2))/cos(\pi/(2N+2)),\ i = 0,...,N \tag{2.15}$$

is "nearly optimal" in that the error is close to that of an optimal polynomial. A key factor in this argument is the Lebesgue constants, Λ_N, as defined by:

$$\Lambda_N = ||\sum_{i=0}^{N} l_i(x)||_\infty. \tag{2.16}$$

where $||f(x)||_\infty = max|f(x)|$ for $a \le x \le b$. It is worth remarking, see [29], that for evenly spaced nodes Λ_N has the asymptotic behavior that

$$\Lambda_N \sim \frac{2^{N+1}}{e\ N\ log(N)}\ as\ N \to \infty, \tag{2.17}$$

whereas for the expanded Chebyshev points de Boor, [11], quotes Brutman's result that

$$\frac{2}{\pi} log(N+1) + 0.5 \le \Lambda_N \le \frac{2}{\pi} log(N+1) + 0.73\ . \tag{2.18}$$

The importance of these results is that the Lebesgue constant occurs in the inequality

$$||U - U^L||_\infty \le (1+\Lambda_N)||U - U^*||_\infty, \tag{2.19}$$

where $U^*(x)$ is the polynomial of best approximation to $U(x)$ on $[a,b]$, see Trefethen and Weideman [29], who state that $U^L(x)$ may fail to converge to $U(x)$ if Λ_N grows too quickly. The issues related to the convergence of polynomials are best described by complex plane analysis e.g. see [13, 30]. As well as analyzing the complex case Epperson, [13], points out that *the derivative term must grow quite rapidly to force divergence*, where, in this context derivative refers to the $N+1$th order derivative $U^{(N+1)}$, and proves that if

$$\lim_{N \to \infty} \left[\frac{||U^{(N+1)}||}{\sqrt{N+1}} \left(\frac{e(b-a)}{N+1} \right)^{N+1} \right] = 0 \tag{2.20}$$

then the interpolant $U^L(x)$ based upon $N+1$ uniformly spaced nodes in the interval $[a,b]$ converges to $U(x)$ uniformly. The significance of this is that if the higher derivatives are sufficiently well-bounded then it may be possible, but possibly far from optimal, to use polynomial interpolation based upon evenly spaced data values. Indeed this is in some sense the strategy used by the ENO methods, [18], described in Sections 3 and 4, although these methods do not attempt to construct global polynomial approximations.

2.2. Interpreting the Theory. The above theoretical results require interpreting with some care. For example, with reference to the result (2.18) Epperson [13] (p337) states *The above discussion also appears to contradict the oft-read dictum "divergence of interpolation polynomials is due to the growth of the $l_i(x)$ functions" While the importance of the $l_i(x)$ should not be understated, equation (10) clearly indicates that interpolation will converge for*

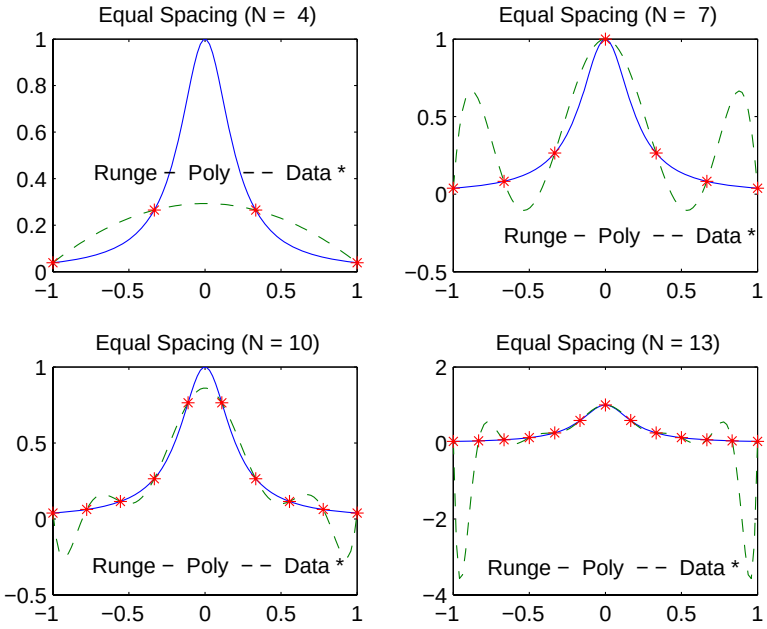

FIG. 2.1. *Runge's Function* $1/(1+25x^2)$*denoted by Runge, Data Points denoted by Data and Even Mesh Polynomial Interpolation using these Data Points denoted by Poly*

functions whose derivatives behave moderately well as $N \to \infty$. Equation (10) in this quote is an intermediate result leading to equation (2.19) above. Of course polynomial interpolation using equally spaced points may still be ill-conditioned. Epperson goes on to give examples of Runge-type functions for which evenly spaced interpolation does converge on smaller sub-intervals.

It is straightforward to construct examples which illustrate some of these issues. For example if one half of Runge's function, say on $[-1,0]$, is approximated by polynomials using evenly-spaced points the minimum L_∞ error of about 8×10^{-8} occurs when a polynomial of degree 44 is used. The norm here is approximately calculated by 2049 equally spaced sample points in the interval $[0,1]$. The error using the Chebyshev points spacing of equation (2.15) is considerably smaller however, 8×10^{-19} and occurs with a polynomial degree of 100. In both cases polynomials of degree close to 44 or 100 respectively give very similar results.

The task then is to find the algorithms to enable equally spaced points to be used in interpolation without spurious oscillations resulting. In this respect this paper will show that adaptive methods and the use of ENO type piecewise polynomial approximations have an important role to play.

2.3. Solutions to the Problem of Oscillations. There are a number of important ways of addressing the problem of oscillating solutions to non-oscillatory data. Gottlieb and Shu [16] describe approaches involving Gegenbauer polynomials for the case of Gibbs oscillations in Fourier series and reference an important series of their papers on this topic. Recent work in this area confirms the great promise of this approach [15, 28] but also indicates that there are outstanding issues, [9].

The polynomial filtering methods of Gottlieb and Shu [16] [17] modify the polynomial

coefficients so as to improve accuracy in the presence of discontinuities. There are some similarities with the approach to be discussed here, see [6].

An alternative solution is to make use of the many successful shape preserving constrained spline approximations such as PCHIP which is a C^1 cubic spline method [14] in which a system of equations is solved as part of the process to construct the interpolation function. Other approaches are the methods in [23, 25]. A general description of the oscillatory nature of many interpolants is given by Jerri, [21]. The approach developed here may be viewed as one type of such a spline approximation, the major differences being that C^1 smoothness is not imposed and that the polynomial order may be arbitrarily high here.

Part of the motivation for this study is that many calculations in fluid mechanics make use of evenly spaced meshes, mostly with low order methods but increasingly with higher order, [2], for systems of conservation laws for which we expect the solution to remain positive and stay bounded [8]. It is thus important to consider how we can devise interpolants on evenly spaced meshes that are bounded by the values at the ends of the interval. In keeping with the definition used by Berzins, [4] such interpolants are referred to as *data-bounded*. In this context a polynomial $U^l(x)$ is said to be data-bounded if

$$\begin{aligned} & U^l(x_i) = U(x_i) \\ & U^l(x_{i+1}) = U(x_{i+1}) \\ & min(U(x_i), U(x_{i+1})) \le U^l(x) \le max(U(x_i), U(x_{i+1})),\ x \in [x_i, x_{i+1}]. \end{aligned} \tag{2.21}$$

If possible we would also like the values produced by the interpolant to be monotone increasing or decreasing as appropriate within an interval.

In this paper three ideas from algorithms in the numerical solution of advection-type equations will be applied to ensure that the values of the polynomial interpolant are bounded by the data values used.

- Adapt the stencil so as to use the smallest divided differences at any stage as is done in ENO methods to reduce oscillations, [18].
- Alter the degree of polynomial approximation so that any discontinuities in higher derivatives are removed, [19].
- Alter the polynomial degree and/or terms so that successive differences in the series do not change dramatically, [5, 6].

The combined effect of these steps will be shown to eliminate oscillations by ensuring that values generated in an interval are bounded by the data point values at either end and are monotone.

