Implementation of an arbitrary-order Finite-Volume Least-Squares WENO for inviscid Euler equations.

Rafael Castro Mota
rafael.mota@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

August 2021

Abstract

In this thesis, an arbitrary order Least-Squares-WENO (LS-WENO) scheme will be applied for both the one-dimensional and two-dimensional finit volume formulation of the Euler set of equations. Regular and triangular meshes will be used. The code developed is intended to serve as a basis for future work regarding robust high-order methods for transonic and supersonic regimes. WENO schemes work by defyning several data sets (stencils) for the same point of interest and then combining the resulting polynomial models into a single final model. As spurious oscillations are to be avoided near discontinuities and shocks, each polynomial model receives a weight dependent on their oscillating behaviour, hence the name WENO (Weighted Essentially Non-Oscillatory). The regression method used will be the Least-Squares Method as it provides flexibility for future applications.
Keywords: WENO, Least-Squares, Euler, Riemann-Solvers, MOOD

1. Introduction

In CFD, obtaining high-order accuracy allows for the capture of finer flow structures in coarser meshes. However, in the presence of sharp gradients gradients, discontinuities and shocks, typical polynomial regressing methods fail to produce coherent results. Important properties, such as monotonicity and positivity, are not preserved and the Gibbs phenomenon and associated unphysical oscillatory behaviour appear in any polynomial model that approximates data with such nuisances. These challenges normally limit commercial CFD code to second-order TVD schemes.

In this work, an arbitrary order of accruacy Weighted-ENO scheme will be applied for both the one-dimensional and two-dimensional finit volume formulation of the Euler equations. WENO schemes allow for high-order of accuracy in smooth regions of the domain while maintaing positivity and monotonicity near sharp gradients, discontinuities and shocks. They work by defining several data sets, called stencils, for the same point of interest and then combining the resulting approximated polynomial models. As spurious oscillations are to be avoided, each polynomial model receives a weight dependent on their oscillatory behaviour, hence the name WENO (Weighted Essentially Non-Oscillatory). As the underlying architecture does not need complicated algorithm for choice of data, stencils are only chosen once at the pre-processing stage of the CFD code. For future flexibility, the regression method used will be the Weighted-Least-Squares Method.

2. Background

2.1. Compressible 1D Euler Equations

In equation 1, the conservative form for the one-dimensional compressible Euler equations is shown.

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{dx} = 0. \tag{1}$$

The dependent Euler variables vector (Q) and flux vector (F) are described in equation 2.

$$Q = \begin{bmatrix} \rho \\ \rho u_x \\ E \end{bmatrix} \qquad F(x) = \begin{bmatrix} \rho u_x \\ \rho {u_x}^2 + p \\ u_x(E+p) \end{bmatrix} \tag{2}$$

2.1.1 Finite Volume Formulation for the Compressible 1D Euler Equations

A one-dimensional domain can be discretized by N equal lenght cells and immersed in a field of dependent variables Q.

Integrating the set of equations in 1 along a random target cell S_i, one gets equation 3.

$$\int_{x(S_{i1})}^{x(S_{i2})} (\frac{\partial Q}{\partial t} + \frac{\partial F_x}{\partial x}) \;\; \partial x = 0 \tag{3}$$

The cell averaged quantities in the target cell ($\overline{Q}(S_i)$ can be defined as demonstrated in equation

4.

$$\overline{Q}(S_i) = \frac{1}{d(S_i)} \int_{(S_{i1})}^{x(S_{i2})} Q \;\; \partial x \tag{4}$$

Using equation 4, and integrating the flux function ($F(x)$) along the one-dimensional cell, the integral in equation 3 can be expressed by equation 5.

$$d(S_i)\frac{\partial \overline{Q}}{\partial t} = -(F(x\big(S_{i2}) - F(x\big(S_{i1})) \tag{5}$$

The time derivative of the cell averaged quantities ($\frac{\partial \overline{Q}}{\partial t}$) can then, consequently, be expressed as the sum of left and right fluxes , known as residual of the cell ($R(i,t)$), at that instance of time divided by the lenght of the cell ($d(S_i)$). The mathematical form of the previous phrase is shown in equation 6.

$$\frac{\partial \overline{Q}}{\partial t} = -\frac{1}{d(S_i)} R(i,t) \tag{6}$$

2.1.2 Characteristic Form for the Compressible 1D Euler Equations

For decoupling the Euler one-dimensional equations, a Flux Jacobian ($\mathbb{A} = \frac{\partial F}{\partial Q}$) must be computed.

By using eingenvalue decomposition, the Flux Jacobian Matrix can also be decomposed into the product of three matrices, as shown in equation 7.

$$\frac{\partial F}{\partial Q} = \mathbb{X}_R \cdot \Lambda \cdot {\mathbb{X}_R}^{-1} \tag{7}$$

Matrix X_R is composed by the right eigenvectors arranged in columns and matrix Λ by the respective eingenvalues set on the diagonal. Both can be deduced from equation 23 by eliminating the third column and row.

