An efficient algorithm for computing smoothness indicators for WENO schemes

U S Vevek*

School of Mechanical and Aerospace Engineering, Nanyang Technological University,

50 Nanyang Avenue, Singapore 639798.

Currently at Department of Mechanical, Materials and Aerospace Engineering,

University of Liverpool, Liverpool L69 3BX, United Kingdom.

Abstract

WENO schemes are a popular class of shock-capturing schemes which adopt an adaptive-stencil approach to interpolation. WENO schemes rely on smoothness indicators to assess the relative smoothness of the solution within the sub-stencils. Computing these smoothness indicators is the most expensive operation in the WENO reconstruction procedure. In this paper, an efficient algorithm is proposed to compute these quantities without sacrificing the positivity property of the smoothness indicators. The proposed algorithm involves linear combinations of the undivided differences which can be computed efficiently in a recursive manner. This allows the computation of the smoothness indicators to be performed using significantly fewer floating-point operations compared to conventional implementations. Moreover, the proposed algorithm is simple to implement and involves fewer constants.

*Corresponding author: m150086@e.ntu.edu.sg

1 Introduction

Hyperbolic conservation laws admit discontinuous solutions. Solving such problems numerically requires nonlinear reconstruction schemes, known as shock-capturing schemes, which introduce numerical dissipation to prevent the formation of spurious oscillations near sharp gradients. Weighted essentially non-oscillatory (WENO) schemes are one such class of high order shock-capturing schemes. Introduced by Liu, *et al.* [1], WENO schemes adopt an adaptive-stencil approach (as opposed to the fixed-stencil approach of linear schemes) by combining the reconstructions from several sub-stencils (subsets of a stencil) using weights which are determined based on the relative smoothness of the solution within each sub-stencil. This allows WENO schemes to eliminate contributions from sub-stencils containing discontinuities and, at the same time, to achieve the optimal order of accuracy on the stencil when the solution is smooth in all the sub-stencils.

Key to achieving the optimal order of accuracy is the design of the sub-stencil smoothness indicator. The smoothness indicator proposed by Liu, *et al.* [1], though simple to compute, does not satisfy the requirement for optimal accuracy. Jiang and Shu [2] proposed a different smoothness indicator based on the cell average of the squares of derivatives which satisfy the requirement for optimal accuracy in regions away from critical points [3]. The Jiang-Shu smoothness indicators have since become the most popular choice of smoothness indicators. However, they are more expensive to compute compared to the Liu-Osher-Chan smoothness indicators, especially for high order WENO schemes. Since computing these quantities is the most demanding step in the WENO reconstruction procedure, overall computational efficiency could be vastly improved by speeding up this step. While much effort has been devoted to devise new smoothness indicators for better accuracy (e.g., [4-7]), the computational efficiency of such indicators has not received the nearly same attention. Notable exceptions include the works of Teng, *et al.* [8] and Baeza, *et al.* [9]. Teng, *et al.* [8] avoided WENO reconstructions altogether where the solution is deemed nearly uniform based on first order undivided differences. Baeza, *et al.* [9] introduced a set of efficient smoothness indicators for a $(2r-1)\mathrm{th}$ order WENO scheme using squared undivided differences of only the first and $(2r-2)\mathrm{th}$ orders.

Instead of introducing new smoothness indicators, this paper proposes a more efficient algorithm for computing the Jiang-Shu smoothness indicators, the most common choice in the community. The algorithm mimics the form of the Liu-Osher-Chan smoothness indicators in that it uses squared undivided differences up to $(r-1)\mathrm{th}$ order. The proposed algorithm is simpler-to-implement and requires fewer floating-point operations compared to conventional implementations. The paper is organized as follows: First, a review of the WENO reconstruction procedure is described along with a discussion on two common implementations of computing the Jiang-Shu smoothness indicators. Finally, the fast algorithm is derived and a comparison of the computational efficiency of the proposed algorithm is provided by counting the number of floating-point operations.

2 Methodology

2.1 Review of WENO scheme

In the finite volume approach, the computational domain is discretized into non-overlapping control volumes (cells) and the solution is obtained in terms of cell averages. On a uniform grid, the $i\mathrm{th}$ cell average $\bar{u}_i$ of a scalar function $u(x)$ is defined as,

$$\bar{u}_i = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x)\ dx\,, \qquad i = 0, \ldots, N-1 \tag{1}$$

where Δx denotes the cell width, x_i the cell centres and $x_{i\pm1/2} = x_i \pm \Delta x/2$ the cell interface locations. Without loss of generality, let us consider the left-biased reconstruction of the cell averages at the cell interface $x_{i+1/2}$. The right-biased reconstruction can be derived by symmetry. For a $(2r-1)\mathrm{th}$ order WENO scheme, the left-biased reconstruction $u^L_{i+1/2}$ is computed on the stencil $\mathcal{S}^L_{i+1/2} = \{i-r+1, \ldots, i+r-1\}$. $\mathcal{S}^L_{i+1/2}$ is split into r overlapping sub-stencils $\mathcal{s}^L_{j,i+1/2} = \{i-r+1+j, \ldots, i+j\}$ each consisting of r cells. The sub-stencil index j runs from 0 to $r-1$.