3. ENO Schemes - Picking the Smallest Divided Difference . In the remainder of this paper it is convenient to modify the description of the mesh used so that an evenly spaced mesh is used and the mesh points, x_i, are defined around a point x_0 by adding or subtracting integer multiples of an even mesh spacing as denoted by h

$$x_i = x_0 + ih. \tag{3.1}$$

and that the mesh is defined by

$$a = x_{-J} < x_{-J+1} < \ldots < x_{-1} < x_0 < x_1 < \ldots x_K = b \tag{3.2}$$

for non-negative integers J and K, where this may be viewed as a generalization of equation (2.1), with $N = K$ and $J = 0$ in that case. On this mesh we will seek to interpolate the function $U(x)$ for $x \in [x_0, x_1]$ On such a mesh a systematic way of choosing the direction of a difference stencil is given by the widely-used Essentially Non-Oscillatory Schemes, [18].

Consider having formed the divided difference defined by equation (2.4) above. The next divided difference could be obtained by extending the stencil to x_{i-1} to get

$$U[x_{i-1},x_i,x_{i+1},...,x_{i+k}] = \frac{U[x_i,x_{i+1},x_{i+2}...,x_{i+k}]-U[x_{i-1},x_i,x_{i+1},...,x_{i+k-1}]}{x_{i+k}-x_{i-1}}, \tag{3.3}$$

or could be obtained by extending the stencil to x_{i+k+1} to get

$$U[x_i,x_{i+1},...,x_{i+k},x_{i+k+1}] = \frac{U[x_{i+1},x_{i+2}...,x_{i+k+1}]-U[x_i,x_{i+1},...,x_{i+k-1},x_{i+k}]}{x_{i+k+1}-x_i}, \tag{3.4}$$

The central idea behind ENO methods is to attempt to reduce oscillations by choosing between these alternatives, for instance, by choosing the smallest of the absolute values of the two differences and so to adaptively vary the stencil used to define the polynomial. There is no guarantee, but much experimental evidence, that oscillations are either eliminated or reduced by using this method with polynomials of arbitrary degree. In Section 8 we will show that it is possible to prove that this is indeed the case. An extension of ENO methods is the WENO schemes in which a weighted combination of the possible different interpolation polynomials is used [2].

4. Removing Sign Changes in Differences using Harten's Approach. Although ENO schemes allow us to choose the smallest divided differences at any stage they do not control the growth in those differences within the interpolating sequence. In this respect a significant observation (and one consistent with [13]) was made by Harten [19] (see also Arandiga et al. [1], p.9) that the lack of smoothness (or even an approximate lack of smoothness) may effect the quality of the approximation. For example when a steep gradient is found close to areas of zero gradient and may then appear like a discontinuity. Consider the second divided difference

$$U[x_{i-1},x_i,x_{i+1}] = \frac{U[x_i,x_{i+1}]-U[x_{i-1},x_i]}{(x_{i+1}-x_{i-1})} \tag{4.1}$$

and suppose that $U[x_{i-1},x_i] = O(\varepsilon)U[x_i,x_{i+1}]$. It then follows that

$$U[x_{i-1},x_i,x_{i+1}] = (1-O(\varepsilon))\frac{U[x_i,x_{i+1}]}{(x_{i+1}-x_{i-1})} \tag{4.2}$$

and since it is possible that $(x_{i+1}-x_{i-1}) < 1$ the factor $\frac{(1-O(\varepsilon))}{(x_{i+1}-x_{i-1})}$ is an amplification factor and the second divided difference is larger than the first.

In the case when neighboring divided differences have different signs, for example when $U[x_{i-1},x_i] = -\mu U[x_i,x_{i+1}]$ for $\mu > 0$), it follows that

$$U[x_{i-1},x_i,x_{i+1}] = (1+\mu)\frac{U[x_i,x_{i+1}]}{(x_{i+1}-x_{i-1})} \tag{4.3}$$

and again the second divided difference is larger than the first. In both these cases the error may increase by using second or higher differences.

4.1. An Example using Runge's function. For example consider Runge's function as defined by equation (2.14) interpolated at $x = -0.83$ with polynomials of order 7 defined using data at evenly spaced points and at $x = -0.91$ using the extended Chebyshev points. Table 4.1 shows the terms that make up the divided difference form of the interpolating polynomial. The divided differences in *italics* are computed as results of sign changes in lower

TABLE 4.1
Components of Polynomial Approximations

$u(x_i)$	*Mesh*		j=1	j=2	j=3	j=4	j=5	j=6
0.038	Evenly	$\pi_{j,i}(x)$	0.17	-0.028	0.014	-0.012	0.014	-0.020
	Spaced	Divided Differences	0.013e	0.62	1.9	*-8.2*	*13.0*	*-13.0*
0.038	Cheby	$\pi_{j,i}(x)$	0.095	-0.0098	0.0045	-0.0041	0.0055	-0.0094
	Spaced	Divided Differences	0.1	0.37	1.6	*-4.5*	*5.9*	*-5.9*

derivatives. For this example both evenly spaced and Chebyshev polynomials at this order oscillate markedly. One obvious step is to form a polynomial interpolant for each spatial interval and then to eliminate any divided differences produced as a result of sign changes in the divided differences used to form them. In other words with respect to equation (3.3)

$$U[x_{i-1},x_i,x_{i+1},...,x_{i+k}] \;=\; 0 \;\; if \;\; \frac{U[x_i,x_{i+1},x_{i+2}...,x_{i+k}]}{U[x_{i-1},x_i,x_{i+1},...,x_{i+k-1}]} < 0 \tag{4.4}$$

or in the case of equation (3.4)

$$U[x_i,x_{i+1},...,x_{i+k},x_{i+k+1}] \;=\; 0 \;\; if \;\; \frac{U[x_{i+1},x_{i+2}...,x_{i+k+1}]}{U[x_i,x_{i+1},...,x_{i+k-1},x_{i+k}]} < 0. \tag{4.5}$$

This leads to a substantial improvement in the polynomial approximation as is shown in Figure (4.1), which shows the effect of doing this on Runge's function with polynomials of degree 6 in both the evenly spaced and Chebyshev cases. In both cases the new method is designated by *Limited* and provides a bounded monotone interpolant. Eliminating divided differences generated by sign changes in lower differences is not enough to guarantee boundedness or monotonicity in general however. The disadvantage of course is that a different

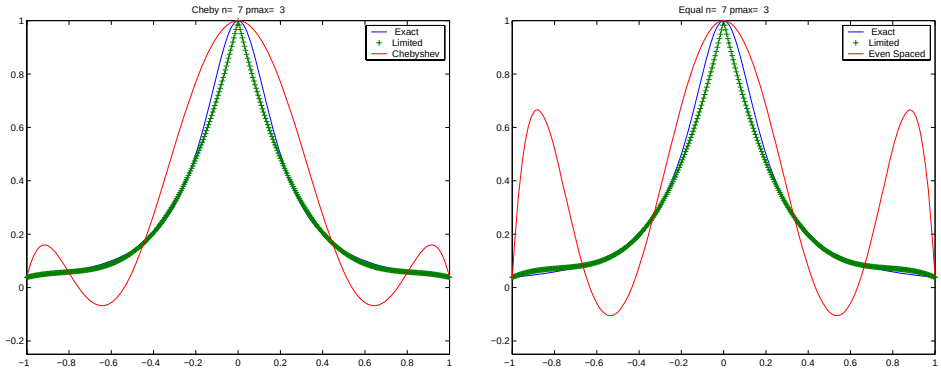

FIG. 4.1. *Eliminating Differences Generated by Sign Changes for Runge's Function*

polynomial is used in each interval and so the composite polynomial is only continuous at the data points and so may be viewed as a C^0 piecewise polynomial spline approximation.