Equation (7) can be used to rewrite the one-dimensional Euler system of equations as 8.

$$\frac{\partial Q}{\partial t} + (X_R \cdot \Lambda \cdot {X_R}^{-1})\frac{\partial Q}{\partial x} = 0 \tag{8}$$

Multiplying the left eigenvalue matrix ($\mathbb{X}_R^{-1}$) on the left of each side of equation 8, one can obtain equation 9.

$${\mathbb{X}_R}^{-1}\frac{\partial Q}{\partial t} + \Lambda \cdot {\mathbb{X}_R}^{-1}\frac{\partial Q}{\partial x} = 0 \tag{9}$$

To circumvent the variation of the right eigenvectors along time and space, an approximated set of equations is used. A constant local approximation of the Jacobian Matrix is employed, as expressed in equation 10.

$$\frac{\partial F}{\partial Q} \approx \widetilde{\mathbb{A}} = \widetilde{\mathbb{X}}_R \cdot \widetilde{\Lambda} \cdot \widetilde{\mathbb{X}}_R^{-1} \tag{10}$$

With the enployment of the locally approximated Flux Jacobian Matrix, equation 9 can be decoupled , taking the form of equation 11.

$$\frac{\partial(\widetilde{\mathbb{X}}_R^{-1}Q)}{\partial t} + \Lambda \cdot \frac{\partial(\widetilde{\mathbb{X}}_R^{-1}Q)}{\partial x} = 0 \Rightarrow \frac{\partial Z}{\partial t} + \Lambda \cdot \frac{\partial Z}{\partial x} = 0 \tag{11}$$

Each individual characteristic quantity in the set of local characteristic variables ($Z = \widetilde{R}^{-1}Q$) remains constant along their respective characteristic curve. The Z quantities are each related to Riemman invariants (J) of the one-dimensional Euler equations.

2.2. Compressible 2D Euler Equations

In conservative form, the compressible two-dimensional Euler equations take the configuration of equation 12.

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0. \tag{12}$$

The Euler dependent variables vector (Q), the flux vector for the xx axis direction (F) and yy axis direction (G) are described in equation 13.

$$Q = \begin{bmatrix} \rho \\ \rho u_x \\ \rho u_y \\ E(x) \end{bmatrix}; F = \begin{bmatrix} \rho u_x \\ \rho {u_x}^2 + p \\ \rho u_x u_y \\ u_x(E+p) \end{bmatrix}; G = \begin{bmatrix} \rho u_x \\ \rho u_x u_y \\ \rho {u_y}^2 + p \\ u_y(E+p) \end{bmatrix} \tag{13}$$

2.2.1 Finite Volume Formulation for the Compressible 2D Euler Equations

A two-dimensional domain, immersed in a field of properties Q, may be discretized by N number of two-dimensional cells, such as the four-sided one in figure 1.

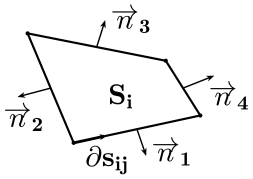

Figure 1: Four-sided target cell (S_i).

Integrating equation 12 on the target two-dimensional cell (S_i), one gets equation 14.

$$\iint_{S_i} (\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y}) \partial S_i = 0 \tag{14}$$

The Green theorem can be used to transform the integral of the fluxes in equation 14 into equation 15.

$$\iint_S (\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y})\partial x \partial y = \oint_{s_i} F\partial y - G\partial x \quad (15)$$

If the outter boundary (s_i) of the target cell is traversed in anti-clockwise fashion, as represented in figure 1 (for face j), equation 15 can be manipulated into equation 16.

$$\oint_{s_i} F\partial y - G\partial x = \oint_{s_i} (F \cdot n_x + G \cdot n_y)\partial s_i \quad (16)$$

An integral around the boundary of the target cell can be described as a sum of the integral in each face. Therefore, it is possible to express equation 16 as equation 17

$$\sum_{j=1}^{4} (\int_{s_{ij}} F \cdot n_x(S_{ij}) + G \cdot n_y(S_{ij}))\ \ \partial s_{ij} = R(S_i, t) \quad (17)$$

Finally, the time derivative of the cell averaged quantities (analogous to equation 4) can be expressed by the negative of the residual for the target cell at that given instance of time divided by the area of said cell ($A(S_i)$). Equation 18 reflects the final form of a finit volume formulation for the two-dimensional Euler equations.

$$\frac{\partial \overline{Q}}{\partial t} = -\frac{1}{A(S_i)} \cdot R(S_i, t) \quad (18)$$

2.2.2 Characteristic Form for the Compressible 2D Euler Equations

Although it is not possible to decouple the original two-dimensional Euler equations in the way demonstrated in section 2.1.2, a different set of differential equations can be derived for each face. By projecting the Euler equations onto an axis system based on the normal ($\vec{n}$) and tangent directions ($\vec{v}$) for a said face, the formulation in equation 19 is possible.

$$\frac{\partial Q^{nv}}{\partial t} + \frac{\partial F^n}{\partial n} + \frac{\partial F^v}{\partial v} = 0 \quad (19)$$

The vector for the projected Euler dependent variables (Q^{nv}) and the flux vectors for the tangent (F^v) and normal (F^n) directions are shown in equation 20.

$$Q^{nv} = \begin{bmatrix} \rho \\ \rho u_n \\ \rho u_v \\ E \end{bmatrix}; F^n = \begin{bmatrix} \rho \\ \rho {u_n}^2 + p \\ \rho u_n u_v \\ u_n(E+p) \end{bmatrix}; F^v = \begin{bmatrix} \rho \\ \rho u_n u_v \\ \rho {u_v}^2 + p \\ u_v(E+p) \end{bmatrix}; \quad (20)$$

As a local approximation, considering the Q^{nv} quantities to not change along the tangent direction allows the previous system of equations 19 to be rewritten as equation 21.

$$\frac{\partial Q^{nt}}{\partial t} + \frac{\partial F^n}{\partial n} = 0 \quad (21)$$

To the previous equation (21), one can apply a similar method to that of section 2.1.2. In equation 22, the Flux Jacobian is expressed as a product of three matrices (eigenvalue decomposition).

$$\frac{\partial F^n}{\partial Q^n} = \mathbb{X}_R \cdot \Lambda \cdot {\mathbb{X}_R}^{-1} \quad (22)$$

As in section 2.1.2, matrix X_R represents the right eigenvectors for the Flux Jacobian matrix arranged in columns and matrix Λ the diagonal matrix of the respective eingevalues (equation 23).