Each sub-stencil $\mathcal{s}^L_{j,i+1/2}$ yields an $r\mathrm{th}$ order approximation $u^L_{j,i+1/2}$ which can be computed as,

$$u_{j,i+1/2}^{L} = \sum_{k=0}^{r-1} \mathrm{M}_{jk}^{(r)} \bar{u}_{i-r+1+j+k} \tag{2}$$

using a reconstruction matrix $\mathrm{M}^{(r)}$. The WENO reconstruction $u_{i+1/2}^{L}$ is computed as a convex combination of $u_{j,i+1/2}^{L}$ given below.

$$u_{i+1/2}^{L} = \sum_{j=0}^{r-1} \omega_j u_{j,i+1/2}^{L} \tag{3}$$

The nonlinear sub-stencil weights ω_j in Eq. (3) are computed using the ideal weights $d_j^{(r)}$ smoothness indicators $IS_j^{(r)}$ are as follows.

$$\omega_j = \frac{d_j^{(r)} / \left(IS_j^{(r)} + \epsilon\right)^2}{\sum_{j=0}^{r-1} d_j^{(r)} / \left(IS_j^{(r)} + \epsilon\right)^2} \tag{4}$$

The smoothness indicator proposed by Jiang and Shu [2] is given below.

$$IS_j^{(r)} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \sum_{l=1}^{r-1} \left(\frac{d^l \mathcal{U}_{j,i+1/2}^{L}(x)}{dx^l} \Delta x^l \right)^2 \, dx \tag{5}$$

$\mathcal{U}_{j,i+1/2}^{L}(x)$ in the above definition refers to the $r\mathrm{th}$ order polynomial reconstructed on sub-stencil $\mathscr{s}_{j,i+1/2}^{L}$. Computation of $IS_j^{(r)}$ is usually implemented using the cell averages belonging to sub-stencil j in the following form,

$$IS_j^{(r)} = \sum_{m=1}^{r-1} Q_m \left(\sum_{k=0}^{r-1} P_{j,mk}^{(r)} \bar{u}_{i-r+1+j+k} \right)^2 \tag{6}$$

where Q_m and $P_{j,mk}^{(r)}$ are constants. This is referred to as the compact implementation because when the polynomial $\mathcal{U}_{j,i+1/2}^{L}(x)$ in Eq. (5) is expressed in the basis of Hermite polynomials, the expressions for $IS_j^{(r)}$ reduce to a compact form in terms of the modal coefficients [10-12]. Since $IS_j^{(r)}$ is computed as a sum of squares, the compact implementation ensures its positivity, and this property has been demonstrated to improve accuracy [13]. Despite its name, the compact implementation is computationally expensive. As the order of the WENO scheme increases, the number of sub-stencils increases and, the expressions for each $IS_j^{(r)}$ become longer and more unwieldy. It should come as no surprise that computation of smoothness indicators is the most demanding operation in the WENO reconstruction procedure. Therefore, it would be of tremendous advantage to devise a faster algorithm to compute them without sacrificing their positivity.

2.2 Proposed algorithm for computing $IS_j^{(r)}$

To motivate the design of a more efficient algorithm, consider the compact form of $IS_0^{(4)}$.

$$\begin{aligned} IS_0^{(4)} = \left(\tfrac{1}{3}\bar{u}_{i-3} - \tfrac{3}{2}\bar{u}_{i-2} + 3\bar{u}_{i-1} - \tfrac{11}{6}\bar{u}_i\right)^2 + \tfrac{13}{3}\left(\tfrac{1}{2}\bar{u}_{i-3} - 2\bar{u}_{i-2} + \tfrac{5}{2}\bar{u}_{i-1} - \bar{u}_i\right)^2 \\ + \tfrac{781}{20}\left(\tfrac{1}{6}\bar{u}_{i-3} - \tfrac{1}{2}\bar{u}_{i-2} + \tfrac{1}{2}\bar{u}_{i-1} - \tfrac{1}{6}\bar{u}_i\right)^2 \end{aligned} \tag{7}$$

Eq. (7) can be re-arranged into a slightly different form below.

$$\begin{aligned} IS_0^{(4)} &= \left[-\tfrac{1}{3}(\bar{u}_{i-2} - \bar{u}_{i-3}) + \tfrac{7}{6}(\bar{u}_{i-1} - \bar{u}_{i-2}) - \tfrac{11}{6}(\bar{u}_i - \bar{u}_{i-1})\right]^2 \\ &+ \tfrac{13}{12}[(\bar{u}_{i-1} - 2\bar{u}_{i-2} + \bar{u}_{i-3}) - 2(\bar{u}_i - 2\bar{u}_{i-1} + \bar{u}_{i-2})]^2 \\ &+ \tfrac{781}{720}[-(\bar{u}_i - 3\bar{u}_{i-1} + 3\bar{u}_{i-2} - \bar{u}_{i-3})]^2 \end{aligned} \tag{8}$$