5. Recursive Formulation of Difference Schemes. 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

$$r^{[i,i+1]}_{[i-1,i]} = \frac{U[x_i, x_{i+1}]}{U[x_{i-1}, x_i]}, \tag{5.1}$$

with obvious extensions to higher differences and other indices. As an example by using this notation, when a divided difference approximation incorporates a new point from the right x_{i+k} we get

$$U[x_{i-1}, x_i, x_{i+1}, ..., x_{i+k}] = \frac{\left(r^{[x_i,...,x_{i+k}]}_{[x_{i-1},...,x_{i+k-1}]} - 1\right)}{x_{i+k} - x_{i-1}} U[x_{i-1}, x_i, ..., x_{i+k-1}] \tag{5.2}$$

or an equivalent form when a divided difference approximation incorporates a new point from the left x_{i-1} is

$$U[x_{i-1}, x_i, ..., x_{i+k}, x_{i+k}] = \frac{\left(1 - r^{[x_{i-1},...,x_{i+k-1}]}_{[x_i,...,x_{i+k}]}\right)}{x_{i+k} - x_{i-1}} U[x_i, x_{i+1}, ..., x_{i+k}]. \tag{5.3}$$

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

$$U[x_i, x_{i+1}, x_{i+2}, ..., x_{i+k+1}] = \frac{\left(r^{[x_{i+1},...,x_{i+k+1}]}_{[x_i,...,x_{i+k}]} - 1\right)}{x_{i+k+1} - x_i} U[x_i, x_{i+1}, ..., x_{i+k}]. \tag{5.4}$$

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

$$\frac{\left(|1 - r^{[x_{i-1},...,x_{i+k-1}]}_{[x_i,...,x_{i+k}]}|\right)}{|x_{i+k} - x_{i-1}|} < \frac{\left(|r^{[x_{i+1},...,x_{i+k+1}]}_{[x_i,...,x_{i+k}]} - 1|\right)}{|x_{i+k+1} - x_i|}, \tag{5.5}$$

or picks $U[x_i, x_{i+1}, x_{i+2}, ..., x_{i+k+1}]$ otherwise. In order to write the divided difference interpolating function in recursive form if equation (5.5) holds we define

$$\lambda_{k+1} = \left(1 - r^{[x_{i-1},...,x_{i+k-1}]}_{[x_i,...,x_{i+k}]}\right) \tag{5.6}$$

Alternatively if equation (5.5) does not hold let

$$\lambda_{k+1} = \left(r^{[x_{i+1},...,x_{i+k+1}]}_{[x_i,...,x_{i+k}]} - 1\right). \tag{5.7}$$

Using the notation of equations (5.6) and (5.7) the polynomial defined by equation (2.5), for a set of points $x_0 < x_1 < ... < x_n$ may be rewritten as

$$\begin{aligned} U^I(x) = U[x_0] + (x - x_0)U[x_0, x_1](1 + \frac{(x - x_1)}{(x_2 - x_0)}(\lambda_2 \times \\ (1 + \frac{(x - x_2)}{(x_3 - x_0)}\lambda_3 \, (1 + \frac{(x - x_3)}{(x_4 - x_0)}\lambda_4 \times .... \\ \times (1 + \frac{(x - x_{n-1})}{(x_n - x_0)}\lambda_n)...), \end{aligned} \tag{5.8}$$

where the final set of dots indicate $n-3$ right brackets. Providing that the values of $r^{[\ldots]}_{[\ldots]}$ satisfy the restriction

$$0 \leq r^{[\ldots]}_{[\ldots]} \leq 1 \tag{5.9}$$

then in the case of equation (5.7) it follows that

$$-1 \leq \lambda_k \leq 0 \tag{5.10}$$

and in the case of equation (5.6) it follows that

$$0 \leq \lambda_k \leq 1 \tag{5.11}$$

and that all the subsequent divided differences are bounded by early differences multiplied by the values of λ_k.

5.1. Example. As an example consider the function

$$U(x) = x^P, P > 0 \tag{5.12}$$

on the interval $[0,1]$. In this case as the function is monotone increasing it is straightforward to calculate that

$$r^{[i,i+1]}_{[i-1,i]} > 1 \tag{5.13}$$

but

$$r^{[i-1,i]}_{[i,i+1]} < 1 \tag{5.14}$$

and similarly for other values of of $r^{[\ldots]}_{[\ldots]}$. Hence in interpolating values on the interval $[x_i, x_{i+1}]$ the ENO interpolant will choose a stencil from points including and to the left of this interval, i.e. $[x_{i+1}, x_i, x_{i-1}, x_{i-2}, x_{i-3}\ldots]$.

5.2. Error for ENO Type Approximations. The use of ENO type approximations is subject to standard polynomial error results namely that if a polynomial of degree N is used then the error between the exact and interpolated polynomial satisfies, [20],

$$U(x) - U^L(x) = (x-x_0)(x-x_1)\ldots(x-x_N)\frac{U^{(N+1)}(\xi)}{(N+1)!} \tag{5.15}$$

where $\xi \in [x_0, x_N]$. The different choices of points possible with ENO schemes lead to different forms of this expression as well as different bounds for $U^{N+1}(\xi)$. In the case of the example in Section 5.1 the ENO stencil may well lead to a smaller derivative bound than if the stencil $[x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}\ldots]$ is used.

6. Applying Limiters to Divided Differences. Although it will be possible to chose the smallest divided differences at some point the ratios such as those defined by equation (5.1) will be larger than one. In this case one possible step in reducing oscillations is to limit the size of the divided difference terms by making use of limiting function denoted by $\Phi(r)$ as defined, for example, by $\Phi(r) = \frac{r+|r|}{1+max(1,|r|)}$ which is the modified form of the van Leer harmonic limiter used by Berzins and Ware [7]. The importance of the modified form of the limiter is that for $0 \leq r \leq 1$ the underlying function is not altered.

This limiter function is applied to divided difference terms to define bounded divided differences denoted by $[....]^B$. The bounded differences are defined in terms of the original differences by expressions such as:

$$U[x_{i-1},x_i,x_{i+1}]^B = \frac{\left(\Phi(r^{[i,i+1]}_{[i-1,i]}) - 1\right)}{(x_{i+1}-x_{i-1})}\, U[x_{i-1},x_i], \tag{6.1}$$

which may be written as

$$U[x_{i-1},x_i,x_{i+1}]^B = \frac{S^{[i-i,i]}_{[i-1,i,i+1]}}{(x_{i+1}-x_{i-1})} U[x_{i-1},x_i], \tag{6.2}$$

where

$$S^{[i-1,i]}_{[i-1,i,i+1]} = \left(\Phi(r^{[i,i+1]}_{[i-1,i]}) - 1\right), \tag{6.3}$$

and following the approach of Section 4

$$S^{[i-1,i]}_{[i-1,i,i+1]} = 0 \; if \; r^{[i,i+1]}_{[i-1,i]} < 0. \tag{6.4}$$

Furthermore

$$S^{[i,i+1]}_{[i-1,i,i+1]} = \left(1 - \Phi(r^{[i-1,i-1]}_{[i,i+1]})\right). \tag{6.5}$$

and so from the definition of $\Phi(r)$ it follows that

$$S^{[i,i+1]}_{[i-1,i,i+1]} = S^{[i-1,i]}_{[i-1,i,i+1]} \tag{6.6}$$

and

$$U[x_{i-1},x_i,x_{i+1}]^B = \frac{S^{[i,i+1]}_{[i-1,i,i+1]}}{(x_{i+1}-x_{i-1})} U[x_i,x_{i+1}]. \tag{6.7}$$

The extensions to higher divided differences follow in a similar way. The bounded form of the function $\Phi(r)$ makes it straightforward to bound the functions $S^{[....]}_{[....]}$ by

$$-1 \le S^{[....]}_{[...]} \le 1. \tag{6.8}$$

6.1. Additional Error in Limiting Differences. Consider the illustrative example of second divided differences and define the error in these differences from limiting by

$$e[x_{i-1},x_i,x_{i+1}] = U[x_{i-1},x_i,x_{i+1}] - U[x_{i-1},x_i,x_{i+1}]^B \tag{6.9}$$

From equations (6.1) and (6.5) this may be written as

$$e[x_{i-1},x_i,x_{i+1}] = \frac{\left(r^{[i,i+1]}_{[i-1,i]} - \Phi(r^{[i,i+1]}_{[i-1,i]})\right)}{(x_{i+1}-x_{i-1})}\, U[x_{i-1},x_i], \tag{6.10}$$

which is zero if $0 \le r^{[i,i+1]}_{[i-1,i]} \le 1$. In the alternative case

$$e[x_{i-1},x_i,x_{i+1}] = \frac{\left(r^{[i-1,i]}_{[i,i+1]} - \Phi(r^{[i-1,i]}_{[i,i+1]})\right)}{(x_{i+1}-x_{i-1})}\, U[x_{i-1},x_i], \tag{6.11}$$

which again is zero if $0 \le r^{[i-1,i]}_{[i,i+1]} \le 1$. The extension of this idea to higher order differences is straightforward. Once the limiter is applied to a difference term then it is clear in a formal sense that it is not worth adding subsequent higher divided difference terms as the formal order of accuracy will be dominated by the error introduced by the limiter in the lower term.

