An implicit WENO scheme for steady-state computation of scalar hyperbolic equations

Sigal Gottlieb
Mathematics Department
University of Massachusetts at Dartmouth
285 Old Westport Road
North Dartmouth, Massachusetts 02747-2300
E-Mail: sgottlieb@umassd.edu

and

Julia S. Mullen
Computing and Communications Center
Worcester Polytecnic Institute
100 Institute Road
Worcester, MA 01609
jsm@wpi.edu

Abstract: Weighted essentially non-oscillatory (WENO) schemes have proved useful in a variety of physical applications. They capture sharp gradients without smearing, and feature high order of accuracy along with nonlinear stability. The high order of accuracy, robustness, and smooth numerical fluxes of the WENO schemes make them ideal for use with Jacobian based iterative solvers, to directly simulate the steady state solution of conservation laws. In this paper, we consider a Newton based implicit WENO solver for scalar conservation laws. A unique interpolation technique is developed, which produces a more efficient iteration. Numerical results are presented.

Keywords: Weighted ENO, Steady-state, Newton iteration, Jacobian based solvers.

1 Introduction

Weighted essentially non-oscillatory (WENO) schemes [7] are high order finite-difference methods with an adaptive-stencil approach. At each point in space, they approximate the derivative with a high order difference formula selected to prevent oscillations. WENO methods capture sharp gradients without smearing, and do not allow oscillations to appear and thus preserve the correct physical behavior by upwinding and stencil choosing. WENO schemes have proven useful in resolving the numerical solution of conservation laws with shocks [8], [7], and are useful whenever sharp gradients are present, to prevent non-physical oscillations from appearing, propagating, and ultimately destroying the reliability of the numerical method. WENO schemes are distinguished by high order of accuracy, robustness, and smooth numerical fluxes. These features make it possible to use WENO in conjunction with Jacobian based iterative solvers to efficiently solve steady state problems directly.

2 The WENO Method in the Context of Earlier Methods

To approximate, in a physically correct way, [3] the solution to a conservation law of the form

$$u_t + f(u)_x = 0,$$

we use a conservative finite difference scheme

$$u_t = -\frac{1}{\Delta x}(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}).$$

The term $\hat{f}_{j+\frac{1}{2}} = \hat{f}(u_{j-k}, ..., u_{j+l})$ is the *numerical flux*, and the points $u_{j-k}, ..., u_{j+l}$ constitute the *stencil.* To be a reasonable approximation, the numerical flux must be (at least) Lipschitz continuous and consistent with the physical flux f, *i.e.* $\hat{f}(u, ..., u) = f(u)$. The numerical flux determines the numerical method and its properties. Any differences between conservative numerical methods are a result of differences in the numerical flux.

The numerical flux determines the order of accuracy of the method, as well as the stability properties. On the left of the shock and the right of the shock we have smooth regions, and in those regions linear stability is enough to ensure nonlinear stability. Instability occurs when points on opposite sides of the shock are used to evaluate the derivative at a given point. This causes oscillations at the shock location, which propagate to the smooth regions, destroying the stability of the solution. The idea behind essentially non-oscillatory (ENO) schemes is stencil switching in order to eliminate oscillations [5], [6]. ENO schemes search for the locally smoothest stencil and use that stencil to calculate the numerical fluxes. This assures that the shock is not crossed.

Liu, Osher and Chan [7] improved upon the ENO schemes by using all candidate stencils instead of using only the smoothest candidate stencil. Each stencil is assigned a weight, which depends on its smoothness. Then all approximations from all the candidate stencils are added up, each with the weight assigned to it.The weights are chosen so that in smooth regions we obtain higher order accuracy whereas near discontinuities the ENO scheme is imitated by assigning near-zero weights to the stencils that contain discontinuities. To get an rth order ENO scheme, a total of $2r-1$ points are examined. Since the WENO scheme uses all the candidate stencils, a clever choice of weights [8] results in a WENO scheme which is of order $2r-1$ in smooth regions.