Observe that the terms inside the first, second and third pair of square brackets are linear combinations of the first, second and third order undivided differences, respectively, computed on the four cells which belong to sub-stencil $j = 0$. Indeed, it is possible to cast $IS_j^{(r)}$ into the general form,

$$IS_j^{(r)} = \sum_{m=1}^{r-1} \frac{Q_m}{(m!)^2} \left[\sum_{k=0}^{r-1-m} A_{j,mk}^{(r)} \Delta^m [\bar{u}_{i-r+1+j+k}]\right]^2 \tag{9}$$

where $\Delta^0[\bar{u}_i] = \bar{u}_i$ and $\Delta^{m+1}[\bar{u}_i] = \Delta^m[\bar{u}_{i+1}] - \Delta^m[\bar{u}_i]$ denote the undivided differences. Q_m are the same constants which appear in Eq. (6) for the compact implementation. Eqs. (6) and (9) result in identical expressions when expended in terms of the cell averages. The main advantage of using Eq. (9) is that it is written in terms of the undivided differences which can be computed efficiently in a recursive fashion, i.e., the first order differences can be computed from the zeroth order differences, the second order differences from the first, and so on.

The constants Q_m and $A^{(r)}_{j,mk}$ can be determined in a straightforward manner. The derivation of Q_m and $A^{(4)}_{0,mk}$ used in Eq. (8) for computing $IS^{(4)}_0$ will be presented next. For $r = 4$, a cubic polynomial is reconstructed from each sub-stencil. Let the cubic polynomial be expressed in terms of the Taylor series coefficients about the cell centre x_i as shown below.

$$\mathcal{U}(x) = \mathcal{U}_i + \frac{d\mathcal{U}}{dx}(x - x_i) + \frac{1}{2}\frac{d^2\mathcal{U}}{dx^2}(x - x_i)^2 + \frac{1}{6}\frac{d^3\mathcal{U}}{dx^3}(x - x_i)^3 \tag{10}$$

$\mathcal{U}_i = \mathcal{U}(x_i)$ refers to the point value at x_i and the derivatives are also evaluated at x_i. Substituting Eq. (10) into Eq. (5) and simplifying yields the general expression for $IS^{(4)}_j$.

$$IS^{(4)}_j = 1 \cdot \left[\frac{d\mathcal{U}}{dx}\Delta x + \frac{1}{24}\frac{d^3\mathcal{U}}{dx^3}\Delta x^3\right]^2 + \frac{13}{12} \cdot \left[\frac{d^2\mathcal{U}}{dx^2}\Delta x^2\right]^2 + \frac{781}{720} \cdot \left[\frac{d^3\mathcal{U}}{dx^3}\Delta x^3\right]^2 \tag{11}$$

The constants which pre-multiply each pair of square brackets in Eq. (11) are $Q_m/(m!)^2$. These constants factor out when the smoothness indicator is written as a sum of squares under the condition that the coefficient of the leading order term inside each pair of square brackets be unity. It must be remarked that these constants are the same for all combinations of r and j. They are listed in Table 3 in the Appendix for $m = 1$ to $m = 5$.

For $j = 0$, the sub-stencil $\mathscr{s}^L_{0,i+1/2}$ consists of cells $\{i-3, i-2, i-1, i\}$. Averaging Eq. (10) over each of these four cells yields the expression for the respective cell average in terms of the Taylor series terms which can be cast into a linear system as follows.

$$\begin{pmatrix} \bar{u}_{i-3} \\ \bar{u}_{i-2} \\ \bar{u}_{i-1} \\ \bar{u}_i \end{pmatrix} = \begin{bmatrix} 1 & -3 & \frac{109}{24} & -\frac{37}{8} \\ 1 & -2 & \frac{49}{24} & -\frac{17}{12} \\ 1 & -1 & \frac{13}{24} & -\frac{5}{24} \\ 1 & 0 & \frac{1}{24} & 0 \end{bmatrix} \begin{pmatrix} \mathcal{U}_i \\ \frac{d\mathcal{U}}{dx}\Delta x \\ \frac{d^2\mathcal{U}}{dx^2}\Delta x^2 \\ \frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \end{pmatrix} \tag{12}$$

The first order undivided differences $(m = 1)$ can be computed from Eq. (12) as follows.

$$\begin{pmatrix} \Delta^1[\bar{u}_{i-3}] \\ \Delta^1[\bar{u}_{i-2}] \\ \Delta^1[\bar{u}_{i-1}] \end{pmatrix} = \begin{bmatrix} -1 & 1 & & \\ & -1 & 1 & \\ & & -1 & 1 \end{bmatrix} \begin{pmatrix} \bar{u}_{i-3} \\ \bar{u}_{i-2} \\ \bar{u}_{i-1} \\ \bar{u}_i \end{pmatrix} = \begin{bmatrix} 1 & -\frac{5}{2} & \frac{77}{24} \\ 1 & -\frac{3}{2} & \frac{29}{24} \\ 1 & -\frac{1}{2} & \frac{5}{24} \end{bmatrix} \begin{pmatrix} \frac{d\mathcal{U}}{dx}\Delta x \\ \frac{d^2\mathcal{U}}{dx^2}\Delta x^2 \\ \frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \end{pmatrix} \tag{13}$$