6.2. Form of the Limited ENO Polynomial. The form of the polynomial in which the points are chosen so that all successive ratios of derivatives lie between zero and one except for the last $n+1$th term of the series which is then limited may now be written down. Before doing this it is helpful to write down 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.

An evenly spaced mesh is used as in equation (3.1) and the mesh points, x_i, are defined around a point x_0 by adding or subtracting integer multiples of an even mesh spacing so that the mesh points chosen by the ENO approach of Sections 3,4, and 5 at each stage are denoted by x_i^e as defined by

$$x_i^e = x_0 + e_i h,\ i \geq 1 \tag{6.12}$$

for some integer e_i and where $e_1 = 1$. 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

$$x_i^l = min(x_i^e, x_{i-1}^l),\ \ x_0^l = x_0, \tag{6.13}$$

$$x_i^r = max(x_i^e, x_{i-1}^r),\ \ x_0^r = x_0. \tag{6.14}$$

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. Using these definitions allows the limited form of the general ENO polynomial, as defined by equation (5.8), to be written in the form:

$$\begin{aligned} U^I(x) = U[x_0] + (x-x_0)U[x_0,x_1](1 + \frac{(x-x_1)}{(x_1^r - x_1^l)}\lambda_2\ \times \\ (1 + \frac{(x-x_2^e)}{(x_2^r - x_2^l)}\lambda_3\ (1 + \frac{(x-x_3^e)}{(x_3^r - x_3^l)}\lambda_4 \times .... \\ ...(1 + \frac{(x-x_{N-1}^e)}{(x_{N-1}^r - x_{N-1}^l)} S_{[p,...,q]}^{[j,...,k]}), \end{aligned} \tag{6.15}$$

where the value of $S_{[p,...,q]}^{[j,...,k]}$ is defined by the stencil being used, as in equations (6.3) to (6.5). The particular values of p, q, j and k being defined as follows:

$$\begin{aligned} j &= (x_{N-1}^l - x_0)/h \\ k &= (x_{N-1}^r - x_0)/h \\ p &= (x_N^l - x_0)/h \\ q &= (x_N^r - x_0)/h \end{aligned} \tag{6.16}$$

with respect to the mesh defined in equation (3.2) and where $(x_N^r - x_N^l)$ defines the final width of the stencil as described in Section 2. As with the simple example in Section 2, this expression may be simplified by defining

$$s = \frac{x - x_0}{x_1 - x_0} \tag{6.17}$$

and rewrite the polynomial series of equation (6.15) as

$$U^l(x) = U[x_0] + s[U(x_1) - U(x_0)](1 + \frac{(s-1)}{2}\lambda_2 \times (1 + \frac{(s-e_2)}{3}\lambda_3 (1 + \frac{(s-e_3)}{4}\lambda_4 \times ....1 + \frac{(s-e_{N-1})}{N}(S^{[j,...,k]}_{[p,...,q]})).$$

(6.18)

or as

$$U^l(x) = U[x_0] + [U(x_1) - U(x_0)](s + s\frac{(s-1)}{2!}\bar{\lambda}_2 + \frac{s(s-1)(s-e_2)}{3!}\bar{\lambda}_3 + \frac{s(s-1)(s-e_2)(s-e_3)}{4!}\bar{\lambda}_4 + ... + \frac{s(s-1)(s-e_2)...(s-e_{N-1})}{N!}\bar{\lambda}_N)$$

(6.19)

where

$$\bar{\lambda}_2 = \lambda_2, \ \bar{\lambda}_3 = \lambda_2 \ \lambda_3, \ \bar{\lambda}_4 = \lambda_2 \ \lambda_3 \ \lambda_4 \quad \textit{and} \ \bar{\lambda}_N = \lambda_2 \ \lambda_3 ... \lambda_{N-1} \ S^{[j,...,k]}_{[p,...,q]}.$$

(6.20)

This form of the polynomial makes it straightforward to define the form of the additional error introduced by limiting as

$$\bar{\lambda}^{error}_{n-1} = \lambda_2 \ \lambda_3 ... \lambda_{n-1} \ \left[\lambda_n - S^{[j,...,k]}_{[p,...,q]}\right]. \tag{6.21}$$

and then to write the error introduced by limiting as the difference between the polynomials as

$$U^L(x) = U^l(x) + [U(x_1) - U(x_0)]\frac{s(s-1)(s-e_2)...(s-e_{n-1})}{n!}\bar{\lambda}^{error}_{n-1} \tag{6.22}$$

where $U^L(x)$ denotes the original form of the divided difference polynomial as defined by equation (2.5). The overall error is given by combining this error with the error due to the ENO form of the divided difference approximation as described in Section(5.2).

This form of the polynomial will be used to investigate the range of possible worst cases in Section 8.

7. Computational Algorithm and Experiments. In this section the computational algorithm employed is described as well as a number of experiments on Runge's function and a number of more or equally challenging examples. In the description of the algorithms the label Chb will refer to the interpolant using the extended Chebyshev points defined by equation (2.15). The label PCH will refer to the Pchip algorithm [14] as implemented in Matlab. The label Lim will refer to the new computational algorithm as given by by the four stage process:

1. Form divided differences of order used (up to 511).

2. Pick ENO Interpolants based on using the smallest divided difference possible.
3. For each interval ensure that the ratios are non-negative and if they are greater than one then limit the divided differences. Once the limiter is applied then the polynomial series is truncated.
4. Calculate the required values by evaluating the (usually) different polynomial order on each interval.

This new algorithm is used with evenly spaced points to illustrate its properties in this case. It is interesting to note that using the algorithm with a Chebyshev mesh spacing produced very similar, but in no case superior, results on the test problems described below. The L_2 and L_∞ norms are approximated by sampling at 2049 evenly spaced points, with the trapezoidal rule being used to estimate the L_2 norm as defined on the interval $[a,b]$ by:

$$||f||_2 = \sqrt{\frac{1}{b-a}\int_a^b f^2(x)dx}. \tag{7.1}$$

The L_∞ norm is defined as in equation (2.16).

7.1. Runge-type problems. The results are given for four test problems all defined on $[-1,1]$. For each case, and purely to assess the effect of reducing the mesh spacing by one half, polynomial interpolants using 15,31,63, 127, 255 and 511 points were created. These points were chosen solely to reflect the effect of doubling the number of points.

7.1.1. Problem A. This is the problem used by de Boor, [11], to illustrate poor convergence of polynomial interpolation. The function is defined by $U(x) = \sqrt{|x|}$. The computational results in Table 7.1 show better accuracy for the new evenly spaced interpolant than for the other two methods. Table 7.2 shows the order used varies between 1 (the minimum for all problems and cases) and the stated maximum of 52. In every case the new interpolant is data-bounded over each interval and also monotone.

7.1.2. Problem B. This is Runge's function as defined by $U(x) = 1/(1+25x^2)$. Table 7.1 shows that the best accuracy comes from the Chebyshev polynomial approximation. For low numbers of points (≤ 30) better accuracy is obtained by the new evenly spaced interpolant and PCHIP. The encouraging result is that the new method (despite using polynomials of degree only as high as 12) is still converging when as many as 511 points are used. In every case the new interpolant is data-bounded and also monotone over each interval.

7.1.3. Problem C. This is a modified form of Runge's Function suggested as a challenging example by Karniadakis and Kirby [22] as given by $U(x) = \varepsilon/(\varepsilon+25x^2)$, where $\varepsilon = 10^{-15}$. The computational results in Table 7.1 show better accuracy for the new evenly spaced interpolant than for the other two methods. Table 7.2 shows the order used varies between 1 (the minimum for all problems and cases) and the stated maximum of 63. In every case the new interpolant is data bounded over each interval and also monotone. The solution to this problem and the various approximations in the region of the spike is shown in Figure 7.1.