$$\mathbb{X}_R = \begin{bmatrix} 1 & 1 & 0 & 1 \\ u-a & u & 0 & u+a \\ u_v & u_v & 1 & u_v \\ H-ua & \frac{1}{2}V^2 & u_v & H+ua \end{bmatrix}; \quad \Lambda = \begin{bmatrix} u_n - a & 0 & 0 & \\ 0 & u_n & 0 & 0 \\ 0 & 0 & u_n & 0 \\ 0 & 0 & 0 & u_n + a \end{bmatrix} \quad (23)$$

The total specific enthalpy (H) can be expressed by equation 24.

$$H = \frac{(E+p)}{\rho} = \frac{a^2}{\gamma - 1} + \frac{1}{2}u^2 \quad (24)$$

Following an analogous procedure to that of section 2.1.2, equation 25 can be achieved.

$$\begin{gathered} \frac{\partial(\widetilde{\mathbb{X}}_R^{-1} Q^n)}{\partial t} + \Lambda \cdot \frac{\partial(\widetilde{\mathbb{X}}_R^{-1} Q^n)}{\partial n} = 0 \\ \Rightarrow \frac{\partial Z^n}{\partial t} + \Lambda \cdot \frac{\partial Z^n}{\partial n} = 0 \end{gathered} \quad (25)$$

2.3. Harten Lax-van Leer Contact (HLLC)

Many numerical schemes in CFD require solving Riemann problems for computation of inter-cell fluxes. The HLLC Riemann solver approximately solves a range of Riemann problems and is an improvement over the HLL Riemann solver, which only considers an expansion front and a shock wave. The contact discontinuity is restored by adding a third party to the wave structure. A possible wave configuration is shown in figure 2.

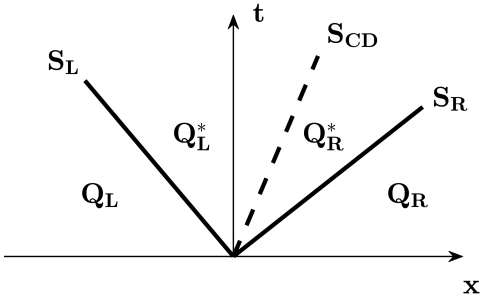

Figure 2: Possible configuration for the wave structure in the HLLC Riemann solver.

Similarly to the HLL Riemann solver, the Q_R^* and Q_L^* states are a spatial average of the analytic solution between their respective waves. The integral form for the two states is shown in equation 26.

$$Q_K^* = \frac{1}{t(S_K - S_{CD})} \int_{tS_{CD}}^{tS_K} Q(x,t) \;\; \partial x \tag{26}$$

The index K in equation 29 can represent either the state between the shock wave and the contact discontinuity ($K = R$) or between the contact discontinuity and the expansion front ($K = L$).

By applying Rankine-Hugoniot jump-conditions on the different waves, the consistency relations in equation 27 are computed.

$$\begin{aligned} F_K^* &= F_K + S_K \cdot (Q_K^* - Q_K) \; ; \\ F_L^* &= F_R^* + S_{CD} \cdot (Q_L^* - Q_R^*) \; ; \end{aligned} \tag{27}$$

The consistency relations in equation 27 can be manipulated in order to define the contact discontinuity wave speed (S_{CD}) as a function of left and right undisturbed state quantities (Q_L and Q_R), which are prescribed. Equation 28 shows the final relation between the forementioned quantities.

$$S_{CD} = \frac{p_R - p_L + \rho_L u_L (S_L - u_L) - \rho_R u_R (S_R - u_R)}{\rho_L (S_L - u_L) - \rho_R (S_R - u_R)} \tag{28}$$

Again, manipulating the system of equations in 27, one can define the averaged states that exist between the shock wave and the expansion wave front. Taken from [3], equation 29 gives a mathematical expression for these averaged states.

$$Q_K^* = \frac{\rho_K (S_K - u_k)}{S_K - S_{CD}} \begin{bmatrix} 1 \\ S_{CD} \\ Q_K(:) \\ \frac{E_K}{\rho_K} + [\frac{(S_{CD} - u_K)}{S_{CD}} + \dots \\ \dots + \frac{p_K (S_{CD} - u_K)}{\rho_K (S_K - u_K)}] \end{bmatrix} \tag{29}$$

The third matrix entry ($Q_K(:)$) represents a random passive scalar quantity. For two-dimensional schemes, tangential velocity at the face is often treated like a passive scalar.

The fluxes in the star region can be completely defined by using equation 29 along with the first and third relations for the consistency conditions in equation 27.

The flux of interest is the one on the t-axis. The wave structure can have different configurations, and therefore, the solver needs to take into account some different scenarios, such as supersonic flows. The HLLC flux is defined in equation 30.

$$F(S_{ij}) = \begin{cases} F_L & , S_L \geq 0 \\ F_L^* & , S_L \leq 0 \cap S_{CD} \geq 0 \\ F_R^* & , S_R \geq 0 \cap S_{CD} \leq 0 \\ F_R & , S_R \leq 0 \end{cases} \tag{30}$$

The HLLC solver does not, by default, preserve entropy across expansion regions and the star region fluxes are not directly obtained from the star region computed quantitites. Equation 31 shows this inequality.

$$F_K^* \neq F(Q_K^*) \tag{31}$$

3. Weighted Least-Squares Method

The Least-Squares Method (LS) is a classic tool in regression analysis useful for approximating a regression model to a set of data $(Y(X), X)$ by minimization of the square of the error in the data points (residuals). Having a set of basis functions Φ_h, the regression model has $\mathbb{H}$ number of terms and can be defined by equation 32.

$$\sum_{h=1}^{\mathbb{H}} C_h \Phi_h = C_1 \Phi_1 + C_2 \Phi_2 + ... + C_H \Phi_{\mathbb{H}} \tag{32}$$

In each data point $(Y(X_n), X_n)$, the respective residual is computed by using equation 33.

