

Copyright
by
Jamie Micayla Pool
2015

The Dissertation Committee for Jamie Micayla Pool
certifies that this is the approved version of the following dissertation:

**A Quadrature Eulerian-Lagrangian WENO Scheme
for Reservoir Simulation**

Committee:

Todd Arbogast, Supervisor

Matthew Balhoff

Thomas Chen

Oscar Gonzalez

Alexis Vasseur

Mary Wheeler

**A Quadrature Eulerian-Lagrangian WENO Scheme
for Reservoir Simulation**

by

Jamie Micayla Pool, B.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2015

In loving memory of Bill and Mary Ann Pool,
who inspire me to live with a sense of purpose and wonder.

Acknowledgments

First and foremost, I would like to extend my deepest gratitude to my advisor Todd Arbogast. I am fortunate to have an advisor who is not only patient, understanding and generous but who also cares about my development as a professional researcher. His vast knowledge and thoughtful guidance has enriched my studies and made this a rewarding journey.

I would also like to thank the Mathematics Department at the University of Texas at Austin for providing me with the opportunity to explore my chosen field and for the support they provided. A special thanks to Matthew Balhoff, Thomas Chen, Oscar Gonzalez, Alexis Vasseur, and Mary Wheeler for serving on my dissertation committee. I am also grateful for the collaboration with Cheih-Sen Huang which directly influenced the development of ideas contained in this dissertation. I would like to express my appreciation to Connie Baxter, Sandra Catlett, Bjorn Engquist, Mika Juntunen, Dan Knopf, Kui Ren, Richard Tsai and Karen Uhlenbeck for their many contributions over the course of my graduate studies.

This work was completed with generous support from KAUST through the Academic Excellence Alliance with UT Austin, and the Center for Subsurface Modeling at the Institute for Computational Engineering and Sciences.

Finally, I would like to thank all my friends and family for being sources

of joy and laughter. Their kindness lifts my spirits and I am deeply honored to have them in my life. My heartfelt appreciation goes to my parents, Jennifer Quick and Michael Pool, who have always encouraged me to follow my dreams and without whom none of this would be possible.

A Quadrature Eulerian-Lagrangian WENO Scheme for Reservoir Simulation

Publication No. _____

Jamie Micayla Pool, Ph.D.
The University of Texas at Austin, 2015

Supervisor: Todd Arbogast

This dissertation focuses on solving the advection problem with the motivation of simulating transport in porous media. A quadrature based Eulerian-Lagrangian scheme is developed to solve the nonlinear advection problem in multiple spatial dimensions. The scheme combines the ideas of Lagrangian traceline methods with high order WENO reconstructions to compute the mass that flows into a given cell over a time step. These schemes are important since they have a relaxed CFL constraint, and can be run in parallel.

In this thesis we provide two improvements to Eulerian-Lagrangian schemes. To do this an integration based WENO (IWENO) interpolation technique is derived by reconstructing the primitive function and differentiating. This technique gives a high order reconstruction of the mass at an arbitrary point.

This WENO scheme is used to solve the linear advection problem. A scheme is derived by backwards tracing of quadrature points located on mesh elements. The mass at these tracepoints is used to compute the mass in the trace region, without resolving its boundary. This process defines a high order quadrature Eulerian-Lagrangian WENO (QEL-WENO) scheme that solves the multi-dimensional problem without the need for a spatial splitting technique.

The second improvement is for solving the nonlinear advection problem using an approximate velocity field. The velocity field is used to transport mass in the manner of a standard Eulerian-Lagrangian scheme. Then a flux correction is applied to compute the flow across the tracelines. The contribution is to use a variation of the IWENO technique to reduce the stencil size of this computation.

Numerical results are presented demonstrating the capabilities of the scheme. An application to two-phase flow in porous media is provided.

Table of Contents

Acknowledgments	v
Abstract	vii
List of Tables	xii
List of Figures	xiii
Chapter 1. Introduction	1
1.1 Problem	2
1.2 Dissertation Outline	4
Chapter 2. Integration-based WENO (IWENO)	6
2.1 History of WENO Schemes	6
2.1.1 Review of WENO Interpolation	7
2.2 IWENO Scheme	10
2.2.1 Implementation Details	12
2.2.2 Rounding Errors	17
2.2.3 Non-Linear Weighting	18
2.2.4 Numerical Results in 1D	21
2.2.5 Reconstruction in Higher Dimensions	23
2.2.6 Numerical Results in 2D	25
Chapter 3. A Quadrature Based Eulerian-Lagrangian Scheme for Linear Advection	27
3.1 History of Characteristic Methods	27
3.2 Quadrature Based Scheme	30
3.2.1 Jacobian Matrix	32
3.2.2 Local Conservation	34

3.2.3	Algorithm	37
3.2.4	Numerical Results in 1D	38
3.2.4.1	Example: Constant Speed Transport	39
3.2.4.2	Example: Time Dependent Velocity	41
3.2.4.3	Example: Spatially Dependent Velocity	41
3.2.4.4	Example: Shu’s Linear Test	43
3.2.5	Numerical Results in 2D	45
3.2.5.1	Example: Constant Speed Transport	46
3.2.5.2	Example: Rigid Body Rotation	46
3.2.5.3	Example: Swirling and Deforming Velocity	47
Chapter 4. EL-WENO for Nonlinear Advection in 1D		50
4.1	CWENO Flux Stencil	51
4.2	A Flux Corrected EL-WENO Scheme	54
4.2.1	Mass Reconstruction	57
4.2.2	Flux Correction	60
4.2.3	The Relaxed CFL Condition	65
4.3	Numerical Results	66
4.3.0.1	Constant Speed Transport	67
4.3.0.2	Time Dependent Velocity	68
4.3.0.3	Space Dependent Velocity	69
4.3.0.4	Burgers’ Equation	69
Chapter 5. QEL-WENO for Nonlinear Advection in 2D		73
5.1	The Space-Time Region, $\mathcal{E}(v)$	73
5.2	The Flux Corrected Problem	74
5.2.1	Surface Parameterization	75
5.2.2	Quadrature	77
5.3	Numerical Results	78
5.3.0.1	Example: Swirling Deforming Flow	78
5.3.0.2	Example: Burgers’ Equation	80

Chapter 6. Two Phase Flow	82
6.1 Global Pressure Formulation	83
6.1.1 Pressure Equation	83
6.1.2 Saturation Equation	84
6.2 Discretization	84
6.2.1 Discretization of Flow	85
6.2.2 Discretization of Transport	85
6.3 Numerical Example	86
Chapter 7. Conclusions and Future Work	90
7.1 Conclusions	90
7.2 Future Work	92
7.2.1 Conservation	92
7.2.2 CWENO3	92
7.2.3 Applications	92
Bibliography	94
Vita	107

List of Tables

2.1	Rounding Errors	18
2.2	IWENO Reconstruction	22
2.3	IWENO Midpoint Reconstruction	22
2.4	IWENO 2D Reconstruction	26
3.1	Runge-Kutta Butcher Tableau	38
3.2	Quadrature Rules	40
3.3	Constant Transport Test 1D	42
3.4	Time Dependent Transport Test 1D	43
3.5	Spatially Dependent Transport Test 1D	44
3.6	Constant Transport Test 2D	46
3.7	Rigid Body Rotation Test 2D	47
3.8	Swirling and Deforming Flow 2D	49
4.1	Time Dependent Transport Test 1D	68
4.2	Spatially Dependent Transport Test 1D	70
4.3	Burgers' Equation	71
5.1	Flux Corrected Swirling and Deforming Flow	80

List of Figures

2.1	WENO Reconstruction Polynomials	8
2.2	Third Order WENO Midpoint Reconstruction	9
2.3	IWENO Reconstruction Polynomials	11
2.4	IWENO Nodal Values.	12
3.1	Eulerian-Lagrangian Scheme	31
3.2	Shu's Linear Test 1D	45
3.3	Swirling Deforming Flow 2D	48
4.1	Central WENO Scheme	52
4.2	Implicit Stencil for CWENO5 Flux Computation	53
4.3	Flux Corrected EL-WENO Scheme	56
4.4	Relaxed CFL Constraint	66
4.5	Shu's Linear Test	67
4.6	Burgers' Equation: Example 1	70
4.7	Burgers' Equation: Example 2	72
5.1	Flux Corrected Swirling and Deforming Flow	79
5.2	Flux Corrected 2-D Burgers'	81
6.1	Permeability and Flow	87
6.2	Brooks-Corey Model	88
6.3	Two-Phase Transport Solution	89

Chapter 1

Introduction

We consider the basic physical phenomena of fluid transport in a velocity field. This type of problem naturally arises in a diverse variety of fields such as continuum mechanics and thermodynamics. While the range of application areas of this dissertation are broad, we will restrict our focus to problems arising in porous media simulation. The primary goal is to look at transport of a miscible tracer and of two phases, oil and water, in a petroleum reservoir system.

For the reservoir problem, the porous media are natural materials such as rocks, soils and sands. These media are characterized by porosity, which measures the void space in the material, and permeability, which measures the ability of a fluid to flow through the void network. Reservoir systems have large spatial scales, with dimensions in terms of kilometers and potentially large time scales. For instance, the geological process of CO₂ sequestration lasts for hundreds of years. For this reason, computational simulation of these phenomena are run on modern parallel supercomputers, and hence we seek to develop numerical algorithms that are suitable in this setting.

1.1 Problem

One of the simplest cases for transport in a porous media is the situation with one substance, the tracer, being transported in another. Conservation of the tracer mass gives

$$\frac{\partial(\phi c)}{\partial t} + \nabla \cdot (cu - D\nabla c) = q_c(c), \quad (1.1)$$

where the unknown tracer concentration $c(x, t)$ is transported in a Darcy velocity field $u(x, t)$. The parameter $\phi(x)$ is the porosity of the medium, D is a tensor describing the molecular diffusion and macroscopic dispersion of the particles based on the pore scale geometry of the medium and $q_c(c)$ is an external source or sink function, possibly describing a well in a reservoir.

This problem contains a parabolic part describing mixing through diffusion; this process smooths data. Equations of parabolic type have an infinite speed of information propagation, so implicit numerical methods must be used. There is also an advective or hyperbolic part, describing particle movement in the velocity field. In the presence of a smooth velocity field and initial condition, this equations permits the development of discontinuous solutions, or shocks. Advection equations propagate information at finite speed and explicit solvers may be used. The differences between the advection and diffusion processes, makes it natural to use a splitting technique to solve for them individually. The first step to solve (1.1) using a splitting technique is to solve the advection equation

$$\frac{\partial(\phi c)}{\partial t} + \nabla \cdot (cu) = q_c(c). \quad (1.2)$$

Its solution is then used as an initial condition to solve the diffusion equation

$$\frac{\partial(\phi c)}{\partial t} - \nabla \cdot (D \nabla c) = 0. \quad (1.3)$$

For the purpose of this dissertation, we will focus on the advection equation. However, there are many numerical solvers for the diffusion equation, including ones based on finite element [12], mixed [13, 33] and discontinuous Galerkin (DG) [2] methods.

The various types of methods to solve the advection problem can be categorized by the type of mesh that is used. Eulerian methods, or fixed grid methods, look to accurately reconstruct the flux across the boundaries of the elements. These methods have had a good deal of success as they are accurate while minimizing numerical dispersion errors. Examples of these schemes include weighted essentially non-oscillatory (WENO) [41, 42, 49, 54, 60] Godnov [15, 27, 50] and DG [7, 65]. These methods all must satisfy the Courant Friedrichs Levy (CFL) condition which states that the length of a time step size is limited by the mesh spacing divided by the maximal magnitude of the velocity field. This condition can require a large number of time steps to be taken, leading to increased computation time and the accumulation of numerical dispersion errors over time.

A Lagrangian method, or a moving grid method, attempts to alleviate the CFL constraint by allowing the grid points to move with the velocity field over time. These types of methods are challenging to extend to higher dimensions where multiple shock interaction provides a challenge to mesh adaptabil-

ity. An example of this type of method is given in [73] by Stockie, Mackenzie and Russell. In order to concentrate grid points in optimal locations, they employ a technique that couples a moving mesh equation with the physical equation.