7.1.4. Problem D. This example is used by Gelb and Tanner [15] in the context of filtering type methods for Fourier series type approximations. The function has two distinct parts and is given by $U(x) = (2e^{(2\pi*(x+1))} - 1 - e^{(\pi)})/(e^{(\pi)} - 1)$ for $x < -0.5$ and for $x \geq -0.5$ it is given
by $U(x) = -sin((2\pi x)/3.0 + \pi/3)$. The computational results in Table 7.1 show similar levels of accuracy for the new evenly spaced interpolant and for PCHIP and that this acuracy is a little better than that of the Chebyshev method. Table 7.2 shows the order used varies

TABLE 7.1
Accuracy Results for Problems A, B, C and D

		Problem A		Problem B		Problem C		Problem D	
N	Meth	L2	L∞	L2	L∞	L2	L∞	L2	L∞
	Chb	1.5e-3	1.9e-1	5.0e-4	4.7e-2	7.1e-3	1.0e-0	2.4e-3	9.0e-1
15	Lim	2.7e-4	5.7e-2	3.9e-4	6.0e-2	3.4e-3	9.8e-1	2.3e-1	9.5e-1
	PCH	7.4e-4	1.4e-1	1.3e-4	1.7e-2	5.0e-3	1.0e-0	2.1e-3	8.9e-1
	Chb	7.5e-4	1.3e-1	2.3e-5	2.1e-3	4.9e-3	1.0e-0	2.1e-1	9.5e-1
31	Lim	1.0e-4	3.4e-2	8.3e-5	1.9e-2	2.1e-3	9.5e-1	1.4e-3	8.1e-1
	PCH	5.9e-4	9.6e-2	5.5e-5	5.4e-3	5.4e-3	1.0e-0	1.8e-3	6.4e-1
	Chb	3.7e-4	9.3e-2	3.9e-8	3.7e-6	3.4e-3	1.0e-0	1.1e-3	5.2e-1
63	Lim	4.0e-5	2.2e-2	1.8e-5	4.4e-3	1.3e-3	8.8e-1	9.2e-4	7.2e-1
	PCH	1.8e-4	6.6e-2	6.0e-6	1.8e-3	2.4e-3	9.8e-1	7.2e-4	5.5e-1
	Chb	1.8e-4	6.5e-2	1.2e-13	1.1e-11	3.4e-3	9.8e-1	1.1e-3	8.4e-1
127	Lim	1.7e-5	1.4e-2	3.2e-6	1.2e-6	7.6e-4	7.5e-1	6.8e-4	6.4e-1
	PCH	8.7e-5	4.6e-2	1.1e-6	4.6e-4	1.4e-3	9.3e-1	4.8e-4	4.6e-1
	Chb	9.2e-4	4.6e-2	7.5e-19	2.2e-16	1.7e-3	9.8e-1	5.3e-4	4.7e-1
255	Lim	8.9e-6	9.3e-3	8.6e-7	2.4e-4	4.3e-4	5.3e-1	4.0e-4	5.7e-1
	PCH	4.3e-5	3.2e-2	1.9e-7	1.1e-4	1.1e-3	7.6e-1	3.4e-4	4.7e-1
	Chb	4.6e-5	3.2e-2	8.5e-19	2.2e-16	1.1e-3	9.5e-1	4.7e-4	6.7e-1
511	Lim	2.9e-6	4.0e-3	3.0e-7	7.2e-5	2.0e-4	2.9e-1	2.5e-4	4.0e-1
	PCH	2.1e-5	1.2e-2	3.4e-8	2.8e-5	6.9e-4	3.0e-1	2.6e-4	4.9e-1

TABLE 7.2
Polynomial Points Used Per Interval For Problems A,B, C and D (minimum=2)

N	Order	Problem A	Problem B	Prob C	Problem D
15	Average	5	4	5	2
	Maximum	7	6	7	3
31	Average	9	5	9	3
	Maximum	15	9	15	7
63	Average	17	6	17	4
	Maximum	31	11	31	15
127	Average	15	6	20	4
	Maximum	53	13	63	14
255	Average	13	6	14	4
	Maximum	44	12	87	11
511	Average	10	6	10	4
	Maximum	53	10	66	10

between 1 (the minimum for all problems and cases) and the stated maximum of 14. In every case the new interpolant is data bounded over each interval and also monotone. The results for $N < 64$ are competitive against the Gegenbauer approach of Gelb and Tanner [15] but thereafter the Gegenbauer basis converges much more more rapidly, see Table 7.3. Part of the reason for this is that there is a steep front in the solution which looks like a shock wave. Figure 7.2 shows the limited even mesh polynomial and the original Chebyshev polynomial with $N = 31, 127i$ and 511, and the errors. As pointed out by the author, [6], the order used by the limited methods is constrained by the number of mesh points contained inside the steep front. For this reason the limited polynomial still only uses a cubic polynomial and so has a much larger error in that interval.

a: Command not found.

7.2. Monotone Interpolants. The method as described was tested on all the test problems described in the survey of Kocic and Milovanovic [23]. In all cases the new approach produced monotone approximations to the data. The values used for Example 2.4 of [23] are shown below in Table 7.3. The approximations produced by a standard polynomial approx-

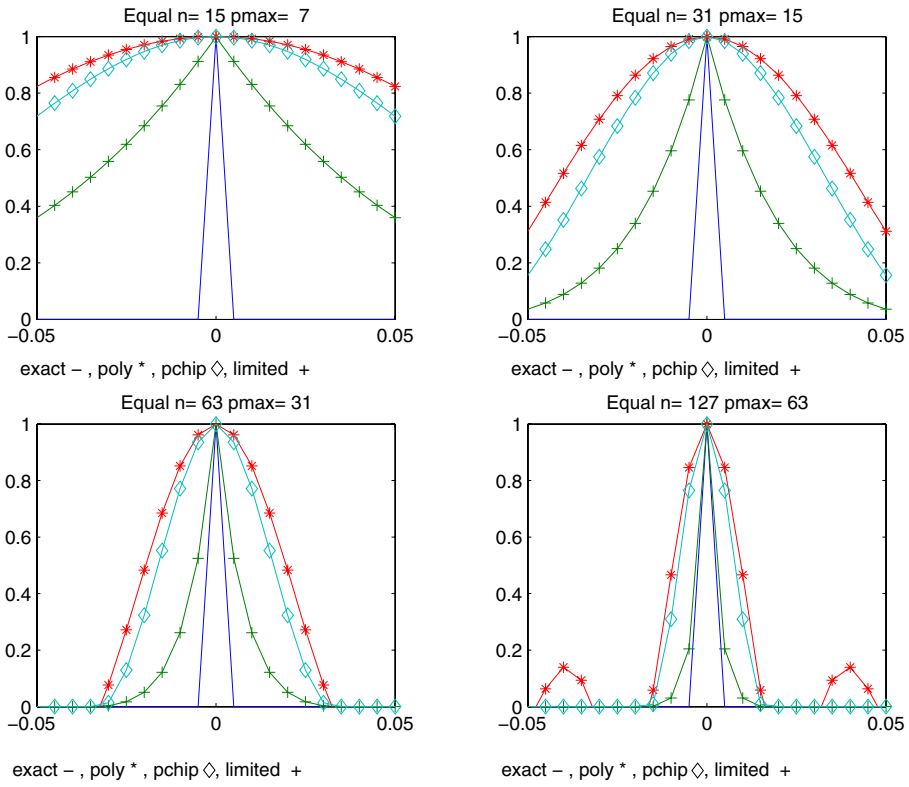

FIG. 7.1. *Even Mesh: Modified Runge's Function* $10^{-15}/(10^{-15}+25x^2)$

TABLE 7.3
Gelb and Tanner Results for Problem D

N	32	64	128	256	512
L_∞	6.0e-1	5.7e-1	1.3e-4	1.5e-6	2.2e-9

imation, by the new method and by the PCHIP method are shown in Figure 7.3. The figure shows both the oscillations of the standard polynomial approximation and also how close the new approach and PCHIP are visually.