Now, a linear combination of the first order undivided differences is sought such that it results in the term inside the first pair of square brackets on the RHS of Eq. (11).

$$\begin{gathered} A^{(4)}_{0,10}\Delta^1[\bar{u}_{i-3}] + A^{(4)}_{0,11}\Delta^1[\bar{u}_{i-2}] + A^{(4)}_{0,12}\Delta^1[\bar{u}_{i-1}] = \frac{d\mathcal{U}}{dx}\Delta x + \frac{1}{24}\frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \\ \begin{pmatrix} A^{(4)}_{0,10} \\ A^{(4)}_{0,11} \\ A^{(4)}_{0,12} \end{pmatrix}^T \begin{pmatrix} \Delta^1[\bar{u}_{i-3}] \\ \Delta^1[\bar{u}_{i-2}] \\ \Delta^1[\bar{u}_{i-1}] \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{1}{24} \end{pmatrix}^T \begin{pmatrix} \frac{d\mathcal{U}}{dx}\Delta x \\ \frac{d^2\mathcal{U}}{dx^2}\Delta x^2 \\ \frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \end{pmatrix} \end{gathered} \tag{14}$$

Eqs. (13) and (14) lead to the following linear system which can be solved for the constants $A^{(4)}_{0,1k}$.

$$\begin{bmatrix} 1 & 1 & 1 \\ -\frac{5}{2} & -\frac{3}{2} & -\frac{1}{2} \\ \frac{77}{24} & \frac{29}{24} & \frac{5}{24} \end{bmatrix} \begin{pmatrix} A^{(4)}_{0,10} \\ A^{(4)}_{0,11} \\ A^{(4)}_{0,12} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{1}{24} \end{pmatrix} \Rightarrow \begin{pmatrix} A^{(4)}_{0,10} \\ A^{(4)}_{0,11} \\ A^{(4)}_{0,12} \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ -\frac{7}{6} \\ \frac{11}{6} \end{pmatrix} \tag{15}$$

Comparing the values of $A^{(4)}_{0,1k}$ with those in Eq. (8), it can be observed that there is a difference in sign. However, this is inconsequential as the term inside each square bracket is squared.

Next, the second order undivided differences $(m = 2)$ can be computed from Eq. (13) as follows.

$$\begin{pmatrix} \Delta^2[\bar{u}_{i-3}] \\ \Delta^2[\bar{u}_{i-2}] \end{pmatrix} = \begin{bmatrix} -1 & 1 & \\ & -1 & 1 \end{bmatrix} \begin{pmatrix} \Delta^1[\bar{u}_{i-3}] \\ \Delta^1[\bar{u}_{i-2}] \\ \Delta^1[\bar{u}_{i-1}] \end{pmatrix} = \begin{bmatrix} 1 & -2 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} \frac{d^2\mathcal{U}}{dx^2}\Delta x^2 \\ \frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \end{pmatrix} \tag{16}$$

Now, a linear combination of the second order undivided differences is sought such that it results in the term inside the second pair of square brackets on the RHS of Eq. (11).

$$A^{(4)}_{0,20}\Delta^2[\bar{u}_{i-3}] + A^{(4)}_{0,21}\Delta^2[\bar{u}_{i-2}] = \frac{d^2\mathcal{U}}{dx^2}\Delta x^2 \tag{17}$$

$$\begin{pmatrix} A_{0,20}^{(4)} \\ A_{0,21}^{(4)} \end{pmatrix}^T \begin{pmatrix} \Delta^2[\bar{u}_{i-3}] \\ \Delta^2[\bar{u}_{i-2}] \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} \frac{d^2\mathcal{U}}{dx^2}\Delta x^2 \\ \frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \end{pmatrix}$$

Eqs. (16) and (17) lead to the following linear system which can be solved for the constants $A_{0,2k}^{(4)}$.

$$\begin{bmatrix} 1 & 1 \\ -2 & -1 \end{bmatrix} \begin{pmatrix} A_{0,20}^{(4)} \\ A_{0,21}^{(4)} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} A_{0,20}^{(4)} \\ A_{0,21}^{(4)} \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \tag{18}$$

Finally, the third order undivided difference $(m = 3)$ can be computed from Eq. (16) as follows.

$$\Delta^3[\bar{u}_{i-3}] = [-1 \quad 1] \begin{pmatrix} \Delta^2[\bar{u}_{i-3}] \\ \Delta^2[\bar{u}_{i-2}] \end{pmatrix} = \frac{d^3\mathcal{U}}{dx^3}\Delta x^3 \tag{19}$$

By inspection with the term inside the third pair of square brackets on the RHS of Eq. (11), it can be seen that the constant $A_{0,30}^{(4)} = 1$. The derivation procedure is similar for other combinations of r and j. The constants $A_{j,mk}^{(r)}$ are listed in Tables 4-7 in the Appendix for $r = 3$ to $r = 6$.