The final type of method is Eulerian-Lagrangian, or Semi-Lagrangian, where at every time step there is a projection between and Lagrangian and Eulerian frameworks. This type of method was first made viable by Douglas and Russell in 1982 with their development of the Modified Method of Characteristics (MMOC) [32]. Further discussion on the history of these types of methods is provided in Chapter 3.

1.2 Dissertation Outline

The dissertation is organized as follows. In Chapter 2, we give the background of the development of WENO schemes. Followed by a detailed development of a new integration based WENO reconstruction technique, with numerical results illustrating formal convergence rates.

Chapter 3 provides a review of characteristic methods for advection problems. We then define a new Eulerian-Lagrangian scheme for the linear advection problem which has the capability of solving the multi-dimensional problem without the aid of a spatial splitting technique.

In Chapters 4 and 5, we define a new algorithm for solving the non-linear advection problem. This algorithm has a compact local stencil for com-

puting the flux integral. Numerical results demonstrating the capabilities of the scheme are presented.

An application to two phase flow is presented in Chapter 6.

This dissertation is finalized in Chapter 7 by giving some conclusions and suggesting directions for future research.

Chapter 2

Integration-based WENO (IWENO)

2.1 History of WENO Schemes

Essentially Non-Oscillatory (ENO) schemes were introduced in 1987 in a paper by Harten, Enquist, Osher and Chakravarthy [41] as a way to develop a shock capturing method to solve hyperbolic conservation laws. The design of these schemes involves construction of multiple polynomials over select stencils. These polynomials have average values on grid intervals that agree with the cell averaged data coming from the scheme at the given time step. The ENO scheme adaptively chooses the polynomial on a stencil that avoids shocks in the data for use in the forward time stepping scheme.

Weighted Essentially Non-Oscillatory (WENO) schemes were later developed in [49, 54] using a convex combination of all the candidate stencils as opposed to the single stencil used in ENO schemes. These schemes achieve a higher order of accuracy in smooth regions, while in regions of discontinuity weight towards the same stencil as the ENO scheme would adaptively choose. In addition WENO schemes are computationally preferable to ENO schemes since they rely on a linear combination of data, instead of the logic statements in ENO schemes. This decreases the total computation time to solve

the problem.

There is an extensive literature detailing the many uses of ENO/WENO schemes for both conservation laws and Hamilton-Jacobi equations. Topics discussed include the use of ENO/WENO schemes in higher dimensions, on triangular grids and theory on stencil choice, stability, and choice of basis functions which can be wavelet, trigonometric or polynomial. In 1997 Shu wrote a set of self contained lecture notes [71] which serve as a more in-depth overview of ENO and WENO schemes.

2.1.1 Review of WENO Interpolation

Our interest in WENO schemes is in the interpolation process. To this end we give a brief overview of the typical WENO reconstruction scheme. We are interested in computing $u(x^*, t^n)$, the value of u at the point x^* located in the grid cell I_0 , given that we have the average value of u on the cells. We are given n , the degree of the lower order polynomial used in the approximation. We take a computational grid with points $x_{-n} < \dots < x_{n+1}$, with corresponding intervals $I_i = [x_i, x_{i+1}]$ for $i = -n, \dots, n$. We further assume that we have cell averaged data for a function $u(x)$, given by

$$\bar{u}_i = \frac{1}{x_{i+1} - x_i} \int_{I_i} u(x) dx, \quad i = -n, \dots, n. \quad (2.1)$$

In order to approximate the value of $u(x^*)$ we define $n+1$ polynomials, p_0, \dots, p_n of degree n so that they agree with the cell averaged data, i.e.,

$$\int_{I_i} p_k dx = (x_{i+1} - x_i) \bar{u}_i, \quad i = -n + k, \dots, k, \quad (2.2)$$

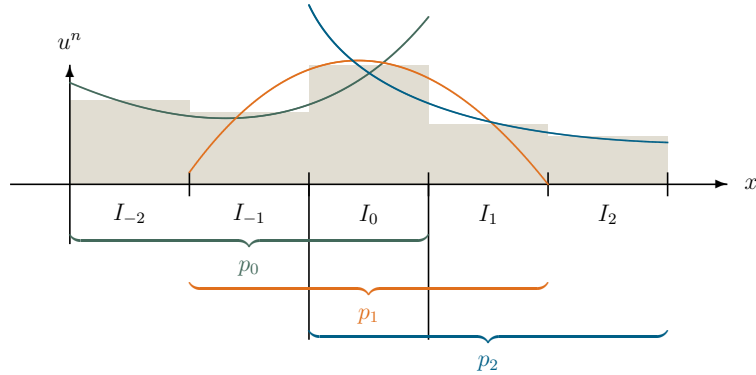


Figure 2.1: **WENO Reconstruction Polynomials.** Illustration of WENO reconstruction polynomials $p_k(x)$ which are used to approximate the value of u at a point $x^* \in I_0$.

as shown in Figure 2.1. Taylor series analysis shows that for smooth $u(x)$ the approximation $p_k(x^*)$ is an approximation of $u(x^*)$ of order $n + 1$, i.e.,

$$p_k(x^*) - u(x^*) = O(h^{n+1}), \quad (2.3)$$

where $h = \max |x_{i+1} - x_i|$. There is also a polynomial of degree $2n$ defined with cell averages agreeing with all the candidate cells that is a higher order approximation to $u(x^*)$. WENO schemes attempt to find weights c_k such that the linear combination of the polynomials and weights achieves this higher order

$$\sum_{k=0}^n c_k p_k(x^*) - u(x^*) = O(h^{2n+1}). \quad (2.4)$$

However, there are situations where such weights cannot be found. A well known example of this is found in attempting to derive a 3rd order central WENO scheme. Here the reconstruction point x^* is the center of the interval I_0 . As seen in Figure 2.2, the linear reconstructions agree at this point.

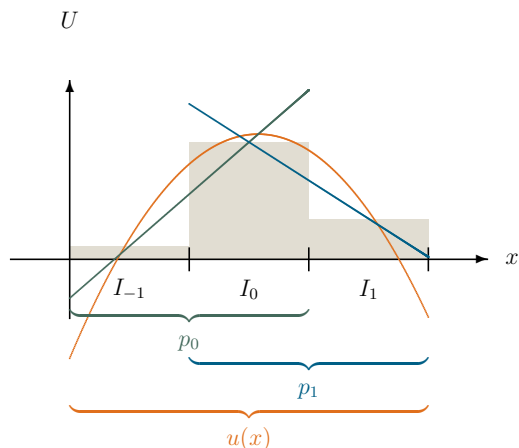


Figure 2.2: **The Third Order WENO Midpoint Reconstruction.** The linear interpolation polynomials p_k have the same value at the midpoint.

Since any linear combination of these values will yield the same value at the midpoint, there does not exist coefficients for the linear combination to agree with the quadratic interpolation. Several different techniques have been used to get around this difficulty. Levy, Puppò and Russo [53] use the quadratic polynomial in addition to the linear interpolations in order to achieve this higher order approximation in smooth regions. Huang, Arbogast and Hung [47] define a re-averaging technique which converts cell averages on one grid to cell averages on another. The re-averaging puts the midpoint on a grid boundary where linear weights are guaranteed to exist. However, this scheme contains a large implicit stencil. This technique will be further addressed in Chapter 4.

For the transport schemes we will develop later, we need a high order approximation at any point x^* in our mesh. We also want to avoid using a larger stencil, since this can lead to numerical diffusion in the transport

scheme. The rest of this chapter will derive an Integration-based WENO (IWENO) scheme which addresses this problem.

2.2 IWENO Scheme

We are motivated by the work of Carlini, Ferretti and Russo in [19], where they show that for problems involving nodal interpolation, as opposed to cell average values, high order reconstructions can be found for any point in the domain. Furthermore, the weighting functions are simply polynomials of the position x^* .

We define polynomials P_k of degree $n + 1$ that interpolate nodal values of the primitive function,

$$U(x) = \int_{x_0}^x u(\xi) d\xi. \quad (2.5)$$

This interpolation is shown in Figure 2.3. Results from [19] state that there are coefficient polynomials $c_k(x)$ of degree $n - 1$ such that for $x^* \in I_0$ there is a high order reconstruction of the primitive function, i.e.,

$$U(x) - P_k(x) = O(h^{r+1}), \quad \forall x \in I_0, \quad (2.6)$$

$$U(x) - \sum_{k=0}^n c_k(x) P_k(x) = O(h^{2r}), \quad \forall x \in I_0, \quad (2.7)$$

$$\sum_{k=0}^n c_k(x) = 1. \quad (2.8)$$

Here $r + 1 = (n + 1) + 1$ is the order of polynomial interpolation of the primitive function.

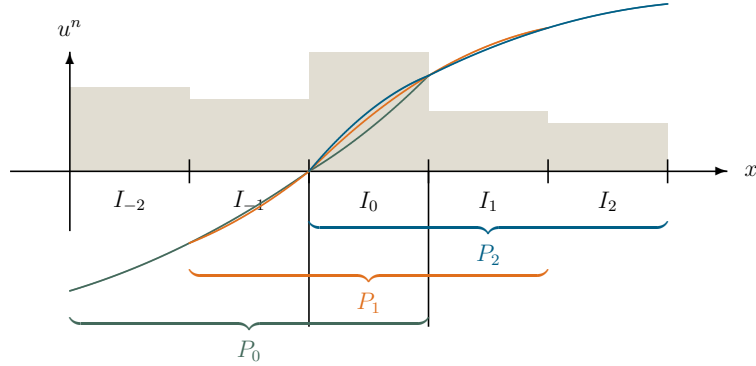


Figure 2.3: **IWENO Reconstruction Polynomials.** Illustration of IWENO reconstruction polynomials $P_k(x)$ which are used to approximate the value of the primitive function $U(x)$ at a point $x^* \in I_0$.

The Lagrange interpolation satisfies

$$U(x) = P_k(x) + \frac{U^{(r+1)}\gamma}{(r+1)!} \prod_{j=k-r+1}^{k+1} (x - x_j) \quad (2.9)$$

for some $\gamma(x) \in (x_{-k-r+1}, x_{k+1})$. If $u(x) \in C([x_{-k+r+1}, x_{k+1}])$ then for any $x_{-k-r+1} \leq \alpha < \beta \leq x_{k+1}$

$$\frac{1}{\beta - \alpha} \int_{\alpha}^{\beta} u(x) dx = \frac{P_k(\beta) - P_k(\alpha)}{\beta - \alpha} + O(h^r). \quad (2.10)$$

Which comes from (2.9) and the mean value theorem. The limit as $\alpha \rightarrow \beta$ gives

$$u(\alpha) = P'_k(\alpha) + O(h^r). \quad (2.11)$$

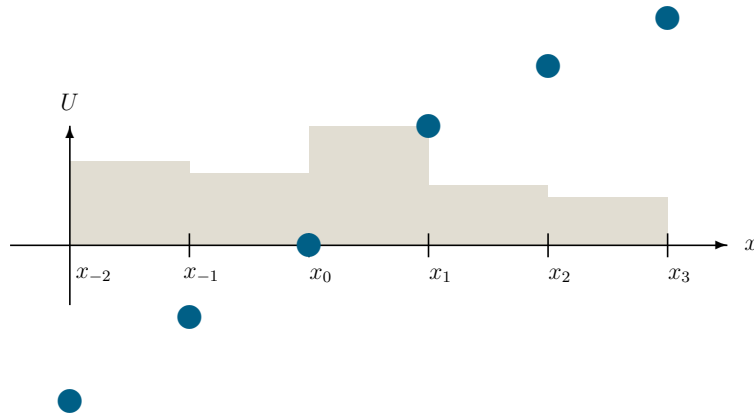


Figure 2.4: **IWENO Nodal Values.** Cell average data is used to compute the value of the primitive function at grid nodes.