8. Theoretical Properties. In this section the theoretical properties of the new limiting approach will be considered by proving three theorems that will demonstrate the data-bounded nature of the interpolant in two "worst case" scenarios. These scenarios are derived by noting that in the case of an evenly spaced mesh with mesh spacing h the values of the functions $\pi_{1,i}(x)$ defined in Section 2 are at their largest for interpolation at x in the interval $[x_i, x_{i+1}]$ when a stencil based upon points as far away from x as possible is used as then the absolute value of the multiplier $(x-x_j)$ is as large as is possible. This will be important in Theorem 2 as it is the straight forward to establish the worst case bounding sum for the magnitude of polynomials of a given degree when the ratios of divided differences are limited.

Theorem 1: The difference table corresponding to Harten's method as described in Sections 3 and 4 is composed of successive levels of divided differences all having the same sign at each level.

Proof: The highest order divided difference is, by definition, composed by subtracting two

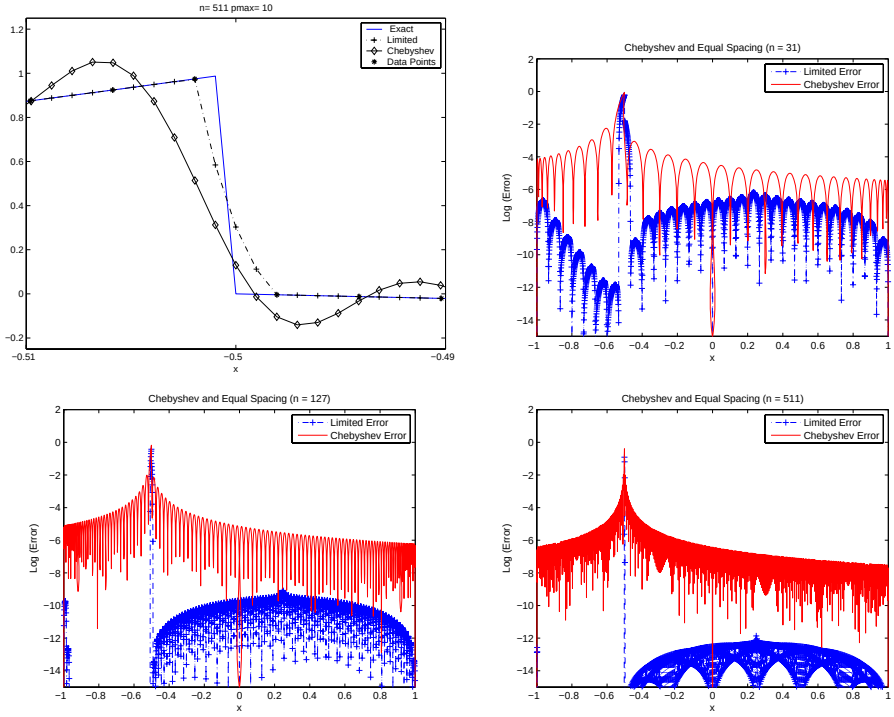

FIG. 7.2. *Problem D Detail of Approximation and Error Distribution*

TABLE 7.4
Values used for Example 2.4 Kocic and Milovanovic

x(i)	0	2	3	5	6	8	9	11	12	14	15
y(i)	10	10	10	10	10	10	10.5	15	50	60	85

differences each of which, as it was selected on the basis of the algorithm in Section 5, has the same sign. Similarly each of these differences is in turn composed by the subtraction of one difference from another. Again by construction using the algorithm in Section 5, each of these pairs of differences must have the same sign. Both pairs must also have the same sign as they have one difference in common. For example suppose

$$U[x_{i-1},x_i,\ldots,x_{i+2}] = \frac{U[x_i,x_{i+1},x_{i+2}] - U[x_{i-1},x_i,x_{i+1}]}{x_{i+2}-x_{i-1}}, \tag{8.1}$$

is calculated using values $U[x_i,x_{i+1},x_{i+2}]$ and $U[x_{i-1},x_i,x_{i+1}]$ which have the same sign. It then follows that as

$$U[x_i,x_{i+1},x_{i+2}] = \frac{U[x_{i+1},x_{i+2}] - U[x_i,x_{i+1}]}{x_{i+2}-x_i}, \tag{8.2}$$

the values $U[x_i,x_{i+1}]$ and $U[x_{i+1},x_{i+2}]$ have the same sign. Similarly as

$$U[x_{i-1},x_i,x_{i+1}] = \frac{U[x_i,x_{i+1}] - U[x_{i-1},x_i]}{x_{i+1}-x_{i-1}}, \tag{8.3}$$

the values $U[x_i,x_{i+1}]$ and $U[x_{i-1},x_i]$ have the same sign. From this it follows that all the values $U[x_{i-1},x_i]$, $U[x_i,x_{i+1}]$ and $U[x_{i+1},x_{i+2}]$ must have the same sign and so the values $U(x_{i-1})$,

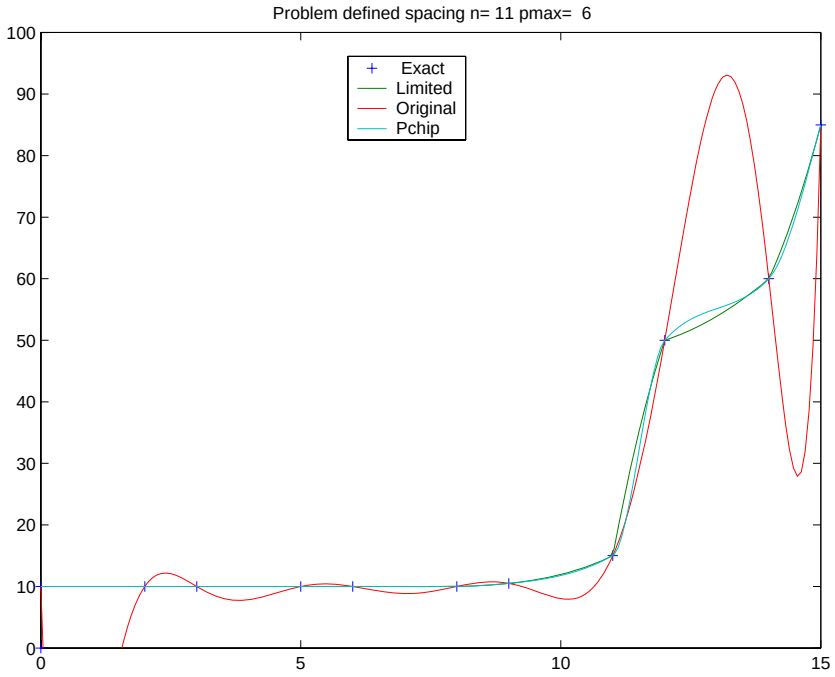

FIG. 7.3. *Monotone Interpolation Problem 2.4 of [23]*

$U(x_i)$,$U(x_{i+1})$, $U(x_{i+2})$ are either all non-increasing or non-decreasing. The argument extends recursively from higher differences down to lower ones. The polynomial interpolant is thus locally weakly k monotone in the sense of Kopotun [24].