$$r_n = Y(X_n) - \sum_{h=1}^{\mathbb{H}} C_h \Phi_h(X_n) \tag{33}$$

If the data set has $N_{\mathbb{S}}$ number of data points, we can present the residuals in matrix form, as shown in the equations contained in 34.

$$\begin{aligned} r = Y^{LS} - D^{LS} \cdot C_f ; r = \begin{bmatrix} r_1 \\ ... \\ r_{N_S} \end{bmatrix} ; C_f = \begin{bmatrix} C_1 \\ .. \\ C_{\mathbb{H}} \end{bmatrix} ; \\ Y^{LS} = \begin{bmatrix} Y(X_1) \\ ... \\ Y(X_{N_S}) \end{bmatrix} ; D^{LS} = \begin{bmatrix} \Phi_1(X_1) \; ... \; \Phi_H(X_1) \\ ... \\ \Phi_1(X_{N_S}) \; ... \; \Phi_H(X_{N_S}) \end{bmatrix} ; \end{aligned} \tag{34}$$

It is very important to state that in a typical least squares method application the number of data points is greater than the number of coefficients ($N_{\mathbb{S}} > H$). Having more data points than coefficients provides robustness and flexibility.

In CFD, the approximate model is usually only locally pertinent as it is used for approximating local numerical fluxes and flow quantities. Consequently, the model should be more accurate for points closer to the area of interest. To provide this feature on the regression, the Weighted-Least-Square Problem, presented in equation 35, is used.

$$min(\sum_{n=1}^{K} W_n^{LS}(r_n)^2) \quad w.r.t \;\; C_f \tag{35}$$

Based on [5], the weight function in equation 36 will be one used for a p^{th} degree polynomial model centered on a generic point with coordinate vector X_0.

$$W_n^{LS} = (\frac{1}{X_1 - X_0})^{\frac{\wp}{2}} \tag{36}$$

The weight function also has a reducing effect on the condition number of the matrices calculated (see [4]). Equation 35 can be expressed, in matrix form, by equation 37.

$$min((Y^{LS} - DC_f)^T W^{LS}(Y^{LS} - D \cdot C_f)) \quad w.r.t \;\; C_f \tag{37}$$

Matrix W^{LS} is a diagonal matrix with the respective weight for the residual of each data point. The configuration of W^{LS} is shown in equation 38.

$$W^{LS} = \begin{bmatrix} W_1^{LS} & & \\ & \ddots & \\ & & W_{N_{LS}}^{LS} \end{bmatrix} \tag{38}$$

As the residuals depend linearly on the model coefficients, the minimum of the weighted squared residuals can be found by derivating with respect to C_f and equalling to zero, as depicted in equation 39.

$$\frac{d(r^T \cdot W^{LS} \cdot r)}{dC_f} = 0 \tag{39}$$

Equation 39 can be expressed as equation 40.

$$(D^T \cdot W^{LS} \cdot D) \cdot C_f = (D \cdot W^{LS}) \cdot Y_s \tag{40}$$

In equation 41, the terms that only depend on the independent variables of the data set are compiled into a single matrix (P) and the formula for the coefficients in the Weighted-Least-Square Problem are presented.

$$C_f = P^{LS} \cdot Y_s; \quad P^{LS} = (D^T \cdot W^{LS} \cdot D)^{-1} \cdot D \cdot W^{LS} \tag{41}$$

It should be noted that matrix P is only dependent on the independent variables (X_n) of the data set.

4. Implementation

2D WENO - Implementation and Results

Throughout this chapter, a proposed version of a WENO scheme for a one and two-dimensional finite-volume formulation of the Euler equations is presented and its inner-workings explained. The scheme is used to approximate the residual of the finite-volume formulation, present in section 2.2, of a given target cell (S_i) on a domain discretized by N number of cells.

The polynomial model used is presented in section 4.1 of this chapter. The regression method used is the Least-Squares Method which is explained with detail in section 3.

For a given target cell, several polynomials of equal degree, each one resultant from a different collection of data (stencils), are computed. The procedure of stencil creation is explained in section 4.2.

The setup of the Least-Squares Problem in the presented WENO scheme is discussed in section 4.3.

A final polynomial for a given target cell (WENO polynomial) is calculated using a combination of the polynomials resultant from each stencil. As the Gibbs phenomenon is to be mitigated in this final polynomial, more emphasis is given to polynomials that present less oscillations. The procedure is detailed in section 4.4

The resulting polynomials for certain types of data are used to extrapolate adequate quantities on the cell faces. A Riemann solver is then used to resolve the numerical inter-cell fluxes given the two intersecting extrapolated values: one from the polynomial models centred on the target cell, the others from the polynomial models centred on the neighbouring cell (see section 4.5)

4.1. Polynomial Model

The polynomial model is introduced in section 3 as a linear combination of basis functions. For the proposed WENO schemes of arbitrary order of accuracy (r), the basis functions of up to $\wp^{th}$ degree were chosen to be modified orthogonal Taylor functions (B_h^{2D}) fo the two-dimensional schemes and Legendre polynomials for the one-dimensional ones (B_h^{1D}). The functions are expressed in equation 42.

$$a,b \in \mathbb{N}; a+b \leq \wp : \begin{cases} B^{1D} = \frac{1}{2^a \cdot a!} \cdot \frac{d^a}{dx^a}(x^2-1)^a; \\ \\ B^{2D} = x^a y^b + Cte \end{cases} \tag{42}$$

As an added constraint on the polynomial model, the basis functions are required to have a null integration value in the target cell (S_i). For the two-dimensional basis functions, this property can be obtained by defining the constant Cte as done in equation 43.

$$Cte = -\int_{S_i} x^a y^b \ \partial S_i \tag{43}$$

If left alone, the previous requirement would imply different basis functions for every cell, as happens in [4]. For square and triangular meshes, a computationally friendlier way of ensuring a null integration value is through the usage of auxiliary coordinate systems (ϵ, η), in which all the cells are mapped into the same standard cell (S_{st}). For the one-dimensionals scenario, this tactic is also useful as Legendre polymials possess a null integration value in the $[-1, 1]$ interval.