Determining the constants $A_{j,mk}^{(r)}$ for a $(2r-1)\mathrm{th}$ order WENO scheme requires solving $r-1$ linear systems, one for each order of undivided differences from first order to $(r-1)\mathrm{th}$ order. Obviously, all $r-1$ matrices must remain non-singular for the procedure to be successful. The fact that the matrices remain non-singular for any r can be proved as follows. The general expression for the cell average $\bar{u}_{i+k}$ can be obtained from an $(r-1)\mathrm{th}$ order polynomial approximation as given below.

$$\begin{aligned} \bar{u}_{i+k} = \Delta^0[\bar{u}_{i+k}] &= \frac{1}{\Delta x}\int_{x_i+(k-1/2)\Delta x}^{x_i+(k+1/2)\Delta x} \left[\sum_{n=0}^{r-1} \frac{1}{n!}\frac{d^n\mathcal{U}}{dx^n}(x-x_i)^n\right] dx \\ &= \sum_{n=0}^{r-1}\left\{\frac{1}{(n+1)!}\left[\left(k+\frac{1}{2}\right)^{n+1} - \left(k-\frac{1}{2}\right)^{n+1}\right]\right\}\frac{d^n\mathcal{U}}{dx^n}\Delta x^n \end{aligned} \tag{20}$$

For simplicity, the notation $\frac{d^0\mathcal{U}}{dx^0} = \mathcal{U}_i$ has been introduced in the above result. The derivation procedure begins with the linear system $\Delta^0[\overline{\boldsymbol{u}}] = R_0 d\boldsymbol{\mathcal{U}}_0$ similar to Eq. (12) where $\Delta^0[\overline{\boldsymbol{u}}]$ represents the vector of r cell averages in sub-stencil j and $d\boldsymbol{\mathcal{U}}_0$ the vector of the Taylor series terms $\frac{d^n\mathcal{U}}{dx^n}\Delta x^n$

from $n = 0$ to $n = r - 1$. Elements of the $r \times r$ matrix R_0 are the coefficients inside the curly brackets in Eq. (20) evaluated for appropriate values of k and n. Since the point value $\mathcal{U}_i = (R_0^{-1} \Delta^0[\overline{\boldsymbol{u}}])$ can be uniquely determined from the polynomial approximation, R_0 must be invertible.

The vector of first order undivided differences $\Delta^1[\overline{\boldsymbol{u}}]$ is obtained by multiplying the $(r-1) \times r$ difference matrix D_r to $\Delta^0[\overline{\boldsymbol{u}}]$ as shown below.

$$\Delta^1[\overline{\boldsymbol{u}}] = \overbrace{\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}}^{D_r} \Delta^0[\overline{\boldsymbol{u}}] = (D_r R_0) d\boldsymbol{\mathcal{U}}_0 \tag{21}$$

The difference matrix D_r has a rank of $r - 1$. The $(r-1) \times r$ matrix $(D_r R_0)$ must also have a rank of $r - 1$ since multiplication by an invertible matrix preserves rank. Substituting $n = 0$ into the term inside the curly brackets in Eq. (20), it can be easily verified that the coefficient of $\frac{d^0 \mathcal{U}}{dx^0} \Delta x^0$ in $\Delta^0[\bar{u}_{i+k}]$ is one for all values of k. Therefore, the first column of R_0 consists of ones. Applying the difference matrix D_r to R_0 results in the first column of $(D_r R_0)$ being all zeros, i.e., $(D_r R_0)$ has the form $(D_r R_0) = [\mathbf{0} \quad R_1]$ where R_1 is an $(r-1) \times (r-1)$ matrix. For $(D_r R_0)$ to have rank $(r-1)$, R_1 must have the full rank of $r - 1$ since $\text{span}\{\text{cols}(D_r R_0)\} = \text{span}\{\text{cols}(R_1)\}$. Thus, R_1 is also non-singular. Since the first column of $(D_r R_0)$ consists of zeros, Eq. (21) can be simplified to $\Delta^1[\overline{\boldsymbol{u}}] = R_1 d\boldsymbol{\mathcal{U}}_1$ where $d\boldsymbol{\mathcal{U}}_1$ is the vector of Taylor series terms $\frac{d^n \mathcal{U}}{dx^n} \Delta x^n$ from $n = 1$ to $n = r - 1$. This is precisely the result obtained earlier in Eq. (13) for the particular case of $r = 3$ and $j = 0$. Comparing the matrices in Eqs. (13) and (15), it can be concluded that determining constants $A_{j,1k}^{(r)}$ requires R_1^T to be non-singular. Since $(R_1^T)^{-1} = (R_1^{-1})^T$ and since R_1 has been shown to be non-singular, R_1^T is also non-singular and the constants $A_{j,1k}^{(r)}$ can be uniquely determined.