WENO schemes take each candidate stencil

$$S_k = (f^+(u_{j-r+1+k}), .., f^+(u_{j+k}))$$

and assign it a smoothness measurement,

$$IS_k = \sum_{l=1}^{r-1} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \Delta x^{2l-1} (q_k^{(l)}(x))^2 dx,$$

where $q_k^{(l)}(x)$ are the lth derivatives of the interpolation polynomial associated with the stencil S_k. For each stencil, a weight ω_k (based on the smoothness measurement) is assigned. These weights sum to one and approach the ENO weights in nonsmooth regions. The numerical flux is then defined as:

$$\hat{f}_{j+\frac{1}{2}} = \sum_{k=0}^{r-1} \omega_k q_k^r(S_k)$$

and the resulting method is of order $2r-1$ in smooth regions, and rth order near the shock.

3 Convergence to Steady State of WENO using a Newton iteration

A classical problem with ENO schemes, which is due to the stencil switching in ENO, is that their numerical flux is only Lipschitz continuous but not smoother. This lack of smoothness causes problems in steady state calculations, in that the residual never settles down to machine zero but hangs at the truncation error level, usually within 10^{-3} to 10^{-5}. WENO schemes, on the other hand, can settle down to machine zero residual while still maintaining a sharp, non-oscillatory steady state shock resolution together with uniformly high order accuracy in the smooth part of the solution. The smoothness of the numerical flux of WENO also facilitates the usage of Jacobian based iterative solvers for steady states, such as Newton and damped Newton methods.

The time dependent problem

$$u_t + f(u)_x = g(u, x)$$

becomes, at steady state

$$f(u)_x = g(u, x) \tag{1}$$

This equation can be solved directly using an iterative solver such as Newton's method for nonlinear systems, rather than solving the time dependent problem until steady state is reached. This method will converge if the starting values are sufficiently close to the real solution.

Since the $f(u)_x$ term is approximated using Weighted ENO, for which the numerical fluxes are smooth, there is no trouble in calculating the Jacobian matrix needed for the Newton's iteration. The Jacobian matrix is a sparse matrix, and this can be used to reduce the cost of the iteration. The choice of initial guess is the key to quick convergence. In the next section, we show how well this method works, and we explore a interpolation technique for generating a good initial guess.

4 Numerical Examples

Example 1: To validate this method we choose a test problem which features a steady shock as the viscosity vanishes. The boundary value steady state problem:

$$\delta u'' - (\frac{1}{2}u^2)' - u = 0 \qquad 0 \leq x \leq 1$$

$$u(0) = \frac{3}{4} \qquad u(1) = -\frac{1}{2}.$$

has exact solution is:

$$u(x) = \begin{cases} 0.75 - x & \text{if } 0 \leq x < 0.625 \\ -x + 0.5 & \text{if } 0.625 < x \leq 1 \end{cases}$$

The computations are performed with $\delta = 10^{-6}$, and initial guess $u(x) = \frac{3}{4} - \frac{5}{4}x$. The direct problem was computed using a Newton iteration for which the convergence criteria was 10^{-15} (machine precision). For comparison, the associated time-dependent problem was run to steady state explicitly with a Runge-Kutta timestepping method. For each simulation (Newton and RK) two sets of numerical experiments were carried out. In the first set of experiments, the method is applied in a straightforward manner, using a linear initial guess, to the full mesh. In the second set of experiments, the approximation to the solution is computed in the straightforward way with linear initial guess on a coarse mesh of 20 gridpoints. This numerical solution is then interpolated onto 40 gridpoints, and used as the initial guess for the numerical solution at 40 gridpoints. This process is repeated, to produce an initial guess for the numerical simulation on the full mesh. As seen in Table 1, the interpolation technique results in an overall more efficient scheme when compared to the numerical simulation on the full mesh starting from standard initial data. Note that the CPU time reported for the interpolated experiments includes not merely the solution on the fine grid from the interpolated initial guess, but the entire interpolation cascade and solution on the fine mesh. Furthermore, it reflects the fact that a full matrix solver was used, for inverting the Jacobian in the Newton iteration, instead of an efficient sparse solver.