The complete form for the basis functions is described in equation 44.

44.

$$\begin{gathered} B^{1D}(\epsilon) = \frac{1}{2^a \cdot a!} \cdot \frac{d^a}{d^a}(\epsilon^2-1)^a; \\ B^{2D}(\epsilon,\eta) = \epsilon^n \eta^m - \textstyle\int_{S_{st}} \epsilon^a \eta^b \ \partial S_{st} \end{gathered} \tag{44}$$

The complete polynomial model for an arbitrary quantity U in a auxiliary coordinate system is shown in equation 45.

$$M(\epsilon,\eta) = \overline{U}(S_i) + \textstyle\sum_{h=1}^{H} C_h B_h(\epsilon,\eta) \tag{45}$$

$$H^{1D} = \wp; \quad H^{2D} = \tfrac{1}{2}(\wp+1)(\wp+2) - 1;$$

Linear mapping ensures that a random cell averaged quantity ($\overline{U}$) remains the same for both the original and auxiliary coordinate systems and, thus, respects the property in equation 46.

$$\overline{U}(S_i) = \int_{S_i} \frac{U(x,y,t)}{A(S_i)} \ \partial S_i = \int_{S_{st}} \frac{U(\epsilon,\eta,t)}{A(S_{st})} \ \partial S_{st} \tag{46}$$

4.1.1 Auxiliary Coord. Transformation for 1D Cells

Each one-dimensional target cell gets subjected to the linear transformation in equation 47 and mapped into an cell ranging from minus unity to unity ($[-1, 1]$).

$$\epsilon = J \cdot x - x_c(S_i); \quad J = \frac{1}{2 \cdot d(S_i)} \tag{47}$$

4.1.2 Auxiliary Coord. Transformation for Triangular Cells

Each triangular target cell gets subjected to the linear transformation in equation 48 and mapped into an equilateral triangle with unitary side.

$$\begin{gathered} \begin{bmatrix} x \\ y \end{bmatrix} = J \begin{bmatrix} \epsilon \\ \eta \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix}, \\ J = \begin{bmatrix} 2x_1 - 3x_c + x_3 & \sqrt{3}(x_3 - x_c) \\ 2y_1 - 3y_c + y_3 & \sqrt{3}(y_3 - y_c) \end{bmatrix} \end{gathered} \tag{48}$$

4.1.3 Auxiliary Coord. Transformation for Square Cells

Each square target cell gets subjected to the linear transformation in equation 49 and mapped into a square with unitary side.

$$\begin{bmatrix} x \\ y \end{bmatrix} = J \begin{bmatrix} \epsilon \\ \eta \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix}, J = \begin{bmatrix} |x_2 - x_1| & 0 \\ 0 & |y_2 - y_1| \end{bmatrix} \tag{49}$$

4.2. Stencil Creation

As aforementioned, WENO schemes require various polynomial regressions for propper functioning around discontinuities. A central and various directional stencils with non-intersecting data are for this reason used. For the two-dimensional, the methodology in [4] is followed and stencil search areas are defined through the angle formed by the center of the target cell and two of its vertices.

Each target cell has a number of directional stencils equal to its number of faces plus a central one. The construction of the directional stencils is done by adding the subsequent neighbouring cells with respect to the search area in which they are located. The procedure is done until the desired number of cells in each stencil is reached. Next to boundaries, if a diretional stencil fails to be completed it is simply discarded.

In equation 50 the number of data in each stencil (N_S) is given as a function of the the degree of the polynomial model used (p) for the one and two-dimensional cases.

$$N_S^{1D} = \wp + 1; \quad N_S^{2D} = (\wp+1)(\wp+2) - 2 \tag{50}$$

The central stencil is made up of the closest cells to the target cell taken from the various diretional stencils.

4.3. Least-Square Problem for Characteristic Variables

Having the necessary data sets (cells in central and diretional stencil) for a given target cell and quantity, the Least-Squares Method can be used to approximate the polynomial model in section 4.1. A Least-Squares Problem and the respetive matrices must be setup.

4.3.1 $\mathbf{D^{LS}}$ Matrix

The Least-Square Problem is set up by approximating the average value of the polyomial model in each stencil cell ($\mathbb{S}_{imn}$) to that of the stencil cell itself. As the average value of the polynomial model is the sum of cell average of each basis functions ($\overline{B}_h(\mathbb{S}_{imn})$) multiplied by the respective coefficient (C_h), the D^{LS} matrix can be constructed using the averaged value for each basis function in the stencil cells, as seen in equation 51.

$$\overline{B}_h(\mathbb{S}_{imn}) = \frac{1}{A(\mathbb{S}_{imn})} \int_{\mathbb{S}_{imn}} B_h \, \partial\mathbb{S}_{imn} \tag{51}$$

Integration can be done in either the physical coordinate system or the auxiliary coordinate system, as linear mapping does not change cell averages.

$\mathbf{D^{LS}}$ Matrix - Integration in 1D cells: For the one-dimensional cells, integration is done analitacilly.

$\mathbf{D^{LS}}$ Matrix - Integration in triangular cells: Integration in triangular elements is done through the use of a Gauss-Legendre quadrature. The integration points (δ_g, ζ_g) and weights (W^{GL}) are defined for a unitary rectangle triangle (S_{ret}) and are taken from [1]. One can linearly map those same points into the corresponding ones (ϵ_g, η_g) in any cell in the stencil and fhe values of the basis functions at those points known. The mathematical procedure is described in equation 52.

$$\overline{B}_h(\mathbb{S}_{imn}) = \frac{1}{d(\mathbb{S}_{imn})} \sum_{g=1}^{4} W_g^{GL} \cdot B_h(\epsilon_g, \eta_g) \tag{52}$$

For this thesis, the Gauss-Legendre quadrature points are passed from the unitary rectangle triangle to the auxiliary coordinate system and not to the original coordinate system.