Using this relation the differentiation of equations (2.6) - (2.8) gives

$$u(x) - p_k(x) = O(h^r), \quad \forall x \in I \quad (2.12)$$

$$u(x) - \sum_{k=0}^n (c'_k(x) P_k(x) + c_k(x) p_k(x)) = O(h^{2r-1}), \quad \forall x \in I \quad (2.13)$$

$$\sum_{k=0}^n c'_k(x) = 0, \quad (2.14)$$

where $p_k(x)$ is the derivative of $P_k(x)$. Showing the reconstruction of $u(x)$,

$$\sum_{k=0}^n (c'_k(x) P_k(x) + c_k(x) p_k(x)), \quad (2.15)$$

is of order h^{2r-1} . The sum (2.15) is the IWENO reconstruction using linear weighting coefficients.

2.2.1 Implementation Details

To implement this scheme we first need to convert our cell averages to nodal values by using the definition of the primitive function. An illustration

of this process is seen in Figure 2.4. The nodal values of the primitive function are sums of the cell averages scaled by their corresponding element size, so

$$U(x_i) = - \int_{x_i}^{x_0} u(\xi) d\xi = - \prod_{j=i}^{-1} (x_{j+1} - x_j) \bar{u}_j, \quad i < 0, \quad (2.16)$$

$$U(x_0) = \int_{x_0}^{x_0} u(\xi) d\xi = 0, \quad (2.17)$$

$$U(x_i) = \int_{x_0}^{x_i} u(\xi) d\xi = \prod_{j=0}^{i-1} (x_{j+1} - x_j) \bar{u}_j, \quad i > 0. \quad (2.18)$$

In order to compute the polynomials as shown in Figure 2.3, a polynomial interpolation can be evaluated using Lagrange interpolation polynomials. The polynomial P_k is defined on the stencil x_{-n+k}, \dots, x_{1+k} and is reconstructed as

$$P_k(x) = \sum_{i=-n+k}^{1+k} U(x_i) \mathcal{L}_{(k,i)}(x), \quad (2.19)$$

$$\text{where } \mathcal{L}_{(k,i)}(x) = \prod_{\substack{j=-n+k \\ j \neq i}}^{j=1+k} \frac{(x - x_j)}{(x_i - x_j)}. \quad (2.20)$$

The Lagrange interpolation polynomials, $\mathcal{L}_{(k,i)}$, have values at nodes given by the Kronecker delta, i.e.,

$$\mathcal{L}_{(k,i)}(x_j) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}. \quad (2.21)$$

Differentiation of polynomials P_k gives the polynomials p_k

$$p_k(x) = \frac{dP_k}{dx} = \sum_{i=-n+k}^{1+k} U(x_i) \frac{d\mathcal{L}_{k,i}}{dx}, \quad (2.22)$$

$$\text{where } \frac{d\mathcal{L}_{k,i}}{dx} = \sum_{\substack{m=-n+k \\ m \neq i}}^{1+k} \left(\frac{1}{(x-x_m)} \prod_{\substack{j=-n+k \\ j \neq i}}^{1+k} \frac{(x-x_j)}{(x_i-x_j)} \right). \quad (2.23)$$

Algorithm 1 for the computation of the primitive function and its derivative on an arbitrary grid.

```

P = 0
dP = 0
for i = -n + k, i < 2 + k do

    Lag = Ui
    Denom = 1.0
    for j = 0, j < 4, j ≠ i do
        Lag *= (x - xj)
        Denom *= (xi - xj)
    end for
    P += Lag/Denom

    dLag = 0
    for m = -n + k, m < k + 2, m ≠ i do
        dL = Ui
        for j = -n + k, j < k + 2, j ≠ m, j ≠ i do
            dL *= (x - xj)
        end for
        dLag += dL
    end for
    dP += dLag/Denom

end for

```

Computationally, an algorithm for this process is given by Algorithm

1. However, this is sensitive to rounding errors, when x is near a grid point.

For irregular or adaptive grids using a Newton divided difference algorithm to compute these polynomials minimizes this problem. For uniform meshes computing the coefficients of the polynomials ahead of time reduces this error. The 3rd order reconstruction of $P_k(x)$ and $p_k(x)$ on a grid with uniform spacing h are

$$P_k(x_{loc,k}) = (2U_{-1+k} + x_{loc,k}(-3U_{-1+k} + 4U_k - U_{1+k}) + x_{loc,k}^2(U_{-1+k} - 2U_k + U_{1+k})) / 2, \quad (2.24)$$

$$p_k(x_{loc,k}) = (1/2(3\bar{u}_{-1+k} - \bar{u}_k) - (\bar{u}_{-1+k} - \bar{u}_k)x_{loc,k}) / h, \quad (2.25)$$

$$\text{where } x_{loc,k} = (1 - k) + \alpha \quad \text{and} \quad k = 0, 1$$

and the 5th order scheme has polynomials

$$P_k(x_{loc,k}) = (U_{-2+k} + x_{loc,k}(-11U_{-2+k} + 18U_{-1+k} - 9U_k + 2.0U_{k+1}) + 3x_{loc,k}^2(2U_{-2+k} - 5U_{-1+k} + 4U_k - U_{k+1}) + x_{loc,k}^3(-U_{-2+k} + 3U_{-1+k} - 3U_k + U_{k+1})) / 6, \quad (2.26)$$

$$p_k(x_{loc,k}) = (11\bar{u}_{-2+k} - 7\bar{u}_{-1+k} + 2\bar{u}_k) / 6 - x_{loc,k}(2\bar{u}_{-2+k} - 3\bar{u}_{-1+k} + \bar{u}_k) - x_{loc,k}^2(-\bar{u}_{-2+k} + 2\bar{u}_{-1+k} - \bar{u}_k) / 2) / h \quad (2.27)$$

$$\text{where } x_{loc,k} = (2 - k) + \alpha \quad \text{and} \quad k = 0, 1, 2.$$

These polynomials are evaluated at the local reference position $x_{loc,k}$ defined through a scaling factor $\alpha \in [0, 1]$ describing the relative position of $x^* \in I_0$ namely $\alpha = (x^* - x_0)/h$.

Carlini, Ferretti and Russo [19] give the algorithm to compute the linear weighting coefficients. To do this first set

$$W_k^m = \prod_{j=-n}^{-n+k-1} (x_{m+1} - x_j) \prod_{j=2+k}^{n+1} (x_j - x_{m+1}), \quad (2.28)$$

for $k, m = 0, \dots, n-1$, then solve the lower triangular matrix problem

$$\begin{bmatrix} W_0^0 & 0 & \cdots & 0 \\ W_1^0 & W_1^1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ W_{n-1}^0 & W_{n-1}^1 & \cdots & W_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} \hat{c}_0 \\ \hat{c}_1 \\ \vdots \\ \hat{c}_{n-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \quad (2.29)$$

On a uniform mesh the solution is

$$\hat{c}_k = \binom{n}{k} \frac{n!}{(2n+1)!} h^n. \quad (2.30)$$

Finally, the linear weights are computed from W_k^m as

$$c_k(x) = \hat{c}_k \prod_{j=-n}^{-n+k-1} (x - x_j) \prod_{j=2+l}^{2n-1} (x_j - x) \quad l = 0, 1, \dots, n-1, \quad (2.31)$$

$$c_n(x) = 1 - \sum_{l=0}^{n-1} c_k(x). \quad (2.32)$$

For a scheme with uniform mesh size we compute these terms directly.

For the 3rd order scheme the coefficients are

$$c_0(x(\alpha)) = (2 - \alpha)/3, \quad c'_0(x(\alpha)) = -1/(3h), \quad (2.33)$$

$$c_1(x(\alpha)) = (1 + \alpha)/3, \quad c'_1(x(\alpha)) = 1/(3h). \quad (2.34)$$

The 5th order scheme has coefficients

$$c_0(x(\alpha)) = (6 - 5\alpha + \alpha^2)/20, \quad c'_0(x(\alpha)) = (-5 + 2\alpha)/(20h), \quad (2.35)$$

$$c_1(x(\alpha)) = (6 + \alpha - \alpha^2)/10, \quad c'_1(x(\alpha)) = (1 - 2\alpha)/(10h), \quad (2.36)$$

$$c_2(x(\alpha)) = (2 + 3\alpha + \alpha^2)/20, \quad c'_2(x(\alpha)) = (3 + 2\alpha)/(20h). \quad (2.37)$$

The IWENO scheme in 1D using linear coefficients on a uniform grid is implemented by solving equation (2.15) using polynomials given in equation (2.24) - (2.27) and coefficients listed in equations (2.33) - (2.37).

2.2.2 Rounding Errors

In this section we demonstrate the performance of the IWENO algorithm. We will first show that the algorithm is sensitive to rounding error accumulation. This can be overcome by adjusting the precision of the code, we will outline these changes, and compare numerical results.

In order to test IWENO we use the 5th order reconstruction polynomials. The point x^* and relative location $\alpha \in [0, 1]$ are given. These parameters define the center cell

$$I_0 = [x^* - \alpha\Delta x, x^* + (1 - \alpha)\Delta x], \quad (2.38)$$

where Δx is the size of the interval, and we assume uniform spacing. We perform successive grid refinements $\Delta x = \Delta x/2$ and analyze the errors.

The function reconstructed is $u(x) = \sin(\pi x)$. The reconstruction is computed at the point $x^* = 0.716$ with relative location $\alpha = 0.334$. This test is run with both double and long double precision. The results are in Table 2.1. Double precision should exhibit precision at the 16th digit. However the IWENO scheme uses a process of differentiation, where we scale out by a factor of Δx , this reduces the total number of digits we see for the computation.

For problems where high precision is necessary, using the long double

Δx^{-1}	Error	Order	Δx^{-1}	Error	Order
10	1.49 E-5		10	1.49 E-5	
20	4.52 E-7	5.04	20	4.52 E-7	5.04
40	1.39 E-8	5.03	40	1.38 E-8	5.03
80	4.29 E-10	5.01	80	4.29 E-10	5.01
160	1.33 E-11	5.01	160	1.33 E-11	5.01
320	4.20 E-13	4.99	320	4.16 E-13	5.00
640	4.50 E-14	3.22	640	1.28 E-14	5.02
1280	8.21 E-15	2.45	1280	2.65 E-16	5.59
2560	1.13 E-14	-0.46	2560	2.00 E-16	0.41
5120	5.53E-13	-5.61	5120	1.28E-16	0.63

Table 2.1: **Rounding Errors.** The error in computing $\sin(0.716\pi)$ using the IWENO scheme. The results on the left use double precision, on the right is long double precision.

precision adds the memory necessary. This is implemented in the scheme by changing the precision of variables p_k and P_k to be of type long double, however to minimize total storage memory the grid variable x , α , $c(x)$, and $c'(x)$ are of type double. The numerical results provided were computed in a code that is implemented in C++. Here we note the use of the math library `<cmath>` since it overloads mathematical operations for variable type long double.

2.2.3 Non-Linear Weighting

The IWENO reconstruction detailed previously gives a high order accurate reconstruction technique for smooth data. However when the initial data is not smooth it is preferable to use a polynomial interpolation whose stencil does not contain the shock in the data. WENO schemes adjust the weights of the linear coefficients by an appropriate choice of smoothness indicator, IS.

The non-linear weighting polynomials, $w_k(x)$, are defined as a function of these indicators and the linear polynomials $c_k(x)$ as

$$w_k = \frac{\sigma_k}{\sum_{i=0}^n \sigma_i} \quad \text{with} \quad \sigma_i = \frac{c_i}{(\epsilon + IS_i)^p} \quad \text{for} \quad k = 0, \dots, n. \quad (2.39)$$

Here $p \geq 1$ is an integer and $0 < \epsilon \ll 1$ is a small number to avoid dividing by 0. The smoothness indicators IS_i are actually in some sense ‘roughness indicators’ since larger values should indicate that there is a shock in the i^{th} stencil.

The reconstruction approximation defining the IWENO scheme is

$$u(x) \approx \sum_{k=0}^n (w'_k(x) P_k(x) + w_k(x) p_k(x)). \quad (2.40)$$

The linear coefficients, c_k are chosen such that scheme achieves a high order of accuracy. We will next show that as long as $c_k - w_k = O(h^{r-1})$, we maintain the accuracy presented in equations (2.6) - (2.14).