The argument for higher orders proceeds inductively in the same manner. In general, $\Delta^m[\overline{\boldsymbol{u}}] = R_m d\boldsymbol{\mathcal{U}}_m$ and $\Delta^{m+1}[\overline{\boldsymbol{u}}] = D_{r-m} \Delta^m[\overline{\boldsymbol{u}}] = (D_{r-m} R_m) d\boldsymbol{\mathcal{U}}_m$. Here R_m is an $(r-m) \times (r-m)$ matrix, $d\boldsymbol{\mathcal{U}}_m$ is the vector of Taylor series terms $\frac{d^n \mathcal{U}}{dx^n} \Delta x^n$ from $n = m$ to $n = r - 1$, and D_{r-m} is the

$(r-m-1)\times(r-m)$ difference matrix. The crucial point is that the first column of R_m consists of ones for all m. In other words, the coefficient of $\frac{d^m u}{dx^m}\Delta x^m$ in $\Delta^m[\bar{u}_{i+k}]$ is one regardless of the value of k. Denoting this coefficient as C_m, it can be shown from Eq. (20) that C_m has the following form.

$$C_m = \frac{1}{(m+1)!}\sum_{n=0}^{m+1}(-1)^n \binom{m+1}{n}\left(k+m+\frac{1}{2}-n\right)^{m+1} \tag{22}$$

The proof that $C_m = 1$ follows from Eq. (6.22) in Ref. [14]. So, the $(r-m-1)\times(r-m)$ matrix $(D_{r-m}R_m)$ has a first column of zeros. The remaining $(r-m-1)$ columns form the matrix R_{m+1}. Based on the same argument used for $m=0$, it follows that if R_m is non-singular, then so is R_{m+1}. Since an $(r-1)\mathrm{th}$ polynomial reconstruction on uniform grid always produces a non-singular matrix R_0, all matrices R_m for $m=1$ to $m=r-1$ are non-singular. Therefore, the procedure can be successfully extended to any order.

2.3 Comparison of algorithms

Since the undivided differences $\Delta^m[\bar{u}_i]$ can be computed efficiently, the proposed algorithm brings about significant computational savings compared to the compact implementation. The approximate number of floating-point operations required for computing the r smoothness indicators of a $(2r-1)\mathrm{th}$ order WENO scheme using the different implementations is listed in Table 1. The operations required to compute the undivided differences have been accounted for in the operation count for the proposed algorithm. The proposed algorithm requires only about 60% the number of multiplication operations and about 80% the number of addition/subtraction operations as the compact implementation. A comparison of the number of constants required (including Q_m) is also given in Table 1. It can be seen that the proposed algorithm requires only about half the number of constants as the compact implementation. Hence, the proposed algorithm can be implemented relatively faster.

The compact and proposed algorithms were implemented in an in-house Euler code which uses a hybrid flux methodology [15]. Unlike conventional Euler codes which reconstruct fluxes, the hybrid flux methodology relies on the reconstruction of primitive variables predominantly. This allows the

Table 1: Comparison of number of floating-point operations (±, ×) and constants (*C*) required for smoothness indicators

	Compact [Eq. (6)]			Proposed [Eq. (9)]		
WENO5	15±	30×	20C	13±	18×	11C
WENO7	44±	72×	51C	35±	44×	27C
WENO9	95±	140×	104C	71±	85×	54C
WENO11	174±	240×	185C	124±	144×	95C

undivided differences to be reused for the left- and right-biased WENO reconstructions. The in-house code was used to compute the Shu-Osher shock-entropy wave interaction problem [16] and the double Mach reflection problem [17] at several difference resolutions. The double Mach reflection problem was set up using the second alternative method described in Ref. [18] to obtain clean, artefact-free solutions. Solutions obtained from both algorithms were identical. The speedups achieved by the proposed algorithm over the compact algorithm are given in Table 2.

Though the speedups were somewhat marginal for fifth order, they started to increase quickly for higher orders. The proposed algorithm shortened the computation time by about 6% and 10% for ninth and eleventh orders, respectively. The speedups are expected to increase further for even higher orders. With a proper implementation, there is a potential for greater savings when the undivided differences are computed for an entire row/column of cells at a time in structured Cartesian grids as adjacent faces along a row/column share all but one sub-stencils.

3 Conclusion

Computing sub-stencil smoothness indicators is the most expensive operation in the WENO reconstruction procedure. In this paper, an efficient algorithm for computing these quantities is presented. For a $(2r-1)\mathrm{th}$ order WENO scheme, a table of undivided differences is constructed up to order $r-1$ in a recursive manner. Then, the smoothness indicators are computed as squares of linear combinations of these undivided differences ensuring positivity of the computed values. It has

Table 2: Comparison of speedups achieved using proposed algorithm [Eq. (9)] over compact algorithm [Eq. (6)]

Case	Resolution	WENO5	WENO7	WENO9	WENO11
Shock-entropy	400	1.014	1.024	1.049	1.081
wave interaction	800	1.016	1.011	1.058	1.117
problem	1600	1.021	1.018	1.064	1.130
Double Mach	480×120	1.002	1.017	1.067	1.092
reflection	960×240	1.008	1.032	1.062	1.094
problem	1920×480	1.003	1.038	1.067	1.098

been shown that the proposed algorithm requires considerably fewer floating-point operations and fewer constants compared to the compact implementation.