$\mathbf{D^{LS}}$ Matrix - Integration in square elements: Altough integration in square elements within uniform meshes can be done analytically, for futher flexibility of the developed code, the square cells are divided onto two triangles (S_{t1} and S_{t2}) and then each one subjected to the integration method previously described.

$\mathbf{D^{LS}}$ Matrix - Final form: Finally, matrix D^{LS} can be constructed with the integrals of the basis functions on the stencil cells:

$$D_{im}^{LS} = \begin{bmatrix} \overline{B}_1(\mathbb{S}_{im1}) & ... & \overline{B}_{\mathbb{H}}(\mathbb{S}_{im1}) \\ ... & ... & ... \\ \overline{B}_1(\mathbb{S}_{imN_S}) & ... & \overline{B}_{\mathbb{H}}(\mathbb{S}_{imN_S}) \end{bmatrix} \tag{53}$$

4.3.2 $\mathbf{Y^{LS}}$ Vector

Characteristic quantities ($\tilde{Z}$) provide smoother polynomials and, as such, are used as the dependent variables of the data set. In each interface of a cell, the approximated formulation of the either the two-dimensional Euler equations in 21 or the one-dimensional equations in 11 is set.

The approximated right eigenvectors matrix ($\widetilde{\mathbb{X}}_R$) and eigenvalue diagonal matrix ($\widetilde{\Lambda}$) in equation 23 are computed through the use of Roe averaged quantities (denoted by a tilde), which are, in turn, calculated with data from the target cell ($\overline{Q}_L$) and those of the face face neighbouring cell ($\overline{Q}_R$). As an example, the process for specific entahlpy (H) is shown in equation 54.

$$\tilde{H}_{ij} = \frac{\rho_L H_L + \rho_R H_R}{\rho_L + \rho_R} \tag{54}$$

For each interface, the data contained in each entry of a given stencil must be converted to the respective characteristic variables. The methodology is shown in equation 55.

$$\overline{Z}(\mathbb{S}_{imn}) = \tilde{R}_{ij} \overline{Q}(\mathbb{S}_{imn}) \tag{55}$$

The dependent variables vector (Y) of the Least-Squares Problem is given by equation 56.

$$Y_{imjk} = \begin{bmatrix} \overline{Z}_k(\mathbb{S}_{im1}) - \overline{Z}_k(S_i) \\ ... \\ \overline{Z}_k(\mathbb{S}_{imNs}) - \overline{Z}_k(S_i) \end{bmatrix} \tag{56}$$

The process here described is also used in [4].

4.3.3 Weight Diagonal Matrix ($\mathbf{W^{LS}}$)

The use of a weight diagonal matrix is justified in [4], [5] and in section 3. The weight function used is presented in equation 36, in general form, and is described by equation 57 for the two-dimensional scenario. The one for the one-dimensional case is analogous.

$$W_{imnn}^{LS} = \frac{1}{\left(\epsilon(\mathbb{S}_{imn})^2 + \eta(\mathbb{S}_{imn})^2\right)^{\frac{p}{2}}} \tag{57}$$

4.4. WENO Polynomial Model

After obtaining the different polynomial models for a certain characteristic variable, the complete polynomial model in the respective cell and face results from a convex combination of the polynomial coefficients of those same polyomial models. As WENO schemes aim to avoid polynomials that exhibit the Gibbs phenomen, the weight given to each polynomial is a function of how much they oscillate. The WENO full polynomial model (M^{CW}) can be expressed by equation 58.

$$M_{ijk}^{CW} = W_{ijk1}^{CW} \cdot M_{ijk1} + W_{ijk2}^{CW} \cdot M_{ijk2} + ... \quad (58)$$

The non-linear weight function (W^{CW}) is defined in equation equation 59.

$$W^{CW} = \frac{\Gamma}{\sum_{m=1}^{3||4||5} \Gamma_e}; \quad \Gamma = \frac{w}{(\iota + SI)^r} \quad (59)$$

Concerning equation 59, w is the linear weight of the stencil. Central stencils are normally given a higher value ($w = 0^3$) as they are more accurate in smoother areas of the domain (see [4]). For the directional stencils it takes the value of unity ($w = 1$). SI is the smoothness indicator of the polynomial (see [4]) and is computed in accordance with equation 60. The value of ι is usually very small ($\iota = 10^{-5}$) and is used to avoid divison by zero.

$$SI = \sum_{i=1}^{\wp} \int_{S_{st}} \left(\frac{d^i}{dx_t{}^i} \cdot M(\epsilon, \eta)\right)^2 \partial S_{st} \quad (60)$$

4.5. Face Flux Calculation

After obtaining the complete WENO polynomial models for a said face, the residual of each target cell may be computed. Numerical flux integration on every face can be done on the auxiliary coordinate system of said target cell and then passed to the original coordinate system.

As all target cells get their faces tranformed into the ones in the standard cells, it is possible to, without too much effort, analytically compute the required integrals. However, as the use of GLQ methods provides code that is both simpler and more flexible for future applications, this route was chosen.

Computing the flux integral depends on knowing the original euler variables (Q) at the GLQ points. As such, the full WENO polynomials for the characteristic quantities (Z) can be used to approximate the actual characterictic quantities at those same points, as demonstrated in equation 61.

$$Z_k(\epsilon(s_g), \eta(s_g)) \approx M_{ijk}^{CW}(\epsilon(s_g), \eta(s_g)) \quad (61)$$

The approximate characteristic quantities can then be used to calculate the original Euler quantities, through equation 62.

$$Q(\epsilon(s), \eta(s)) \approx \widetilde{R}_{ij} \cdot Z((\epsilon(s), \eta(s))) \quad (62)$$

For the same GLQ point, two different values for the same quantity are calculated, coming from the WENO polynomials from the cells that share the interface (Q_L and Q_R). A Riemann problem is set in each integration point and the HLLC approximate solver used to calculate the flux on the integration point:

$$F^n(\epsilon(s_g), \eta(s_g)) \approx HLLC(Q_L, Q_R) \quad (63)$$

The resultant flux can then be passed to the original coordinate system and the residual for each cell on the current time-step calculated.