Theorem 2.2.1. *Let $c_k(x)$, $p_k(x)$, and $P_k(x)$ satisfy equations (2.6) - (2.14) and $w_k(x)$ be such that $c_k - w_k = O(h^{r-1})$, then $U(x) - \sum w_k P_k = O(h^{2r})$ and $u(x) - \sum (w_k p_k + w'_k P_k) = O(h^{2r-1})$.*

Proof. First we consider the primitive function

$$\begin{aligned} U(x) - \sum w_k P_k &= \left(U(x) - \sum c_k P_k \right) + \sum (c_k - w_k) P_k \\ &= O(h^{2r}) + \sum (c_k - w_k) (P_k - U) \quad \text{since} \quad U \sum (c_k - w_k) = U(1 - 1) = 0 \\ &= O(h^{2r}) + O(h^{r-1}) O(h^{r+1}) = O(h^{2r}). \end{aligned}$$

Then considering $u(x)$ we have

$$\begin{aligned}
u(x) - \sum (w_k p_k + w'_k P_k) &= \left(u(x) - \sum (c_k p_k + c'_k P_k) \right) + \sum ((c_k - w_k)' P_k + (c_k - w_k) p_k) \\
&= O(h^{2r-1}) + \sum (c_k - w_k)' P_k + \sum (c_k - w_k) p_k \\
&= O(h^{2r-1}) + \sum (c_k - w_k)' (P_k - U) + \sum (c_k - w_k) (p_k - u)
\end{aligned}$$

since $U \sum (c'_k - w'_k) = U(0 - 0) = 0$ and $u \sum (c_k - w_k) = u(1 - 1) = 0$

$$= O(h^{2r-1}) + O(h^{r-2})O(h^{r+1})_k + O(h^{r-1})O(h^r) = O(h^{2r-1}).$$

□

In the classic paper [49], Jiang and Shu show that choosing smoothness indicators of the form

$$\sum_{l=1}^{r-1} \int_I h^{2l-1} \left(p_k^{(l)} \right)^2 dx \quad (2.41)$$

satisfy this requirement, at least for $r = 2$ or 3 , that is schemes of 3rd and 5th order. For a uniform grid the smoothness indicators for the 3rd order scheme are

$$IS_0 = (\bar{u}_{-1} - \bar{u}_0)^2, \quad (2.42)$$

$$IS_1 = (\bar{u}_0 - \bar{u}_1)^2, \quad (2.43)$$

and for the 5th order scheme they are

$$IS_0 = 4\bar{u}_{-2}^2 + 25\bar{u}_{-1}^2 - 31\bar{u}_{-1}\bar{u}_0 + 10\bar{u}_0^2 + \bar{u}_{-2}(-19\bar{u}_{-1} + 11\bar{u}_0), \quad (2.44)$$

$$IS_1 = 4\bar{u}_{-1}^2 + 13\bar{u}_0^2 - 13\bar{u}_0\bar{u}_1 + 4\bar{u}_1^2 + \bar{u}_{-1}(-13\bar{u}_0 + 5\bar{u}_1), \quad (2.45)$$

$$IS_2 = 10\bar{u}_0^2 + 25\bar{u}_2^2 - 19\bar{u}_1\bar{u}_2 + 4\bar{u}_2^2 + \bar{u}_0(-31\bar{u}_1 + 11\bar{u}_2). \quad (2.46)$$

The IWENO scheme solves equation (2.40) for a uniform grid we use reconstruction polynomials (2.24) - (2.27) with weights defined in (4.33), using coefficients in equations (2.33) - (2.37) and smoothness indicators in equations (2.42) - (2.46).

2.2.4 Numerical Results in 1D

In this section we will briefly demonstrate the performance the of the IWENO algorithm. We proceed with the same set up as in section 2.2.2 using long double precision.

We reconstruct the function $u(x) = \sin(\pi x)$ at the point $x = 0.716$ with the relative location $\alpha = 0.334$ in I_0 . The results for the 3rd and 5th order schemes are given in Table 2.2. These serve to verify that non-linear weighting preserves the order of the scheme.

This example is repeated to illustrate the performance at the midpoint where $\alpha = 0.5$. Numerical results are given in Table 2.3. On the coarse mesh we have smaller errors than when $\alpha = 0.334$. Also, as the mesh size is reduced there is a super convergence effect where our third order scheme converges at fourth order, and our fifth order scheme converges at sixth order. This variability will influence the convergence rate of the transport schemes derived later.

Δx^{-1}	Error	Order	Δx^{-1}	Error	Order
10	7.16 E-4		10	1.49 E-5	
20	8.61 E-5	3.06	20	4.52 E-7	5.04
40	1.05 E-5	3.03	40	1.39 E-8	5.03
80	1.30 E-6	3.02	80	4.29 E-10	5.01
160	1.62 E-7	3.01	160	1.33 E-11	5.01
320	2.02 E-8	3.00	320	4.16 E-13	5.00
640	2.51 E-9	3.00	640	1.28 E-14	5.02
1280	3.14 E-10	3.00	1280	2.65 E-16	5.59
2560	3.92 E-11	3.00	2560	2.00 E-16	0.41
5120	4.91 E-12	3.00	5120	1.28E-16	0.63

Table 2.2: **IWENO Reconstruction.** The error in computing $\sin(\pi x)$ at $x = 0.716$ at a relative cell location of 0.334 using the IWENO scheme. The results on the left are for the 3rd order scheme and on the right is the 5th order scheme.

Δx^{-1}	Error	Order	Δx^{-1}	Error	Order
10	3.53 E-5		10	5.16 E-7	
20	2.22 E-6	3.99	20	8.13 E-9	5.99
40	1.39 E-7	4.00	40	1.27 E-10	6.00
80	8.68 E-8	4.00	80	1.99 E-12	6.00
160	5.42 E-10	4.00	160	3.10 E-14	6.00
320	3.39 E-11	4.00	320	6.51 E-16	5.57

Table 2.3: **IWENO Midpoint Reconstruction.** The error in computing $\sin(\pi x)$ at $x = 0.716$ located at the midpoint of the cell, using the IWENO scheme. The results on the left are for the 3rd order scheme and on the right is the 5th order scheme.

2.2.5 Reconstruction in Higher Dimensions

The purpose of this section is to outline how this algorithm extends to multiple spatial dimensions. To do this we first define the primitive function as

$$U(x) = \int_{x_{1_0}}^{x_1} \cdots \int_{x_{d_0}}^{x_d} u(\xi) d\xi_d \dots d\xi_1, \quad x = (x_1, \dots, x_d). \quad (2.47)$$

The point $x_0 = (x_{1_0}, \dots, x_{d_0})$ is a vertex of the center element. Here elements are of the form

$$E = [x_{1_{i_1}}, x_{1_{i_1+1}}] \times \cdots \times [x_{2_{i_d}}, x_{2_{i_d+1}}], \quad \text{for } i_j = -n, \dots, n \quad \text{and } j = 0, \dots, d.$$

The function $u(x)$ is the multicomponent derivative of the primitive function, i.e.,

$$u(x) = \frac{d}{dx_1} \cdots \frac{d}{dx_d} U(x). \quad (2.48)$$

For indexing simplicity, the remainder of this section will be presented for three spatial dimensions. The extension to higher dimensions is straightforward. The function $U(x)$ is reconstructed as a tensor product of the 1D reconstruction polynomials

$$\begin{aligned} \sum_{k,l,m=0}^n c_k(x_1) P_k(x_1) c_l(x_2) P_l(x_2) c_m(x_3) P_m(x_3) \\ = \sum_{k,l,m=0}^n c_{klm}(x) P_{klm}(x), \end{aligned} \quad (2.49)$$

where the polynomials $c_{klm}(x)$ and $P_{klm}(x)$ are

$$c_{klm}(x) = c_k(x_1) c_l(x_2) c_m(x_3), \quad (2.50)$$

$$P_{klm}(x) = P_k(x_1) P_l(x_2) P_m(x_3). \quad (2.51)$$

The function $u(x)$ is computed by

$$\frac{d}{dx_1} \frac{d}{dx_2} \frac{d}{dx_3} \sum_{k,l,m=0}^n c_{klm}(x) P_{klm}(x), \quad (2.52)$$

using the product rule for differentiation. These are computed using the one dimensional equations given in (2.24) - (2.27) and (2.33) - (2.37).

The nonlinear weighting is defined in the usual way as

$$w_{klm}(x) = \frac{\sigma_{klm}(x_i)}{\sum_{abc} \sigma_{abc}(x)} \quad \text{with} \quad \sigma_{klm}(x) = \frac{c_{klm}(x)}{(\epsilon + IS_{klm})^p} \quad (2.53)$$

The smoothness indicator, IS_{klm} must contain information from the entire polynomial $P_{k,l,m}(x)$ in order to maintain the high order reconstruction. Theorem 2.2.1 generalizes to the following result.

Theorem 2.2.2. *Let $c_{klm}(x)$ and $P_{klm}(x)$ satisfy the following relations*

$$\begin{aligned} U - P_{klm} &= O(\Delta x_1^{r+1} + \Delta x_2^{r+1} + \Delta x_3^{r+1}), \\ U - \sum P_{klm} c_{klm} &= O(\Delta x_1^{2r} + \Delta x_2^{2r} + \Delta x_3^{2r}), \\ \sum_{\gamma=0}^n c_\gamma &= 1, \end{aligned}$$

and $w_{klm}(x)$ be such that the $\sum (c_{klm} - w_{klm}) = O(\Delta x_1^{r-1} + \Delta x_2^{r-1} + \Delta x_3^{r-1})$.

Then

$$\begin{aligned} U - \sum w_{klm} P_{klm} &= O(\Delta x_1^{2r} + \Delta x_2^{2r} + \Delta x_3^{2r}) \\ u - \frac{d}{dx_1} \frac{d}{dx_2} \frac{d}{dx_3} \sum w_{klm} P_{klm} &= O(\Delta x_1^{2r-1} + \Delta x_2^{2r-1} + \Delta x_3^{2r-1}). \end{aligned}$$

Proof. Follow exactly as in the proof for Theorem 2.2.1 . □

The code implements the standard set of smoothness indicators for a WENO scheme. In multiple spatial dimensions these are

$$\text{IS}_{klm} = \sum_{|\alpha|=1}^{r-1} \int_{\text{E}} h^{2|\alpha|-1} \left(p_{klm}^{(\alpha)} \right)^2 dx. \quad (2.54)$$

2.2.6 Numerical Results in 2D

In this section we give an example of the performance of the 5th order IWENO scheme in two spatial dimensions. To do this we define the point (x^*, y^*) that we wish to reconstruct along with a relative locations $\alpha, \beta \in [0, 1]$ inside cell I_0 that (x^*, y^*) is located. I_0 is the rectangular element

$$[x^* - \alpha \Delta x, x^* + (1 - \alpha) \Delta x] \times [y^* - \beta \Delta y, y^* + (1 - \beta) \Delta y]. \quad (2.55)$$

The function we reconstruct is $u(x, y) = \sin(\pi x + \pi y)$. The reconstruction is computed for the point $(x^*, y^*) = (0.489, 0.372)$ at the relative locations $(\alpha, \beta) = (0.123, 0.746)$. We use a uniform grid with $\Delta x = \Delta y$, and analyze the errors as we take successive grid refinements $\Delta x = \Delta x/2$. These are presented in Table 2.4.

Note that the rounding error accumulation for these is more than that for the 1D problem, the reason is taking a multicomponent derivative of the primitive function corresponds with dividing out by Δx twice. If increased precision is desired quadratic precision can be implemented.