4 Appendix

The constants $Q_m/(m!)^2$ are given in Table 3.

Table 3: Constants $Q_m/(m!)^2$

m	1	2	3	4	5
$Q_m/(m!)^2$	1	$\frac{13}{12}$	$\frac{781}{720}$	$\frac{1421461}{1310400}$	$\frac{21520059541}{19838649600}$

The constants $A^{(r)}_{j,mk}$ are given in Tables 4-7 for $r = 3$ to $r = 6$. -

Table 4: Constants $A^{(r)}_{j,mk}$ for $r = 3$

		k	
j	m	0	1
0	1	$-\frac{1}{2}$	$\frac{3}{2}$
	2	1	-
1	1	$\frac{1}{2}$	$\frac{1}{2}$
	2	1	-

2	1	$\frac{3}{2}$	$-\frac{1}{2}$
	2	1	-

Table 5: Constants $A^{(r)}_{j,mk}$ for $r = 4$

		k		
j	m	0	1	2
0	1	$\frac{1}{3}$	$-\frac{7}{6}$	$\frac{11}{6}$
	2	-1	2	-
	3	1	-	-
1	1	$-\frac{1}{6}$	$\frac{5}{6}$	$\frac{1}{3}$
	2	0	1	-
	3	1	-	-
2	1	$\frac{1}{3}$	$\frac{5}{6}$	$-\frac{1}{6}$
	2	1	0	-
	3	1	-	-
3	1	$\frac{11}{6}$	$-\frac{7}{6}$	$\frac{1}{3}$
	2	2	-1	-
	3	1	-	-

-

Table 6: Constants $A^{(r)}_{j,mk}$ for $r = 5$

		k			
j	m	0	1	2	3
0	1	$-\frac{1}{4}$	$\frac{13}{12}$	$-\frac{23}{12}$	$\frac{25}{12}$
	2	$\frac{119}{130}$	$-\frac{184}{65}$	$\frac{379}{130}$	-
	3	$-\frac{3}{2}$	$\frac{5}{2}$	-	-
	4	1	-	-	-
1	1	$\frac{1}{12}$	$-\frac{5}{12}$	$\frac{13}{12}$	$\frac{1}{4}$
	2	$-\frac{11}{130}$	$\frac{11}{65}$	$\frac{119}{130}$	-
	3	$-\frac{1}{2}$	$\frac{3}{2}$	-	-

	4	1	-	-	-
2	1	$-\frac{1}{12}$	$\frac{7}{12}$	$\frac{7}{12}$	$-\frac{1}{12}$
	2	$-\frac{11}{130}$	$\frac{76}{65}$	$-\frac{11}{130}$	-
	3	$\frac{1}{2}$	$\frac{1}{2}$	-	-
	4	1	-	-	-
3	1	$\frac{1}{4}$	$\frac{13}{12}$	$-\frac{5}{12}$	$\frac{1}{12}$
	2	$\frac{119}{130}$	$\frac{11}{65}$	$-\frac{11}{130}$	-
	3	$\frac{3}{2}$	$-\frac{1}{2}$	-	-
	4	1	-	-	-
4	1	$\frac{25}{12}$	$-\frac{23}{12}$	$\frac{13}{12}$	$-\frac{1}{4}$
	2	$\frac{379}{130}$	$-\frac{184}{65}$	$\frac{119}{130}$	-
	3	$\frac{5}{2}$	$-\frac{3}{2}$	-	-
	4	1	-	-	-

199 -

Table 7: Constants $A^{(r)}_{j,mk}$ for $r = 6$

		k				
j	m	0	1	2	3	4
0	1	$\frac{1}{5}$	$-\frac{21}{20}$	$\frac{137}{60}$	$-\frac{163}{60}$	$\frac{137}{60}$
	2	$-\frac{54}{65}$	$\frac{443}{130}$	$-\frac{346}{65}$	$\frac{487}{130}$	-
	3	$\frac{114721}{65604}$	$-\frac{81962}{16401}$	$\frac{278731}{65604}$	-	-
	4	-2	3	-	-	-
	5	1	-	-	-	-
1	1	$-\frac{1}{20}$	$\frac{17}{60}$	$-\frac{43}{60}$	$\frac{77}{60}$	$\frac{1}{5}$
	2	$\frac{11}{130}$	$-\frac{22}{65}$	$\frac{11}{26}$	$\frac{54}{65}$	-
	3	$\frac{16315}{65604}$	$-\frac{16358}{16401}$	$\frac{114721}{65604}$	-	-
	4	-1	2	-	-	-
	5	1	-	-	-	-
2	1	$\frac{1}{30}$	$-\frac{13}{60}$	$\frac{47}{60}$	$\frac{9}{20}$	$-\frac{1}{20}$
	2	0	$-\frac{11}{130}$	$\frac{76}{65}$	$-\frac{11}{130}$	-
	3	$-\frac{16487}{65604}$	$\frac{16444}{16401}$	$\frac{16315}{65604}$	-	-