4.6. Flux limiting for the 2D-WENO scheme

In some test cases, neither having different directional stencils or approximating characteristic variables was enough to ensure non-oscillatory behaviour. Simple flux limiting strategies similar to MOOD [2] were implemented on this thesis. The quantities extrapolated on the face are checked for positivity and monotonicity. If the previous conditions are not fulfilled, the quantities on the face (Q_L and Q_R) are extrapolated using a first-order Godunov scheme, for the two-dimensional schemes, and a MUSCL scheme with a min-mod limiter for the one-dimensional schemes. A toggle function (t_w) is used. Both are monotonicity preserving. Equation 64 shows both methodologies.

$$\begin{aligned} Q^{1D} &= Q^{MUSCL} + t_w(Q^{WENO} - Q^{MUSCL}) \\ Q^{2D} &= Q^{Godunov} + t_w(Q^{WENO} - Q^{Godunov}) \end{aligned} \quad (64)$$

4.7. Time integration

The time integration used was the third-order Runge-Kutta method, presented in [4] and expressed by equation 65.

$$\overline{Q}(S_i, t_0 + \Delta t) = \overline{Q}(S_i, t_0) + \frac{\Delta t}{A(S_i)} \cdot RK^4(Q, t) \quad (65)$$

5. Results

5.1. 1D: Sod's Shock Tube

As the developed WENO schemes need to be sufficiently robust for handling shock waves, the Sod's Shock Tube test case was implement. The ruling differential equations are the Euler set presented in 2.1 with the initial ($t = 0s$) point-wise values of the Euler variables defined in equation 66.

$$Q(x,y) = \left[\rho;\ \rho u;\ E\right] = \begin{cases} \left[1;\ 0;\ 2.5\right], & x \leq 0.5 \\ \left[0.1;\ 0;\ 0.125\right], & x > 0.5 \end{cases} \tag{66}$$

The WENO methodology in conjunction with the use of local characteristic variables was used to solve Sod's Shock tube test case at $t = 0.2s$. To evaluate if the proposed WENO scheme could provide advantageous levels of accuracy, the third, fourth, fifth and seventh-order schemes were implemented and compared with a first-order Godunov and a MUSCL scheme with a min-mod slope limiter. Both were taken from [3].

In figure 3, the L_2 norm for the error in cell averaged density at $t = 0.2s$ is presented as a function of average cell size (d_{ref}) and solver run-time (SRT).

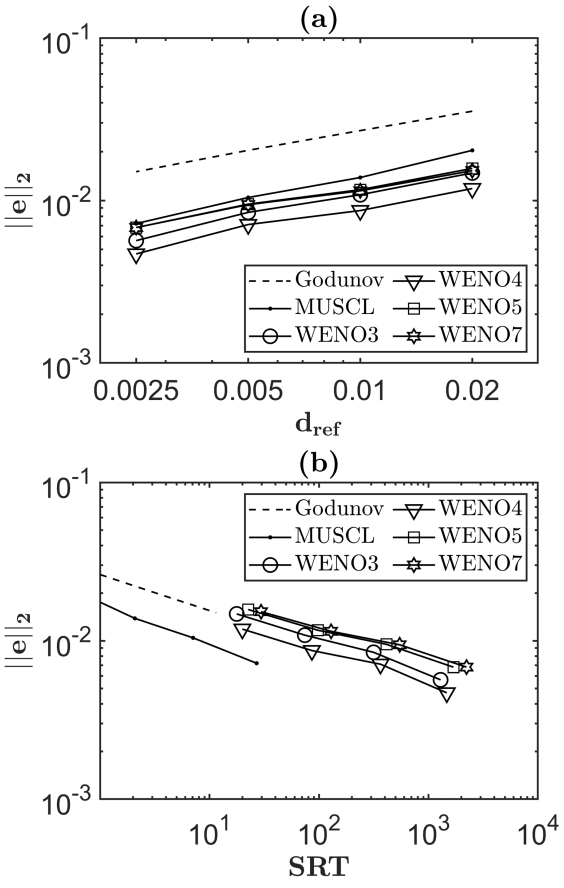

Figure 3: L_2 norm for the error in cell averaged density at $t = 0.2s$ as a function of (a) average cell size (d_{ref}) and (b) solver run-time (SRT) (without flux limiting).

All WENO schemes provided better accuracy than the MUSCL and Godunov schemes. The fourth-order WENO scheme provided the best overall accuracy, followed by the third-order WENO scheme. In terms of computational efficiency. the Godunov scheme surpassed all but the fourth-order WENO scheme. The MUSCL scheme provided the best computational efficiency.

The non-oscillattory behaviour of the computed cell averaged density at $t = 0.2s$ for the fourth-order WENO scheme, which the best results, can be seen in figure 4.