Theorem 2: The interpolating function constructed on an evenly spaced mesh using steps (i), (ii), (iii) and (iv). in Section (7) is data bounded in that

$$Min(U(x_i),U(x_{i+1})) \leq U^l(x) \leq Max(U(x_i),U(x_{i+1})) \tag{8.4}$$

Proof: The approach taken is to consider the worst possible cases within the constraints of the algorithm. This means considering the divided difference series used to form $U^l(x)$ and identifying the bounding cases for the polynomials defined by the algorithm in Section (7). In the case of the evenly spaced mesh the polynomial is given by equation (6.18) and requirement for data boundedness is that

$$0 \leq S_N(s) \leq 1 \tag{8.5}$$

where

$$S_N(s) = s + s\frac{(s-1)}{2}\bar{\lambda}_2 + \frac{s(s-1)(s-e_2)}{3!}\bar{\lambda}_3 + \frac{s(s-1)(s-e_2)(s-e_3)}{4!}\bar{\lambda}_4 \tag{8.6}$$
$$\ldots + \frac{s(s-1)(s-e_2)\ldots(s-e_{N-1})}{N!}\bar{\lambda}_N$$

where s is defined by equation (6.16). In a worst case analysis it follows that the values of the terms in equation (8.6) must either all be positive and as large as possible, or every term

except the first must be negative and as large as possible. From equation (6.12) it follows that for an evenly spaced mesh that

$$e_i > 1 \text{ or } e_i < -1. \tag{8.7}$$

Assuming that the points selected by the ENO algorithm are as far away as possible from the interval on which interpolation is taking place. Hence the points used to define the polynomial are given by:

$$e_i = i, \text{ or } e_i = -(i-1),\ i > 1. \tag{8.8}$$

The justification for looking at these cases is that as

$$\begin{aligned} s(s-1)(s+1)...(s+(N-1)) \le s(s-1)(s-e_2)...(s-e_{N-1})\bar{\lambda}_{N-1} \text{ and} \\ s(s-1)(s-e_2)...(s-e_{N-1})\bar{\lambda}_{N-1} \le (-1)^{N-1}s(s-1)(s-2)...(s-(N-1)) \end{aligned} \tag{8.9}$$

then any terms in the series of the form of equation (8.6) are bounded above and below by the terms in the two series consisting of the terms in the previous equation. The multiplicative factor $S\begin{bmatrix}...\\...\end{bmatrix}$ does not change this situation thanks to equation (6.8). In the same vein the values of λ_k are ± 1 e.g. see equations (5.10) and (5.11). With regard to the two cases of interest mentioned above. In the first case the values of λ_k are chosen so that all the terms in equation (8.6) are positive and so that $\bar{\lambda}_k = (-1)^{k+1}$ Then the stencil is purely composed of values further right of the starting point and the limiting polynomial is given by

$$\begin{aligned} S_R^N(s) = s - \frac{s(s-1)}{2!} + \frac{s(s-1)(s-2)}{3!} - \frac{s(s-1)(s-2)(s-3)}{4!} + ... \\ +(-1)^{N+1}\frac{s(s-1)(s-2)(s-3)...(s-(N-1))}{N!} \end{aligned} \tag{8.10}$$

where $0 \le s \le 1$. The second worst case occurs when the stencil is purely composed of values all to the left of the initial interval and that $\lambda_k = 1$. The limiting polynomial is then defined by

$$\begin{aligned} S_L^N(s) = s + \frac{s(s-1)}{2!} + \frac{s(s-1)(s+1)}{3!} + \frac{s(s-1)(s+1)(s+2)}{4!} + ... \\ +\frac{s(s-1)(s+1)(s+2)...(s+(N-2))}{N!} \end{aligned} \tag{8.11}$$

where $0 \le s \le 1$. Let $\bar{s} = 1 - s$ then the above equation may be rewritten by noting that:

$$(s+j) = (-1)(\bar{s} - (j+1)) \tag{8.12}$$

and that

$$s(s-1) = \bar{s}(\bar{s}-1) \tag{8.13}$$

and then rewriting equation (8.11) in terms of $\bar{s}$ to get:

$$\begin{aligned} S_L^N(s) = 1 - \bar{s} + \frac{\bar{s}(\bar{s}-1)}{2!} - \frac{\bar{s}(\bar{s}-1)(\bar{s}-2)}{3!} + \frac{\bar{s}(\bar{s}-1)(\bar{s}-2)(\bar{s}-3)}{4!} - ... \\ +(-1)^{N+1}\frac{\bar{s}(\bar{s}-1)(\bar{s}-2)(\bar{s}-3)...(\bar{s}-(N-1))}{N!} \end{aligned} \tag{8.14}$$

and hence that

$$S_L^N(s) = 1 - S_R^N(\bar{s}). \tag{8.15}$$

The series form of $S_R^N(s)$ may be written as a truncated binomial expansion

$$S_R^N(s) = 1 - \sum_{k=0}^{N} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k. \tag{8.16}$$

It is worth mentioning that similar expansions are used by [29]. As all the terms in $S_R^N(s)$ are then positive it then follows that

$$S_R^N(s) \leq 1 - \sum_{k=0}^{\infty} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k. \tag{8.17}$$

From Raabe's test [12] the binomial expansion for zero given by

$$(1+(-1))^s = \sum_{k=0}^{\infty} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k \tag{8.18}$$

is absolutely convergent, see Sokolnikoff [27] pp.303-304, as

$$\lim_{k \to \infty} \left(k \left(\begin{array}{c} s \\ k \end{array} \right) (-1) \left(\begin{array}{c} s \\ k+1 \end{array} \right)^{-1} - 1 \right) = 1 + s, \;\; 0 \leq s \leq 1.$$

Hence, after treating the cases $s = 1$ and $s = 0$ separately it follows that

$$0 \leq S_L^N(s) \leq S_N(s) \leq S_R^N(s) \leq 1 \tag{8.19}$$

and the limited polynomial expansion is data bounded. The bounding series $S_R^N(s)$ and $S_L^N(s)$ are shown in Figure (8.1) for $N = 1024$. For larger values of N the graphs of $S_R^N(s)$ and $S_L^N(s)$ steepen and move even closer to $x = 0$ and $x = 1$ respectively.

Theorem 3: The interpolating function constructed on an evenly spaced mesh using steps (i), (ii), (iii) and (iv) in Section (7) is monotone in that

$$\frac{U^l(x)}{dx} = (U(x_{i+1}) - U(x_i)) \; f(x) \tag{8.20}$$

where $f(x) \geq 0$ for $x \in [x_i, x_{i+1}]$.

Proof: The approach taken is to consider the worst possible cases within the constraints of the algorithm. This means considering the divided difference series used to form $U^l(x)$ and identifying the bounding cases for the polynomials defined by the algorithm in Section (7). In the case of the evenly spaced mesh the polynomial is given by equation (6.18) and requirement for monotonicity is that

$$\frac{dS_N(s)}{ds} \geq 0 \tag{8.21}$$

where $S_N(x)$ is defined by equation (8.6) and so

$$\frac{dS_N(s)}{ds} = 1 + \frac{(2s-1)}{2}\bar{\lambda}_2 + \sum_{i=2}^{N} [s(s-1)p_i(s) + (2s-1)]) \; q_i(s) \frac{\bar{\lambda}_{i+1}}{i!} \tag{8.22}$$

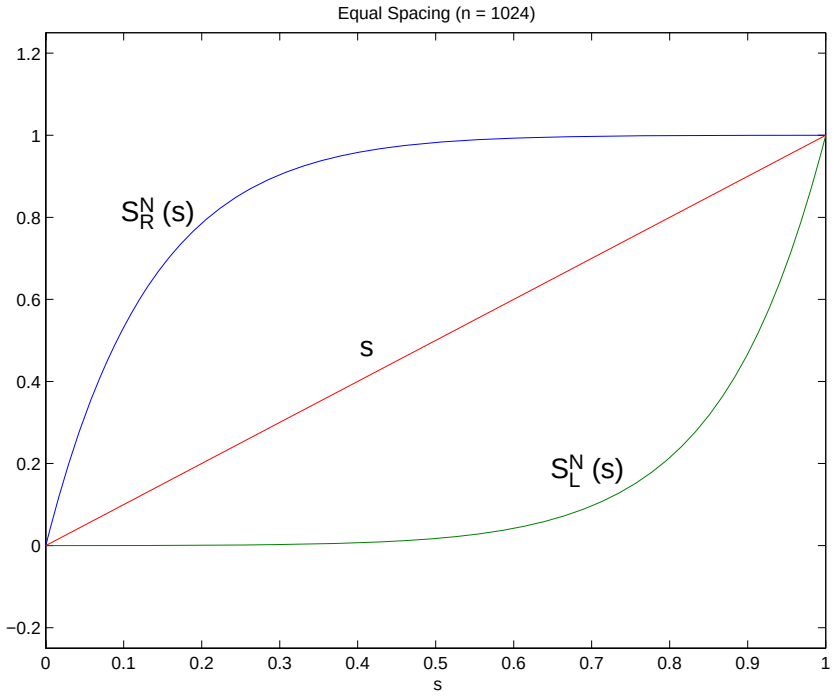

FIG. 8.1. *Upper and Lower Bounds for the Limited polynomials*

and where

$$p_i(s) = \sum_{j=2}^{i} \frac{1}{(s-e_j)} \text{ and } q_i(s) = \prod_{j=2}^{i} (s-e_j).$$