	4	0	1	-	-	-
	5	1	-	-	-	-
3	1	$-\frac{1}{20}$	$\frac{9}{20}$	$\frac{47}{60}$	$-\frac{13}{60}$	$\frac{1}{30}$
	2	$-\frac{11}{130}$	$\frac{76}{65}$	$-\frac{11}{130}$	0	-
	3	$\frac{16315}{65604}$	$\frac{16444}{16401}$	$-\frac{16487}{65604}$	-	-
	4	1	0	-	-	-
	5	1	-	-	-	-
4	1	$\frac{1}{5}$	$\frac{77}{60}$	$-\frac{43}{60}$	$\frac{17}{60}$	$-\frac{1}{20}$
	2	$\frac{54}{65}$	$\frac{11}{26}$	$-\frac{22}{65}$	$\frac{11}{130}$	-
	3	$\frac{114721}{65604}$	$-\frac{16358}{16401}$	$\frac{16315}{65604}$	-	-
	4	2	-1	-	-	-
	5	1	-	-	-	-
5	1	$\frac{137}{60}$	$-\frac{163}{60}$	$\frac{137}{60}$	$-\frac{21}{20}$	$\frac{1}{5}$
	2	$\frac{487}{130}$	$-\frac{346}{65}$	$\frac{443}{130}$	$-\frac{54}{65}$	-
	3	$\frac{278731}{65604}$	$-\frac{81962}{16401}$	$\frac{114721}{65604}$	-	-
	4	3	-2	-	-	-
	5	1	-	-	-	-

5 Acknowledgements

I would like to thank Prof Chan Wai Lee, School of Mechanical and Aerospace Engineering, Nanyang Technological University for his valuable comments which helped to improve this paper significantly.

6 References

[1] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200-212, 1994

[2] G.S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202-228, 1996

[3] A.K. Henrick, T.D. Aslam, and J.M. Powers. Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points. *J. Comput. Phys.*, 207:542-567, 2005

[4] C.H. Kim, Y. Ha, and J. Yoon. Modified non-linear weights for fifth-order weighted essentially non-oscillatory schemes. *J. Sci. Comput.*, 67:299-323, 2016

[5] Y. Ha, C. Ho Kim, Y. Ju Lee, and J. Yoon. An improved weighted essentially non-oscillatory scheme with a new smoothness indicator. *J. Comput. Phys.*, 232:68-86, 2013

[6] Y. Shen and G. Zha. Improvement of the WENO scheme smoothness estimator. *Int. J. Numer. Meth. Fl.* 64:653-675, 2010

[7] S. Zhang and C.-W. Shu. A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions. *J. Sci. Comput.*, 31:273-305, 2007

[8] F. Teng, L. Yuan, and T. Tang. A speed-up strategy for finite volume WENO schemes for hyperbolic conservation laws. *J. Sci. Comput.*, 46:359-378, 2011

[9] A. Baeza, R. Bürger, P. Mulet, and D. Zorío. On the efficient computation of smoothness indicators for a class of WENO reconstructions. *J. Sci. Comput.*, 80:1240-1263, 2019

[10] D.S. Balsara, S. Garain, and C.-W. Shu. An efficient class of WENO schemes with adaptive order. *J. Comput. Phys.*, 326:780-804, 2016

[11] D.S. Balsara, C. Meyer, M. Dumbser, H. Du, and Z. Xu. Efficient implementation of ADER schemes for Euler and magnetohydrodynamical flows on structured meshes – Speed comparisons with Runge–Kutta methods. *J. Comput. Phys.*, 235:934-969, 2013

[12] D.S. Balsara, T. Rumpf, M. Dumbser, and C.-D. Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *J. Comput. Phys.*, 228:2480-2516, 2009

[13] R. Wang, H. Feng, and C. Huang. A new mapped weighted essentially non-oscillatory method using rational mapping function. *J. Sci. Comput.*, 67:540-580, 2016

[14] J. Quaintance and H.W. Gould, *Combinatorial identities for Stirling numbers: the unpublished notes of HW Gould*. 2015: World Scientific.

[15] U S Vevek, B. Zang, and T.H. New. An efficient hybrid method for solving Euler equations. *J. Sci. Comput.*, 81:732-762, 2019

[16] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, *Upwind and High-Resolution Schemes*. 1989, Springer. p. 328-374.

[17] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115-173, 1984

[18] U S Vevek, B. Zang, and T.H. New. On alternative setups of the double Mach reflection problem. *J. Sci. Comput.*, 78:1291-1303, 2019