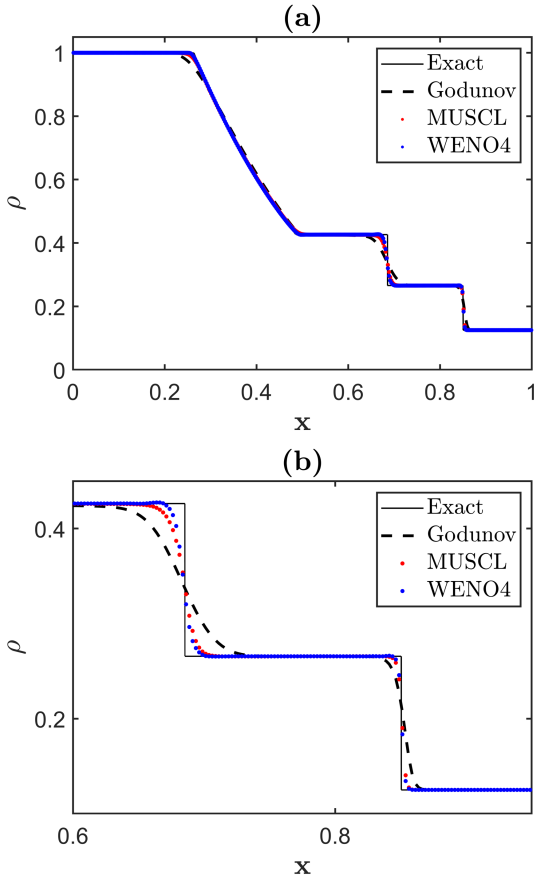

Figure 4: Density as a function of the x-coordinate for the Godunov, MUSCL and flux limited fourth-order WENO schemes at $t = 0.2s$ with (b) detail over the contact discontinuity and shock wave.

5.2. 2D: Supersonic Forward Facing Step

The Supersonic Forward-Facing Step test case consists of an inviscid supersonic flow (Mach=3) over a forward facing step. In this section, only regular cells will be used to obtain results for the fourth-order WENO scheme as triangular cells provided unsatisfatory results for the other test cases. The necessary flow properties are prescribed by equation 67, as well as the inlet flow quantities, at the left boundary of the domain.

$$\gamma(x,y) = 1.4; \quad R(x,y) = 0.714; \quad p(0,y) = 1Pa;$$
$$T(0,y) = 1K; \quad u_x(0,y) = 3ms^{-1}; \quad u_y(0,y) = 0; \tag{67}$$

The initial quantities inside the domain were the same as the inlet ones. The outlet received a supersonic outflow boundary condition (null Neumann boundary condition in x). The boundary conditions of every other boundary of the domain were all reflective (symmetry planes). The developed code was left to iterate until it reached $t = 0.5s$ for the fourth-order WENO, Godunov and MUSCL schemes. The time-step used was again set by achieving temporal convergence for the Godunov scheme. The mesh used was a regular square mesh with significant refinement ($\Delta x = \Delta y = \frac{1}{60}$). In figure 5, thirty density countours are shown for the Godunov, MUSCL and fourth-order WENO. The reference solution at $t = 0.5s$ was taken from [6].

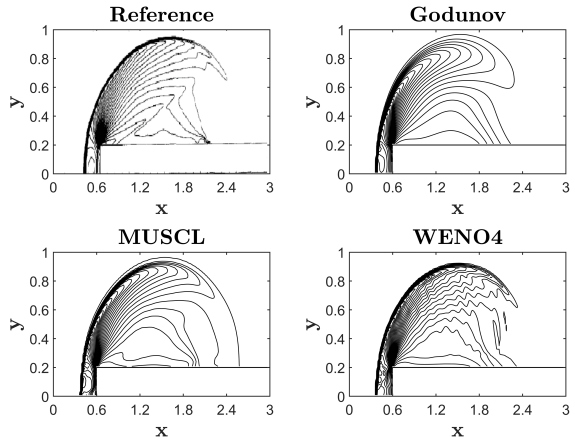

Figure 5: Density contours for the SFFS test case at $t = 0.5s$.

From the results on figure 5 it becomes apparent that the fourth-order WENO scheme offered a major resemblance to the reference solution, depicting the thickness of the weak shock that forms on the left face of the step and the expansion waves that emanate fthe left step corner very closely. The kink in the density contours at the top right of the "buble" is only depicted by the WENO scheme. The wiggles on the density contours for the expansion region are far better depicted. By contrast, the Godunov and MUSCL schemes provided very smooth density contours and the thickness of the shock is depicted mediocrely. The interaction between the expansion fan and the shock is also not as sharp.

6. Conclusions

Altough the objective of bringing arbitrary high-order accuracy to discontinuity-ridden solutions was met, it was only accomplished through the use of computationally expensive characteristic quantities. WENO methodology was also found to be sensitive to the type of mesh employed as the use of triangular cells provided lackluster results. The behaviour of stencil search areas for triangular cells near discontinuities and shock contributed for these unimpressive results. The developed WENO schemes provided better results, accuracy-wise, than the two commercially popular schemes, Godunov and MUCL, for all the test cases run. They did, however, provide worse computational efficiencies. The analysis of the numerical results obtained in this work also seemed to suggest that for real-world applicability, the desired order of accuracy for a WENO scheme should be limited to fourth or lower.

Acknowledgements

I would like to thank my family, friends and supervisor Dr. Duarte Albuquerque for the intellectual and emotional means to conclude this work.

References

[1] J. E. Akin. *Finite element analysis with error estimators : an introduction to the FEM and adaptive error analysis for engineering students.* Elsevier/Butterworth-Heinemann, England, 2005.

[2] P. S. Farmakis, P. Tsoutsanis, and X. Nogueira. Weno schemes on unstructured meshes using a relaxed a posteriori mood limiting approach. *Computer Methods in Applied Mechanics and Engineering*, 363:112921, 2020.

[3] E. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, pages 163–212. 03 2009.

[4] P. Tsoutsanis, V. Titarev, and D. Drikakis. Weno schemes on arbitrary mixed-element unstructured meshes in three space dimensions. *Journal of Computational Physics*, 230(4):1585–1601, 2011.

[5] A. G. Vasconcelos, D. M. Albuquerque, and J. C. Pereira. A very high-order finite volume method based on weighted least squares for elliptic operators on polyhedral unstructured grids. *Computers Fluids*, 181:383–402, 2019.

[6] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115–173, 1984.