Furthermore the results using non-linear weighting can develop oscillations in convergence rates. In the example provided, we see what appears to

Δx^{-1}	Error	Order	Δx^{-1}	Error	Order
10	1.40E-5		10	1.40E-5	
20	3.98 E-7	5.13	20	3.98 E-7	5.13
40	1.17E- 8	5.08	40	1.17E- 8	5.08
80	3.56E- 10	5.04	80	3.56E- 10	5.04
160	1.17 E- 11	4.92	160	1.09 E- 11	5.02
320	1.64 E-12	2.83	320	3.41 E-13	5.00
640	1.54 E-11	-3.23	640	1.37 E-14	4.64

Δx^{-1}	Error	Order	Δx^{-1}	Error	Order
10	7.74 E-4		10	7.74 E-4	
20	3.43E-5	4.50	20	3.43E-5	4.50
40	4.52E-7	6.24	40	4.52E-7	6.24
80	9.04E-10	8.97	80	9.04E-10	8.97
160	1.22E-11	6.21	160	1.15E-11	6.30
320	1.64E-12	2.89	320	3.42 E-13	5.06
640	1.54E-11	-3.23	640	1.36E-14	4.65

Table 2.4: **IWENO 2D Reconstruction.** The error in computing $\sin(\pi x + \pi y)$ using the 5th order IWENO scheme. On the top are results using linear weighting coefficients, the results on the bottom use non-linear weighting. The left side is double precision, and the right is long double precision.

be a super convergence effect, however this is due to the larger errors on the coarse meshes. For the results using the non-linear weighting and long double precision, the convergence rate for points corresponding to $\Delta x^{-1} = 80, \dots, 640$ is 5.31, as found through a linear regression analysis.

Chapter 3

A Quadrature Based Eulerian-Lagrangian Scheme for Linear Advection

3.1 History of Characteristic Methods

We overview the history of the development of characteristic methods to solve the advection problem,

$$u_t + \nabla \cdot f(u) = q. \quad (3.1)$$

This equation models the transport of a conserved quantity u , such as mass, energy or momentum. The fact that u is conserved means that assuming no mass is created or destroyed over the boundaries of the domain Ω that the quantity $\int_{\Omega} u \, dx$ is constant in time. The flux function $f(u)$ describes the local movement of u in the domain and q is an external source term. One method to theoretically analyze conservation equations is to look at the characteristics curves $\xi(t)$. These are solutions of the ordinary differential equation (ODE) $\xi'(t) = f'(u)$. The solution of the advection problem restricted to these characteristic curves is constant, i.e.,

$$\begin{aligned} \frac{d}{dt}u(\xi(t), t) &= \frac{\partial}{\partial t}u(\xi(t), t) + \nabla u(\xi(t), t) \cdot \xi'(t) \\ &= u_t + \nabla \cdot f(u) \\ &= 0. \end{aligned} \quad (3.2)$$

The linear advection problem is defined by setting $f(u) = v(x, t) u$. In this case the characteristics curves are the same as the streamlines or particle velocity paths. Characteristic numerical methods are developed by taking advantage of the solution being constant along these curves as a way to relieve the restrictive CFL constraint.

Characteristic schemes were first made viable with the modified method of characteristics (MMOC) defined by Douglas and Russell in [32]. In order to solve the linear problem on a time interval $[t^n, t^{n+1}]$, they develop a method to backward trace along characteristic curves. Their approach traces node points x at t^{n+1} to traceback points \tilde{x} at t^n . These values are used to approximate the derivative in the characteristic direction as

$$u_\tau(x, t) \equiv \frac{1}{(1 + v^2)^{1/2}} (u_t + v \cdot \nabla u(x, t)) \approx \frac{u(x, t^{n+1}) - u(\tilde{x}, t^n)}{\Delta t}, \quad (3.3)$$

resulting in the finite difference approximation

$$u(x, t^{n+1}) = u(\tilde{x}, t^n) + \Delta t q. \quad (3.4)$$

Since this method is based on individual points, it fails to be either mass or volume conservative. This work was furthered by the introduction of the modified method of characteristics with adjusted advection (MMOCAA) [30]. This method provides a correction for the global mass balance; however, there is no local mass conservation.

Eulerian-Lagrangian schemes were then developed in terms of a local mass balance. These types of methods include various Eulerian-Lagrangian

localized adjoint methods (ELLAM) [20, 26, 78, 77, 75] which include characteristic mixed methods (CMM) [9]. Instead of tracing node points, an entire grid element E is traced along the characteristic to a traceback element \check{E} creating a tessellation of the domain. This gives local mass conservation, since all the mass in \check{E} is numerically transported forward into E . These methods are not locally volume conservative, since in multiple dimensions it is not possible to trace the entire boundary of E . Methods were developed tracing a finite subset of the boundary points and connecting them with a piecewise polynomial interpolation of the boundary. Here the volume of the approximated traceback element is different than the volume of the theoretical traceback element. Arbogast and Huang developed the volume corrected characteristic mixed method (VCCMM) [3, 8] by adjusting the locations of the trace points in a way that ensured the method was both locally mass and volume conservative.

In 2011 Qiu, Shu and Christlieb [61, 62] derived semi-Lagrangian WENO finite difference schemes as a way to develop a high order transport scheme. Their scheme only works for constant velocity. In 2012 Arbogast and Huang created the Eulerian-Lagrangian WENO (EL-WENO) finite volume scheme [48] for the linear advection problem. This method does not naturally extend to higher spatial dimensions without the aid of a splitting technique.

The nonlinear problem is challenging since tracing either characteristic or particle paths is a function of the solution over the traceline. In 2012 Arbogast, Huang, and Russell [6] solve the nonlinear two-phase flow system

by tracing particle paths approximately and then adjusting them by solving an expensive minimization problem. The work later developed in [46] by Arbogast and Huang extends EL-WENO schemes to the nonlinear problem by solving the related problem

$$(u_t + vu_x) + (f(u)_x - vu_x) = q, \quad (3.5)$$

where v is chosen as an approximation to either the particle velocity field $f(u)/u$ or the wave velocity field $f'(u)$. The linear advection problem is solved using an EL-WENO technique, leaving a flux correction term which subjects the method to a relaxed CFL constraint. This method alleviates the need to solve the minimization problem, however the technique used to solve the flux correction at a high order accuracy involves a large spatial stencil.

The remainder of this dissertation will be focused on providing improvements to Eulerian-Lagrangian schemes, with the aid of the IWENO technique derived in Chapter 2. This chapter will be focused on developing a method to solve the multi-dimensional linear advection equation.

3.2 Quadrature Based Scheme

Let $u(x, t)$ be the mass which is transported in a velocity field $v(x, t)$ as described by

$$\begin{aligned} u_t + \nabla \cdot (vu) &= 0, & x \in \mathbb{R}^d, \quad t \in \mathbb{R}^+, \\ u(x, 0) &= u_0(x), & x \in \mathbb{R}^d. \end{aligned} \quad (3.6)$$

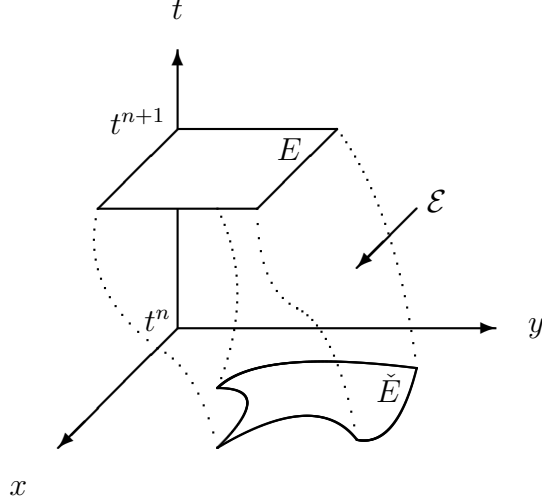


Figure 3.1: **Eulerian-Lagrangian Scheme.** Illustration of a theoretical Eulerian-Lagrangian scheme. All the points in E are traced through the velocity field v in order to find a traceback region \check{E} . This process also defines a space-time region \mathcal{E} for the time step $[t^n, t^{n+1}]$.

We will numerically solve (3.6) using an Eulerian-Lagrangian schemes where the mass in an element E is given by the mass located in a region \check{E} defined through backward characteristic tracing as shown in Figure 3.1. The tracing is defined at every point in E as the curve $\check{x}(t)$ satisfying

$$\begin{aligned} \frac{d\check{x}}{dt} &= v(\check{x}, t), \quad t^n \leq t < t^{n+1} \\ \check{x}(t^{n+1}) &= x. \end{aligned} \tag{3.7}$$

This tracing defines a map $F(x) : E \rightarrow \check{E}$. The volume swept out by this tracing is called the space-time traceback region and is denoted \mathcal{E} . Furthermore, the divergence theorem gives the mass in the traceback region \check{E} is exactly the same as the mass in the cell E at the future time.

$$\int_E u(x, t^{n+1}) dx = \int_{\check{E}} u(x, t^n) dx. \tag{3.8}$$

There is no flow across the boundary since the velocity field is tangent to the

boundary of \mathcal{E} .

Computationally it is not possible in spatial dimensions greater than one to trace all the points on the boundary of E . Meaning that the trace back region \check{E} has unknown geometry. In order to compute the integral over \check{E} a coordinate transformation is applied to the mass integral. This transformation is defined through the map F and the integral after the coordinate transformation is

$$\int_{\check{E}} u(x, t^n) dx = \int_E u(F(x), t^n) \left| \frac{\partial F}{\partial x} \right| dx. \quad (3.9)$$

Where $\left| \frac{\partial F}{\partial x} \right|$ is the determinant of the Jacobian matrix associated with the transformation. This integral is computed with a quadrature rule as

$$\sum_k w_k u(F(x_k), t^n) \left| \frac{\partial F}{\partial x} \right|_{x=x_k} \quad (3.10)$$

where x_k and w_k are quadrature points and weights. This evaluation requires evaluating two terms. The first is the reconstruction of u at time t^n and location $F(x_k)$ which is computed using the IWENO scheme detailed in Chapter 2. The second term is the determinant of the Jacobian matrix evaluated at the quadrature point. In section 3.2.1 we develop a system of ODEs that solve for the Jacobian determinant.

3.2.1 Jacobian Matrix

This section looks at the determinant of the Jacobian Matrix,

$$\left| \frac{\partial F}{\partial x} \right| = \begin{vmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_d}{\partial x_1} & \cdots & \frac{\partial F_d}{\partial x_d} \end{vmatrix}. \quad (3.11)$$

The map $F : E \rightarrow \check{E}$ is defined pointwise by particle tracing through the velocity field; that is, the components of the Jacobian are of the form

$$\frac{\partial F_i}{\partial x_j} = \frac{\partial \check{x}_i}{\partial x_j}. \quad (3.12)$$

The value of these terms at $t = t^{n+1}$ is understood since $\check{x}(t^{n+1}) = x$ or $(\check{x}_1(t^{n+1}), \dots, \check{x}_d(t^{n+1})) = (x_1, \dots, x_d)$ and

$$\left. \frac{\partial \check{x}_i}{\partial x_j} \right|_{t=t^{n+1}} = \delta_{ij}. \quad (3.13)$$

Here δ_{ij} is the Kroneker delta, associated with the identity matrix. In order to get the values of the components at t^n , we take the time derivative, which can be related to the gradient of the velocity field as

$$\frac{d}{dt} \frac{\partial \check{x}_i}{\partial x_j} = \frac{\partial}{\partial x_j} \frac{d\check{x}_i}{dt} = \frac{\partial v_i(\check{x}_i, t)}{\partial x_j} = \nabla v_i \cdot \left(\frac{\partial \check{x}_1}{\partial x_j}, \dots, \frac{\partial \check{x}_d}{\partial x_j} \right). \quad (3.14)$$

This sets up a system of ODEs

$$\frac{d}{dt} \sigma_{ij} = \nabla v_i(\check{x}, t) \cdot (\sigma_{1j}, \dots, \sigma_{dj}), \quad (3.15)$$

$$\frac{d}{dt} \check{x} = v, \quad (3.16)$$

$$\sigma_{i,j}(t^{n+1}) = \delta_{ij}, \quad (3.17)$$

$$\check{x}(t^{n+1}) = x, \quad (3.18)$$

where $\sigma_{ij} = \frac{\partial \check{x}_i}{\partial x_j}$. The solution of the system at $t = t^n$ gives the components of the Jacobian Matrix at any point. This is evaluated for the finite subset of quadrature points and used to compute the value of the determinant at these locations.