From a worst case analysis it follows that the most negative values of $\frac{dS_N(s)}{ds} - 1$ as defined by the righthand side of equation (8.22) are given by the case when the values $(s-e_i)$ are all positive and also that $p_i(s)$, $q_i(s)$ and λ_i are also positive and $\lambda_i = 1$. It immediately follows from equation (8.11) that the function that is defined in this way is

$$S_N(s) = S_L^N(s) \tag{8.23}$$

and from equation (8.15) that

$$S_N(s) = 1 - S_R^N(\bar{s}). \tag{8.24}$$

Furthermore for values of $0 \leq s \leq 0.5$ it follows that as all the terms in the summation on the right side of equation (8.22) are negative apart from the first term then

$$\frac{dS_N(s)}{ds} \geq 1 + \frac{(2s-1)}{2}\bar{\lambda}_2 + \sum_{i=2}^{\infty} [s(s-1)p_i(s) + (2s-1)])\, q_i(s)\frac{\bar{\lambda}_{i+1}}{i!}. \tag{8.25}$$

From equations (8.16) and (8.17) it follows that this equation may be written as

$$\frac{dS_N(s)}{ds} \geq \frac{d}{ds}(1+\alpha)^{1-s}, \alpha = -1. \tag{8.26}$$

The right side of this is non-negative as required.

In the case when $0.5 < s \leq 1$ then the most negative case is defined differently by $(\bar{\lambda}_i\ p_i(s)\ q_i(s)) > 0$. $(\bar{\lambda}_i\ q_i(s)) < 0$ and $\bar{\lambda}_1 < 0$. In this case from equation (8.10)

$$S_N(s) = S_R^N(s) \tag{8.27}$$

and

$$\frac{dS_N(s)}{ds} \geq 1 + \frac{(2s-1)}{2}\bar{\lambda}_2 + \sum_{i=2}^{\infty} [s(s-1)p_i(s) + (2s-1)])\ q_i(s)\frac{\bar{\lambda}_{i+1}}{i!}. \tag{8.28}$$

From equation (8.17) and (8.18) it follows that

$$\frac{dS_N(s)}{ds} \geq -\frac{d}{ds}(1+\alpha)^s, \alpha = -1. \tag{8.29}$$

The right side of this is non-negative as required.

9. Summary. In this paper a novel approach has been explored for preserving positivity for variable-order polynomial interpolation methods based on evenly spaced data points. The approach relies on limiting the divided differences used and employing an adaptive ENO stencil. There are clearly many possible extensions to this work such as the analysis of the Chebyshev cases and non-uniform meshes.

Acknowledgment The author would like to thank Mike Kirby and Frank Stenger for a number of interesting discussions on this topic, Nick Trefethen for helping to improve Section 2 and Ilse Ipsen for her help in clarifying the material.

REFERENCES

[1] Arandiga F. Chiavassa G. and Donat R. Harten's Framework for Multiresolution with Applications from Conservation Laws to Image Compression. GrAN Report 01-01 may 15th 2001. Departament de Matematica Aplicada , Universitat de Valencia, 4619900 Burfassot , Valencia SPAIN.

[2] Balsara D.S. Shi C.W. Monotonicity Preserving Weighted Essentially Non-Oscillatory schemes with Increasingly High Order of accuracy, *J. Comput. Phys.* 2000; **160** 404:452.

[3] Berrut J-P and Trefethen L.N. Barycentric Lagrange Interpolation *SIAM Review* **46**:501:517, 2004.

[4] Berzins M. A Data-Bounded Quadratic Interpolant on Triangular and Tetrahedral Meshes. *Siam Journal On Scientific Computing* **22** 1 pp 177-197, 1998.

[5] Berzins M. Variable-Order Finite Elements and Positivity Preservation for Hyperbolic PDEs *Applied Numerical Mathematics* **48**:271-292, 2004.

[6] Berzins M. Preserving Positivity for Hyperbolic PDEs Using Variable- Order Finite Elements with Bounded Polynomials. *Applied Numerical Mathematics* **52** 2-3 pp 197-217, February 2005.

[7] Berzins M. and Ware J.M. *Positive Cell Centered Finite Volume Discretization Methods for Hyperbolic Equations on Irregular Meshes. Applied Numerical Mathematics*, **16**, 417-438, 1995.

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

[9] Boyd J.P. Trouble with Gegenbauer reconstruction fro defeating Gibbs' phenomenon Runge phenomenon in the diagonal limit of Gegenbauer polynomial approximation *J. Comput. Phys.* 2005; **x** In press.

[10] Cockburn B, Karniadakis GE, Shu C-W. (eds). Discontinuous Galerkin Methods, Theory Computations and Applications. *L*ecture Notes in Computational Science and Engineering **1**1 Springer Berlin Heidelberg, 2000; 3–53.

[11] de Boor C., *A Practical Guide to Splines* Springer-Verlag Berlin Heidelberg, 1978;

[12] Bromwich, T. J. I'A. and MacRobert, T. M. An Introduction to the Theory of Infinite Series, 3rd ed. New York: Chelsea, p. 39, 1991'

[13] Epperson J.F., "On the Runge example," *American Math. Monthly*, **94**, 1987, pp.329-341.

[14] Fritsch, F.N. and R.E. Carlson, "Monotone Piecewise Cubic Interpolation," *Numerical Analysis*, **17**, 1980, pp.238-246.
[15] Gelb A. and Tanner J. Robust reprojection methods for the resolution of Gibbs phenomenon; submitted (October 2004) to Applied and Computational Harmonic Analysis.
[16] Gottlieb D, and Shu C-W. Resolution properties of the Fourier method for discontinuous waves, in *A*nalysis Algorithms and Applications of Spectral and High-Order Methods for P.D.E.s. (Selected Papers from (ICOSAHOM'92), Montpellier, France, June 22-26 1992) Eds. Bernardi C. and Maday Y. North Holland, Amsterdam, 1994.
[17] Gottlieb D, and Hesthaven J.S. Spectral Methods for Hyperbolic Problems. *Journal of Computational and Applied Mathematics* **128** (2001) 83-131.
[18] Harten A. Engquist B. Osher S. and Chakravarthy S.J. uniformly high order essentially non-oscillatory schemes, III, *J. Comput. Phys.* 1987; **71** 231.
[19] Harten A. Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws. *Comm. Pure and Applied Math.* 1995; **XIVIII**: 1305-1342.
[20] Hildebrand F.B. *Introduction to Numerical Analysis* McGraw-Hill Book Company Inc 1956 (reprinted by Dover Publications 1987)
[21] Jerri A. *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximations* Kluwer Academic Publishers, Dordecht, The Netherlands. 1998.
[22] Kirby M. and Karniadakis G. *Parallel Scientific Computing in C++ and MPI* Cambridge University Press, Cambridge UK, 2003
[23] Kocic L.M. and Milanovic G.V Shape Preserving Approximations by Polynomials and Splines *Comput. Math. Appl.* 1997; **33**: 59–79.
[24] Kopotun K. Approximation of k monotone functions. *J. Approx. Theory* 1998; **94**: 481–493.
[25] Kvasov B.I. *Shape Preserving Spline Approximation* World Scientific, Simgapore, New Jersey, London Hong Kong 2000;
[26] Revers M. A Survey on Lagrange Interpolation Based on Equally Spaced Nodes. In Advanced Problems in Constructive Approximation. (Eds) M.D. Buhmann and D.H.Mache. Birkhauser Verlag Basel (ISBN 3-7643-6648-6) *Int. Series of Numer. Math.* 2002; **142**: 153–163.
[27] Sokolnikoff I.S. *Advanced Calculus* McGraw Hill, London New York 1939.
[28] Tadmor E. and Tanner J. Adaptive Mollifiers - High Resolution Recover of Piecewise Smooth Data from its Spectral Information; *Foundations of Computational Mathematics* **2** (2) (2002) 155-189.
[29] Trefethen L.N. and Weideman J.A.C. Two results on Polynomial Interpolation in Equally Spaced Points *Journal of Approximation Theory* **65** (1991) 247–260.
[30] Trefethen L.N. *Spectral Methods in Matlab* Siam Philadelphia 2000.