Example 2: The second test problem is the Embid-Majda problem [4] which has two solutions, one with a standing shock which is stable in time, and one with an unstable standing shock. The WENO method accurately captures the shock behavior, but the solver strategy has a major impact on the solution. The time-stepping RK method converges only to the stable shock. When the Newton iteration is used to solve the steady-state problem directly, the shock captured depends on the initial condition. *e.g.* when the initial guess is a step function with shock location close enough to the stable shock location $x_1 = 0.18$, the iteration converges to the stable shock location. However, when the initial guess is a step functions with shock location close enough to the unstable shock location $x_2 = 0.82$, the iteration converges to the unstable shock location. The Embid–Majda problem is:

$$(\frac{1}{2}u^2)_x = a(x)u \qquad 0 \le x \le 1$$

$$a(x) = 6x - 3.$$

$$u(0) = 1 \qquad u(1) = -0.1$$

Which has the steady exact solution:

$$u(x) = \begin{cases} 3x^2 - 3x + 1 & \text{if } 0 \le x < 0.18 \\ 3x^2 - 3x - 0.1 & \text{if } 0.18 < x \le 1 \end{cases}$$

Once again, a Newton iteration and Runge Kutta time-stepping method are used, with the following cases as initial conditions:

1.

$$u(x) = \begin{cases} 1 & \text{if } 0 \le x < 0.18 \\ -0.1 & \text{if } 0.18 < x \le 1 \end{cases}$$

2.

$$u(x) = \begin{cases} 1 & \text{if } 0 \le x < 0.38 \\ -0.1 & \text{if } 0.38 < x \le 1 \end{cases}$$

In general, the time-stepping method took a shorter time to converge, even though more iterations were needed (Table 2). As with the first example we conjecture that this is due in part to the inefficient matrix solver used in the Newton direct method. Similar to the results presented in Example 1, we observe that the interpolation technique is more efficient for the Newton iteration, but not for the RK timestepping.

5 Conclusions

There are many problems for which the physical phenomena exhibit sharp gradients and thus require stable high order methods that insure that there are no spurious oscillations. It is clear that WENO is ideally suited to such problems. For steady state simulations, we would like to avoid the use of time-stepping, and solve the problem directly. This approach implies an implicit WENO. We present an implicit WENO which employs a Jacobian-based Newton iterative solver. Furthermore, the interpolation procedure presented here – a cascade of approximate solutions on finer and finer grids serving as the initial guess of the next finest grid – clearly increases the speed of convergence. Based on these results it is expected that implicit WENO with the interpolation procedures should be considered for larger problems up to an including systems.

Acknowledgments
This research was supported by NSF grant DMS-0106743. The authors wish to thank Chi-Wang Shu for his very valuable suggestions.

References

[1] Shu C.-W. *TVB Uniformly High Order Schemes for Conservation Laws.* Math. Comp. 1987; 49:105-121.

[2] Shu C.-W. *Total-Variation-Diminishing Time Discretizations.* SIAM J. Sci. Stat. Comput. 1988; 9:1073-1084.

[3] LeVeque R.J. *Numerical Methods for Conservation Laws*, (Lectures in Mathematics). Basel: Birkhauser, 1992.

[4] Embid P., Goodman J., Majda A. *Multiple Steady States for 1-D Transonic Flow.* SIAM J. Sci. Comput. 1984; 5:21-41.

[5] Harten A., Engquist B., Osher S., Chakravarthy S. *Uniformly High Order Essentially Non-Oscillatory Schemes I.* SIAM Journal on Numerical Analysis 1987; 24:279-309.

[6] Harten A., Engquist B., Osher S., Chakravarthy S. *Uniformly High Order Essentially Non-Oscillatory Schemes III.* Journal of Computational Physics 1987; 71:231-303.

[7] Liu X-D., Osher S., Chan T. *Weighted Essentially Non-Oscillatory Schemes.* Journal of Computational Physics 1994; 115(1):200.

[8] Jiang G.-S. and Shu C.-W. *Efficient Implementation of Weighted ENO Schemes.* J. of Comp. Physics 1996; 126:202-228.