An alternative method is to solve

$$\frac{d}{dt} \left| \frac{\partial \check{x}}{\partial x} \right| = \left| \frac{\partial \check{x}}{\partial x} \right| (\nabla \cdot v) \quad (3.19)$$

for the determinant without resolving the individual components as derived in [39].

3.2.2 Local Conservation

Previous work with Eulerian-Lagrangian schemes guaranteed a local mass conservation since the elements tessellated the domain, and mass was directly transported forward. Since the quadrature scheme only traces select points and not the entire domain, it is neither locally mass nor volume conservative. In this section, we will derive a volume conservation equation and demonstrate how to determine the number of quadrature points to use in the scheme.

The volume difference of an element E and that of its traceback element \check{E} is the integral over \mathcal{E} of the divergence of the velocity field, i.e.

$$\int_{\mathcal{E}} \nabla_x \cdot v \, dV = |E| - |\check{E}|. \quad (3.20)$$

This is verified in two spatial dimensions by noting

$$\begin{aligned}
\int_{\mathcal{E}} \nabla_x \cdot v \, dV &= \int_{\mathcal{E}} \nabla_{x,t} \cdot \begin{pmatrix} v \\ 1 \end{pmatrix} dV \\
&= \int_{\partial\mathcal{E}} \begin{pmatrix} v \\ 1 \end{pmatrix} \cdot \eta_{x,t} \, dA \\
&= \int_E \begin{pmatrix} v \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} dx + \int_{\check{E}} \begin{pmatrix} v \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -1 \end{pmatrix} dx \\
&\quad + \sum_{i=1}^4 \int_{S_i} \begin{pmatrix} v \\ 1 \end{pmatrix} \cdot \eta_{x,t} \, dx \\
&= \int_E 1 \, dx + \int_{\check{E}} -1 \, dx + \sum_{i=1}^4 \int_{S_i} 0 \, dx \\
&= |E| - |\check{E}|;
\end{aligned}$$

here S_i denotes a face of the traceback region. A normal vector to a given surface S_i must be orthogonal to the tangent vector on the surface which is $(\check{x}(t), t)' = (v, 1)$. This explains why the the dot product over these surfaces in the direction of the normal is zero. A scheme is called volume conservative if equation (3.20) is satisfied. When $\nabla \cdot v = 0$, we say the flow is divergence free, and in this case volume conservation is simply $|E| = |\check{E}|$.

For the quadrature based scheme, the volume conservation equation is used to evaluate a local volume error. The right hand side of (3.20) is evaluated using the coordinate transformation with respect the the map F

and with quadrature points, x_k , and weights, w_k , as

$$\begin{aligned}
|E| - |\check{E}| &= |E| - \int_{\check{E}} 1 \, dA \\
&= |E| - \int_E \left| \frac{\partial F}{\partial x} \right|_{t^n} dA \\
&= |E| - \sum_k w_k \left| \frac{\partial F}{\partial x} \right|_{(x_k, t^n)}.
\end{aligned} \tag{3.21}$$

The left hand side is similarly computed as

$$\begin{aligned}
\int_{\mathcal{E}} \nabla_x \cdot v \, dV &= \int_{t^n}^{t^{n+1}} \int_{E(t)} \nabla_x \cdot v \, dA \, dt \\
&= \sum_j \tilde{w}_j \int_{E(t_j)} \nabla_x \cdot v \, dA \\
&= \sum_j \tilde{w}_j \int_E (\nabla_x \cdot v) \left| \frac{\partial F}{\partial x} \right|_{t_j} dA \\
&= \sum_j \tilde{w}_j \sum_k w_k (\nabla_x \cdot v) \left| \frac{\partial F}{\partial x} \right|_{(x_k, t_j)},
\end{aligned} \tag{3.22}$$

where \tilde{w}_j and t_j are the quadrature weights and points in time. The local volume error estimate is the difference between equations (3.21) and (3.22),

$$|E| - \left(\sum_k w_k \left| \frac{\partial F}{\partial x} \right|_{(x_k, t_n)} \right) - \left(\sum_j \tilde{w}_j \sum_k w_k (\nabla_x \cdot v) \left| \frac{\partial F}{\partial x} \right|_{(x_k, t_j)} \right). \tag{3.23}$$

The code adaptively chooses the number of quadrature points such that the volume error on a given element is less than a preset tolerance.

Local mass conservation means that the mass on element \check{E} is forward transported to E that is $\int_E u \, dA = \int_{\check{E}} u \, dA$. Since quadrature is used to solve for the mass on \check{E} this requirement is not satisfied. In order to minimize this error a fifth order IWENO reconstruction scheme is used to compute

$u(F(x_k), t^n)$ and the selection of the quantity of quadrature points helps to reduce the error associated with computing $\left| \frac{\partial F}{\partial x} \right|_{x_k}$. In addition, a global mass error is monitored by checking the difference in total mass at the initial state with the total mass at a given time step.

3.2.3 Algorithm

The problem is solved on a uniform grid with spacing Δx in the x -direction and Δy in the y -direction, as is required by the IWENO interpolation scheme to construct the primitive function. The following procedure gives the algorithm to compute the mass at t^{n+1} for a given element.

1. Start with an initial number of quadrature points n_q . For the two dimensional problem we use a tensor product of quadrature points; that is, in both spatial directions we use n_q points for a total of n_q^2 quadrature points. Then solve the system of ODE's (3.15)-(3.18) to obtain the values of \tilde{x} , \tilde{y} and $\left| \frac{\partial F}{\partial x} \right|$ for each quadrature point. The numerical solver implemented in our code is the fifth order components of the Runge-Kutta 45 scheme [35] proposed by Fehlberg in 1969, it is defined by the Butcher tableau in Table 3.1. For complex velocity fields and large time steps it is necessary to take multiple micro time steps in the Runge-Kutta scheme.
2. Use the values of the determinant of the Jacobian $\left| \frac{\partial F}{\partial x} \right|$ to compute the local volume error in equation (3.23). If the error is greater than the set

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	-7200/2197	7296/2197			
1	439/216	-8	3680/513	-845/4104		
1/2	-8/27	2	-3544/2565	1859/4104	-11/40	
	16/135	0	6656/12825	28561/56430	-9/50	2/55

Table 3.1: **Runge-Kutta Butcher Tableau.** The fifth order Runge-Kutta scheme used to solve the ODE system (3.15)-(3.18).

tolerance, the number of quadrature points is incremented by one and step 1 is repeated to solve for \tilde{x} , \tilde{y} and $|\frac{\partial F}{\partial x}|$ associated with the new set of quadrature points.

3. Evaluate $u(\tilde{x}, \tilde{y}, t^n)$ using the IWENO scheme given in Chapter 2.2.5. The values $u(\tilde{x}, \tilde{y}, t^n)$ and $|\frac{\partial F}{\partial x}|$ are used to solve equation 3.10 which approximates the total mass in the element at t^{n+1} .

3.2.4 Numerical Results in 1D

In this section numerical results for a variety of linear advection problems are provided. The problem is set up by giving the initial condition $u_0(x)$ and the velocity field v and its Jacobian $J = \frac{dv}{dx}$ that the mass will transport in. A periodic boundary condition is implemented for all of the examples and the tolerance for the volume computation is set at 1E-9. The global mass loss is verified to be less in magnitude than the computed error. These errors

are presented using the discrete L^1 norm

$$\|u^n - \bar{u}^n\|_1 := \sum_i \left| \frac{1}{h} \int_{E_i} u(x, t^n) dx - \bar{u}_i^n \right| h, \quad (3.24)$$

and the discrete L^∞ norm

$$\|u^n - \bar{u}^n\|_\infty := \sup_i \left| \frac{1}{h} \int_{E_i} u(x, t^n) dx - \bar{u}_i^n \right|. \quad (3.25)$$

The quadrature rule use for the numerical results is Gaussian Quadrature [45] which on the reference element $[-1, 1]$ has points and weights listed in Table 3.2. The order of the approximation is Δx^{2n_q-1} . In order to maintain fifth order accuracy the scheme initializes the number of quadrature points at $n_q = 3$. There are other natural choices for the quadrature rule such as Gauss-Lobatto which includes points on the boundary; however, this particular quadrature rule did not produce significantly different results from the Gaussian quadrature and requires a greater number of points to achieve a given order of accuracy.

3.2.4.1 Example: Constant Speed Transport

We first test our scheme with the simple case of constant speed transport with velocity $v(x, t) = 1$ where $v_x = 0$. The initial condition is $u_0(x) = \sin(\pi x)$ for $x \in [0, 2]$. The exact solution is $u(x, t) = u_0(x - t)$. The trace points and determinant of the Jacobian are computed using the Runge-Kutta scheme using 1 time step. This is more than sufficient since the trace points are of the form $\check{x} = x - \Delta t$ and $\left| \frac{\partial \check{x}}{\partial x} \right| = 1$ and are solved exactly using the single Runge-Kutta time step.

Gaussian Quadrature			Gauss-Lobatto Quadrature		
n_q	x_i	w_i	n_q	x_i	w_i
1	0	2	2	± 1	1
2	$\pm\sqrt{\frac{1}{3}}$	1	3	0	$\frac{4}{3}$
3	0	$\frac{8}{9}$		± 1	$\frac{1}{3}$
4	$\pm\sqrt{\frac{3}{5}}$	$\frac{5}{9}$	4	$\frac{1}{5}\sqrt{5}$	$\frac{5}{6}$
	$\pm\sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18+\sqrt{30}}{36}$		± 1	$\frac{1}{6}$
5	$\pm\sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18-\sqrt{30}}{36}$	5	0	$\frac{32}{45}$
	0	$\frac{128}{225}$		$\pm\frac{1}{7}\sqrt{21}$	$\frac{49}{90}$
5	$\pm\frac{1}{3}\sqrt{5 - 2\sqrt{\frac{10}{7}}}$	$\frac{322+13\sqrt{70}}{900}$	6	± 1	$\frac{1}{10}$
	$\pm\frac{1}{3}\sqrt{5 + 2\sqrt{\frac{10}{7}}}$	$\frac{322-13\sqrt{70}}{900}$		$\pm\sqrt{\frac{1}{21}(7 - 2\sqrt{7})}$	$\frac{1}{30}(14 + \sqrt{7})$
				$\pm\sqrt{\frac{1}{21}(7 + 2\sqrt{7})}$	$\frac{1}{30}(14 - \sqrt{7})$
				± 1	$\frac{1}{15}$

Table 3.2: **Quadrature Rules.** The quadrature points x_i and weights w_i on a reference element $[-1, 1]$. On the left is Gaussian quadrature, on the right is Gauss-Lobatto quadrature.

Numerical results in Table 3.3 are at $t = 1.5$ at three different time steps. The first is for $\Delta t = 0.1$ here elements will trace exactly to other elements in the domain. The results show the error to be within the rounding error of the scheme. The second choice of time step is $\Delta t = 0.12$ the results show 5th order convergence. For the final test $\Delta t = 4.5\Delta x$ this is 4.5 times the CFL time limit of a fixed grid WENO scheme where

$$\Delta t_{\text{CFL}} = \Delta x / \max |v| = \Delta x.$$

. The results show a larger error accumulation in comparison to taking a small fixed number of time steps.

3.2.4.2 Example: Time Dependent Velocity

This example uses the velocity $v = \sin(t)$ with $v_x = 0$. The initial condition is again $u_0(x) = \sin(\pi x)$ for $x \in [0, 2]$. The analytical solution to this problem is $u(x, t) = u_0(x + 1 + \cos(t))$. The trace points and determinant of the Jacobian are computed using the Runge-Kutta scheme with a single step. The error results at $t = 1.5$ using 10 time steps are listed in Table 3.4 show the 5th order convergence rate.