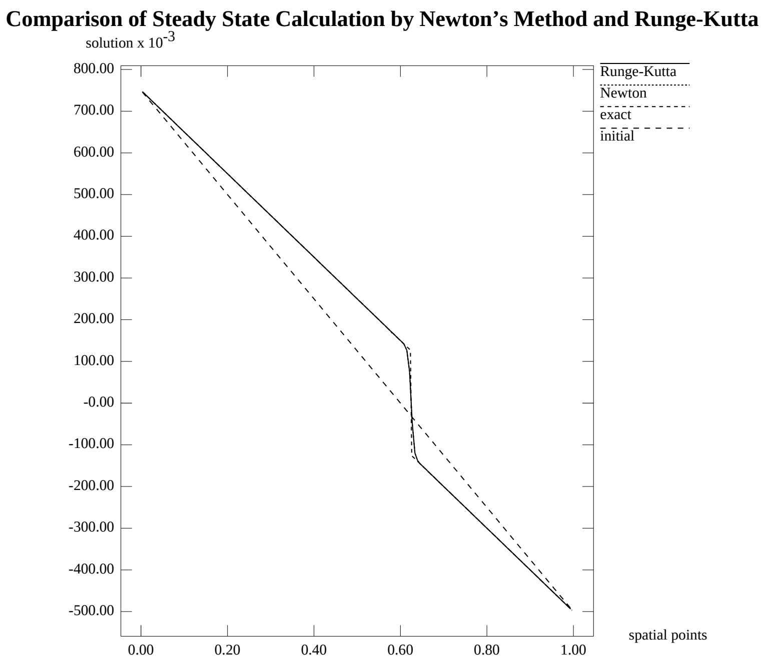
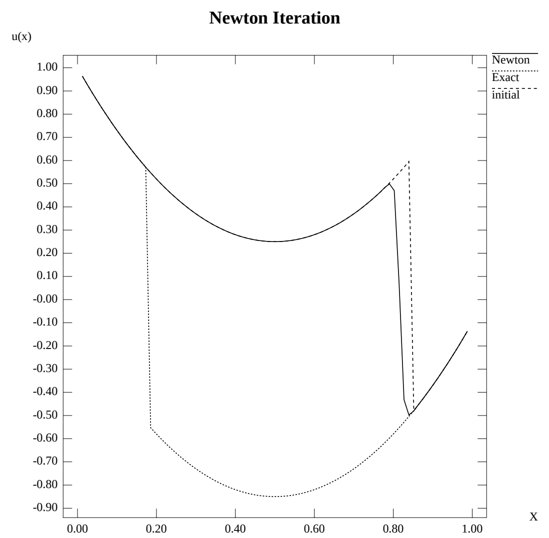


Figure 1: Convergence to steady state of example 1. A comparison of the direct solution using Newton's method and the explicit Runge-Kutta timestepping.



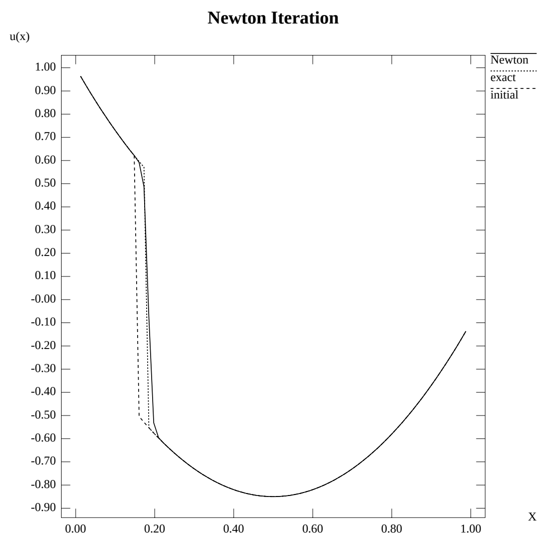


Figure 2: Newton's method applied to Embid-Majda. Left: with initial guess shock location $x_{shock} = 38$, convergence to the unstable shock location. Right: Convergence to the stable shock solution

Table 1: Comparison of Newton's method and Runge-Kutta time stepping

n	Runge-Kutta		Newton		Interp. RK		Interp. Newton	
	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter
160	13.0	2851	113.0	73	29.0	8943	16.0	35
320	249.0	29152	3029.0	277	108.0	17368	113.0	47

Table 2: Comparison of the number of iterations used by Newton and Runge-Kutta, and the CPU time used for the Embid-Majda problem.

n	Runge-Kutta		Interp. RK		Newton		Interp. Newton		i.c.
	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	
160	24.0	5858	30.0	9774	27.0	18	26.0	54	1
320	98.0	11869	113.0	19903	650.0	60	155.0	70	1
40	2.0	1481	2.0	1287	31.0	81	—	—	2
80	6.0	2969	8.0	2549	246.0	919	20.0	342	2