3.2.4.3 Example: Spatially Dependent Velocity

The most challenging test we run in 1D is with the velocity $v = \sin(x)$ with $v_x = \cos(x)$. The initial condition is $u_0(x) = 1$ for $x \in [0, 2\pi]$. The

N	Error L^1	Order	Error L^∞	Order
$\Delta t = 0.1$				
20	4.62E-15	-	6.23E-15	-
40	1.45E-14	-	1.84E-14	-
80	4.82E-14	-	1.09E-13	-
160	9.00E-14	-	3.74E-13	-
320	2.09E-13	-	1.82E-12	-
640	3.74E-13	-	6.28E-12	-
$\Delta t = 0.12$				
10	4.93 E-3	-	3.81 E- 3	-
20	1.42E- 4	5.11	1.11E- 4	5.10
40	2.54E- 6	5.80	2.00E- 6	5.79
80	1.35E- 7	4.23	1.06E- 7	4.23
160	2.25E- 9	5.91	1.77E- 9	5.91
320	1.32E- 10	4.09	1.04E- 10	4.09
640	2.25E- 12	5.87	6.37E- 12	4.02
LR		5.12		4.92
$\Delta t = 4.5\Delta x$				
10	7.89E-4	-	5.69 E-4	-
20	3.06E- 5	4.69	3.90E-5	3.87
40	1.66E-6	4.21	1.94E-6	4.33
80	6.31E-8	4.71	1.09E-7	4.15
160	8.19E-9	2.95	6.64E-9	4.04
320	2.25E-10	4.08	E-10	4.03
640	5.93E-11	4.63	E-11	3.14
LR		4.03		4.00

Table 3.3: **Constant Transport Test 1D.** Error and convergence order for transport with velocity $v = 1$ at time $t = 1.5$ with $\Delta t = 0.1, 0.12,$ and $4.5\Delta x$. LR stands for linear regression, and the order is the slope of the linear regression line through all the data points.

N	Error L_1	Order L_1	Error L_∞	Order L_∞
10	2.92 E-3	-	2.26 E-3	-
20	9.71 E-5	4.91	7.65 E-5	4.89
40	8.20 E-7	6.89	6.42 E-7	6.90
80	3.90 E-8	4.39	3.06 E-8	4.39
160	3.83 E-9	3.35	3.00 E-9	3.35
LR		5.04		5.03

Table 3.4: **Time Dependent Transport Test 1D.** Error and convergence order for transport with velocity $v = \sin(t)$ at time $t = 1.5$ with $\Delta t = 0.15$.

analytical solution to this problem is

$$u(x, t) = \frac{\sin(2 \arctan(e^{-t} \tan(x/2)))}{\sin(x)}. \quad (3.26)$$

The trace points and determinant of the Jacobian are computed using the Runge-Kutta scheme using $N_{\text{RK}} = 10$ micro time steps. This value N_{RK} was sufficient to keep the volume conservation equation within the required tolerance, and experimentation using a larger value of N_{RK} did not improve the error results. These results are presented in Table 3.5 at $t = 1$ with $\Delta t = 0.1$.

3.2.4.4 Example: Shu's Linear Test

The final example in 1D is the standard test case Shu's linear test [14]. In this case $v(x, t) = 1$ and $v_x = 0$, and the analytical solution is $u(x, t) = u_0(x - t)$. The initial condition is

N	Error L_1	Order L_1	Error L_∞	Order L_∞
10	6.39 E-2	-	3.27 E-3	-
20	3.63 E-3	4.13	7.65 E-5	3.61
40	2.12 E-4	4.10	6.42 E-7	3.75
80	7.13 E-6	4.89	3.06 E-8	3.98
160	1.85 E-7	5.27	3.00 E-9	5.32
LR		4.58		4.10

Table 3.5: **Spatially Dependent Transport Test 1D.** Error and convergence order for transport with velocity $v = \sin(x)$ at time $t = 1$ with $\Delta t = 0.1$.

$$u_0(x) = \begin{cases} \frac{1}{6}(G(x, \beta, z - \delta) + G(x, \beta, z + \delta) + 4G(x, \beta, z)), & 0.2 \leq x \leq 0.4 \\ 1, & 0.6 \leq x \leq 0.8 \\ 1 - |10(x - 1.1)|, & 1 \leq x \leq 1.2 \\ \frac{1}{6}(F(x, \alpha, a - \delta) + F(x, \alpha, a + \delta) + 4F(x, \alpha, a)), & 1.4 \leq x \leq 1.6 \\ 0, & \text{otherwise,} \end{cases}$$

where

$$G(x, \beta, z) = e^{-\beta(x-z)^2} \quad \text{and} \quad F(x, \alpha, a) = \sqrt{\max(1 - \alpha^2(x - a)^2, 0)}.$$

The constants are set to $a = 0.5$, $z = 0.3$, $\delta = 0.005$, $\alpha = 10$, and $\beta = \log(2)/(36\delta^2)$. This test is designed to see the resolution properties of a given scheme. One time step is used in the Runge-Kutta scheme solving the ODE system. Two tests are run, the first is for the number of elements $N = 200$ the second is for $N = 201$. The problem is solved after 4 rotations at time $t = 8$ using 20 time steps. The solutions at the final time are in Figure 3.2. The first case exhibits similar behavior to the first test in Section 3.2.4.1. There is no visible diffusion and the global mass error is within rounding error. The

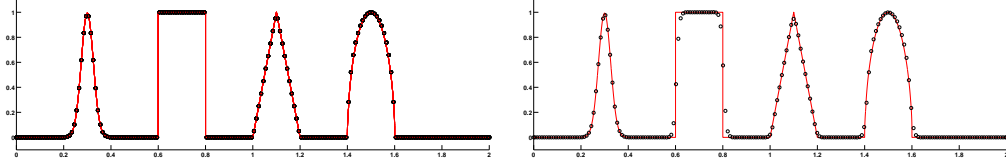


Figure 3.2: **Shu's Linear Test 1D**. Numerical solutions for Shu's Linear Test at time $t = 8$ with 20 time steps. On the left there are 200 grid elements and on the right there are 201 elements.

second test we run with $N = 201$ this time trace element are no longer the same as grid elements. In this case some diffusion accumulates, and the global mass error is of the magnitude $1\text{E}-4$.

3.2.5 Numerical Results in 2D

In 2D the the linear advection equation is

$$u_t + (a_1 u)_x + (a_2 u)_y = 0, \quad (3.27)$$

where the velocity is $v = (a_1, a_2)$ with Jacobian $J(v) = \begin{pmatrix} (a_1)_x & (a_1)_y \\ (a_2)_x & (a_2)_y \end{pmatrix}$. The discrete L^1 norm is

$$\|u^n - \bar{u}^n\|_1 := \sum_{i,j} \left| \frac{1}{h^2} \int_{E_{i,j}} u(x, t^n) dA - \bar{u}_{i,j}^n \right| h^2, \quad (3.28)$$

and the discrete L^∞ norm is

$$\|u^n - \bar{u}^n\|_\infty := \sup_{i,j} \left| \frac{1}{h^2} \int_{E_{i,j}} u(x, t^n) dA - \bar{u}_{i,j}^n \right|. \quad (3.29)$$

All the numerical examples in this section use a tolerance of $1\text{E}-9$ for the volume computation, and use periodic boundary conditions on the uniform grid with $\Delta x = \Delta y$.

N	Error L^1	Order	Error L^∞	Order
10	2.34 E-2	-	9.02 E-3	-
20	4.30 E-4	5.76	1.67 E-4	5.75
40	2.10 E-5	4.36	8.24 E-6	4.32
80	4.06 E-7	5.69	1.59 E-7	5.69
160	2.04 E-8	4.32	8.00 E-9	4.32
LR		5.03		5.02

Table 3.6: **Constant Transport Test 2D.** Error and convergence order for transport with velocity $v = (1, 1)$ at time $t = 1.2$ with $\Delta t = 0.12$.

3.2.5.1 Example: Constant Speed Transport

The first example we give is for the constant velocity field $v = (1, 1)$ with Jacobian $J(v) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. The initial condition is $u_0(x, y) = \sin(\pi(x+y))$ for $(x, y) \in [0, 2]^2$. The analytic solution is $u(x, y, t) = \sin(\pi(x+y-2t))$. The ODE system is solved using one Runge-Kutta time step, which is sufficient to solve for the trace points and Jacobian exactly. The error at time $t = 1.2$ using 10 time steps is given in Table 3.6 which shows the theoretical 5th order convergence rate.

3.2.5.2 Example: Rigid Body Rotation

Two dimensional rigid body rotation is described by

$$u_t - ((y-1)u)_x + ((x-1)u)_y, \quad (3.30)$$

for $(x, y) \in [0, 2]^2$. The Jacobian of the velocity field is

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \quad (3.31)$$

N	Error L^1	Order	Error L^∞	Order
10	1.23 E-3	-	1.49 E-3	-
20	1.33 E-4	3.2	1.46 E-4	3.35
40	5.44 E-6	4.6	8.15 E-6	4.16
80	1.23 E-7	5.5	2.07 E-7	5.3
160	2.90 E-8	2.1	1.33 E-8	4.0
LR		4.08		4.3

Table 3.7: **Rigid Body Rotation Test 2D.** Error and convergence order for transport with velocity $v = (1 - y, x - 1)$ at time $t = 1$ with $\Delta t = 0.1$.

The initial condition is a smooth radial bump function

$$u(x, y, 0) = \frac{2}{5}[\psi(1 + r(x, y))\psi(1 - r(x, y)) + 1], \quad (3.32)$$

$$r(x, y) = \sqrt{(x - 1)^2 + (y - 1)^2}, \quad (3.33)$$

with $\psi(s) = e^{-1/s^2}$ for $s > 2$ and $\psi(s) = 0$ otherwise. The Runge-Kutta scheme uses one time step. The numerical results at time $t = 1$ using 10 time steps are presented in Table 3.7.

3.2.5.3 Example: Swirling and Deforming Velocity

The final and most severe test is obtained using a swirling deformation flow where the velocity field is given by

$$a_1(x, y) = \sin^2\left(\frac{\pi x}{2}\right) \sin(\pi y)g(t) \quad (3.34)$$

$$a_2(x, y) = -\sin^2\left(\frac{\pi y}{2}\right) \sin(\pi x)g(t). \quad (3.35)$$

The function $g(t)$ introduces a time dependence to the flow field

$$g(t) = 2 \cos(\pi t/T) \quad (3.36)$$

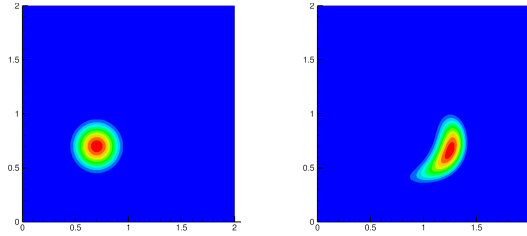


Figure 3.3: **Swirling Deforming Flow 2D.** On the left is the initial condition, on the right is the numerical solution at $t = T/2$.

this equation solved on a time interval $t \in [0, T]$. This velocity field has flow that slows down and reverses directions so that the initial condition is recovered at the final time T . The initial condition is on the domain $[0, 2]^2$ and is a radial bump function centered at the point $(0.7, 0.7)$. Figure 3.3 illustrates the initial condition and the solution at $t = T/2$. This test has a Jacobian given by

$$(a_1)_x = 2\pi \cos(\pi x/2) \sin(\pi x/2) \sin(\pi y) \cos(\pi t/T), \quad (3.37)$$

$$(a_1)_y = 2\pi \sin(\pi x/2)^2 \cos(\pi y) \cos(\pi t/T), \quad (3.38)$$

$$(a_2)_x = -2\pi \sin(\pi y/2)^2 \cos(\pi x) \cos(\pi t/T), \quad (3.39)$$

$$(a_2)_y = -2\pi \cos(\pi y/2) \sin(\pi y/2) \sin(\pi x) \cos(\pi t/T). \quad (3.40)$$

Numerical results in Table 3.8 are for the solution at $T = 1$ using $\Delta t = 0.1$ and 10 micro time steps for the Runge-Kutta scheme.

N	Error L^1	Order	Error L^∞	Order
10	2.95 E-3	-	2.03 E-2	-
20	8.99 E-4	1.7	4.87 E-3	2.06
40	9.22 E-5	3.29	8.34 E-4	2.55
80	4.58 E-6	4.33	9.12 E-5	3.19
160	1.17 E-7	5.28	3.48 E-6	4.71
LR		3.82		3.08

Table 3.8: **Swirling and Deforming Flow 2D.** Error and convergence order for transport with a time and spatially dependent velocity field at time $t = 1$ with $\Delta t = 0.1$.

Chapter 4

EL-WENO for Nonlinear Advection in 1D

We follow the method of Arbogast and Huang in [46] solving the non-linear problem $u_t + f(u)_x = 0$ by solving the related problem

$$(u_t + (vu)_x) + (f(u) - vu)_x = 0, \quad (4.1)$$

for an appropriately chosen velocity v . After integrating over the space-time region $\mathcal{E}(v)$ their method uses an EL-WENO scheme to solve the linear advection problem and uses a quadrature rule to solve for the flux integral

$$\int_{\mathcal{E}(v)} (f(u) - vu)_x dA = \int_{\partial\mathcal{E}(v)} \begin{pmatrix} f(u) - vu \\ 0 \end{pmatrix} \cdot \eta_{x,t} dx. \quad (4.2)$$

This requires the evaluation of u along the boundary of the space-time region. The 5th order scheme of Arbogast and Huang computes this using a Runge-Kutta method that requires a 17 point stencil.

Arbogast and Huang were motivated by the CWENO schemes of Levy Puppo and Russo [51, 53, 52]. This chapter first reviews CWENO schemes as a way to understand the stencil for computing the integral of the flux term. This is a reasonable framework since it is easier to understand in a fixed mesh setting and the EL-WENO method generalizes to a fixed mesh WENO method when the velocity field v is zero.

The purpose of this chapter is to provide a new algorithm for computing the flux term (4.2) using a compact local stencil. Concluding with numerical results directly comparing these methods and demonstrating formal convergence rates for the new scheme.

4.1 CWENO Flux Stencil

Central WENO schemes solve for the staggered cell averages

$$\begin{aligned} \bar{u}_{i+1/2}^{n+1} = & 1/h \int_{x_i}^{x_{i+1}} R(x, t^n) dx \\ & + \frac{\Delta t}{\Delta x} \sum_{l=0}^m \gamma_l [f(u(x_i, t^n + \beta_l \Delta t)) - f(u(x_{i+1}, t^n + \beta_l \Delta t))], \end{aligned} \quad (4.3)$$

where R is a piecewise polynomial reconstruction of the cell averaged data. The values γ_l, β_l are quadrature weights and points describing the intermediate times $t^n + \beta_l \Delta t$ where u is computed. An illustration of this scheme is shown in Figure 4.1.

A *Natural Continuous Extension* (NCE) [85] of a Runge Kutta scheme is used to evolve the mass u at the fixed position x_i in time. The concentration changes in time according to the nonlinear advection equation as $u_t = f(u)_x|_{x_i}$. The NCE Runge-Kutta scheme provides uniform accuracy for the solution in the time interval $[t^n, t^{n+1}]$. Each v -stage Runge Kutta method of order p has a NCE of order $d \leq p$. That is there exist polynomials $b_i(\theta), i = 1, \dots, v$ of degree

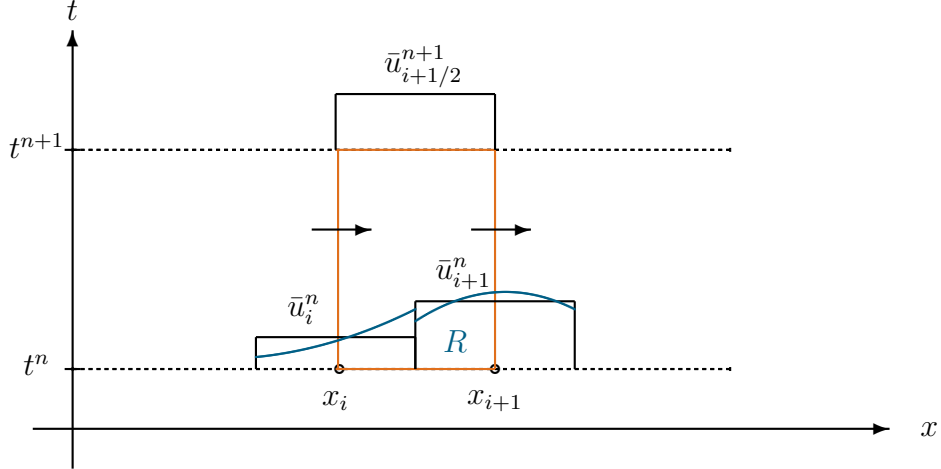


Figure 4.1: **Central WENO Scheme.** An illustration of a time step in a central WENO scheme. The staggered grid element average $\bar{u}_{i+1/2}^{n+1}$ at the advanced time t^{n+1} is computed from the reconstructed solution $R(x)$ over the interval $[x_i, x_{i+1}]$ at the time t^n and a computation of the flux across the midpoints over the time interval $[t^n, t^{n+1}]$.

at most d where

$$u(t^n + \theta\Delta t) = u(t^n) + \Delta t \sum_{i=1}^v b_i(\theta)g^{(i)}, \quad (4.4)$$

$$g^{(i)} = F \left(t^n + \Delta t c_i, u^n + \Delta t \sum_{j=1}^v a_{ij}g^{(j)} \right), \quad (4.5)$$

$$c_i = \sum_j a_{ij}, \quad (4.6)$$

$$F(\tau, u(\tau)) = -f_x(u(t^n + \tau)) \quad (4.7)$$

defines the NCE. The underlying Runge-Kutta scheme is determined by the selection of matrix a and vector b . When matrix a is lower triangular the scheme is explicit. These schemes are useful since they evaluate u at multiple times along the line $x = x_i$ without solving for the parameters g at each time separately.

The 5th order CWENO method uses a 4-stage Runge Kutta scheme. At

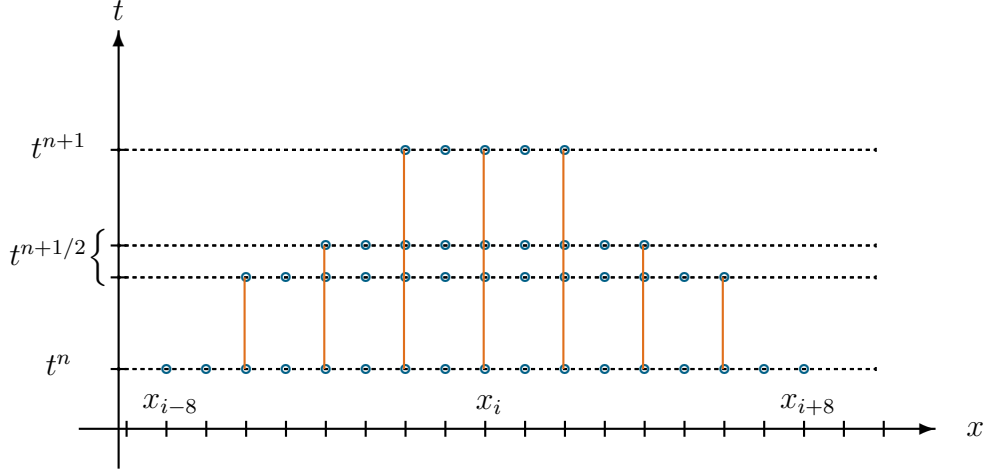


Figure 4.2: **Implicit Stencil for CWENO5 Flux Computation** A depiction of the 4-stage Runge Kutta stencil using a 5 point WENO reconstruction for computing the derivative $f(u(x_i, t))_x$.

time t^n the values of $u(x_{i+m})$ are evaluated at the 17 points with $m = -8, \dots, 8$. Then a 5 point WENO reconstruction scheme is used to get $f(u(x_{i+m}))_x$ for $m = -6, \dots, 6$. The value of u at these 13 points are evolved to the second Runge-Kutta stage. They are used in the WENO scheme to evaluate $f(u(x_{i+m}))_x$ for $m = -4, \dots, 4$. These value of u at these 9 points is computed at stage 3, where the value of $f(u(x_{i+m}))_x$ is reconstructed for 5 points with $m = -2, \dots, 2$. At the fourth stage these values are used to compute $f(u(x_i))_x$. This process is illustrated in Figure 4.2. In a standard CWENO scheme this stencil is implicitly defined since at each stage all the grid points are evolved at the same time.

This method was used by Arbogast and Huang in their EL-WENO scheme solving the nonlinear problem except for each traceline $\tilde{x}(t)$ they temporarily freeze the velocity field and explicitly compute these terms for each grid point separately. There is a detailed explanation of this procedure in their

paper [46]. The contribution in this chapter is to use a variation of the IWENO scheme to develop an alternative method for computing the flux integral with a compact local stencil.

4.2 A Flux Corrected EL-WENO Scheme

This section will develop a complete algorithm for solving the flux corrected EL-WENO scheme for the nonlinear advection problem. This is done by solving the related problem (4.1) in an Eulerian-Lagrangian setting. In order to define the velocity field v the characteristic equation

$$\frac{d\xi}{dt} = \tilde{v}(\tilde{x}, t), \quad t^n \leq t < t^{n+1}, \quad (4.8)$$

$$\xi(t^{n+1}) = x, \quad (4.9)$$

is solved at node points. Here \tilde{v} is an approximation to either the particle velocity field $f(u)/u$ or the wave velocity field $f'(u)$. The velocity field v at a grid point is the constant value $v_i = \frac{x_i - \xi_i}{\Delta t}$ and the velocity on element $E_i = [x_i, x_{i+1}]$ is the linear interpolation

$$v(x_{i+\alpha}) = (1 - \alpha)v_i + \alpha v_{i+1} \quad (4.10)$$

with $x_{i+\alpha} = x_i + \alpha\Delta x$, where $\alpha \in [0, 1]$. The tracelines at grid points are

$$\tilde{x}_i(t) = x_i(t^{n+1}) - v_i(t^{n+1} - t). \quad (4.11)$$

This velocity also defines the space-time region $\mathcal{E}(v)$ and the trace region $\check{E}(v)$ for every element.

The scheme is defined by integrating over the space-time region $\mathcal{E}(v)$

$$\int_{\mathcal{E}(v)} \nabla_{x,t} \cdot \begin{pmatrix} vu \\ u \end{pmatrix} dA + \int_{\mathcal{E}(v)} \nabla_{x,t} \cdot \begin{pmatrix} f(u) - vu \\ 0 \end{pmatrix} dA = 0, \quad (4.12)$$

and applying the divergence theorem

$$\oint_{\partial\mathcal{E}(v)} \begin{pmatrix} vu \\ u \end{pmatrix} \cdot \begin{pmatrix} \eta_x \\ \eta_t \end{pmatrix} dS + \oint_{\partial\mathcal{E}(v)} \begin{pmatrix} f(u) - vu \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \eta_x \\ \eta_t \end{pmatrix} dS = 0. \quad (4.13)$$

The boundary of the space-time region $\partial\mathcal{E}(v)$ has 4 sides. They are E_i with outward unit normal vector $\eta = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, \tilde{E}_i with $\eta = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, S_i with $\eta = \begin{pmatrix} -1 \\ v_i \end{pmatrix} / \sqrt{1 + v_i^2}$ and S_{i+1} with $\eta = \begin{pmatrix} 1 \\ -v_{i+1} \end{pmatrix} / \sqrt{1 + v_{i+1}^2}$. This is depicted in Figure 4.3. Substituting these in the equation yields

$$\begin{aligned} \int_{E_i} u^{n+1} dx &= \int_{\tilde{E}_i} u^n dx + \int_{S_i} (f(u) - vu) \frac{d\sigma}{\sqrt{1 + v_i^2}} \\ &\quad - \int_{S_{i+1}} (f(u) - vu) \frac{d\sigma}{\sqrt{1 + v_{i+1}^2}}. \end{aligned} \quad (4.14)$$

In order to solve the right hand side we apply a two step approach. The first step is a reconstruction $R(x)$ of the integral of the concentration u on the trace element \tilde{E} using WENO techniques. The second step is to solve for the flux integral approximately using a quadrature rule

$$\int_{S_i} (f(u) - vu) \frac{d\sigma}{\sqrt{1 + v_i^2}} = \sum_{k=1}^{n_q} w_k \Delta t \left(f(u(x_k, t_k)) - v_i u(x_k, t_k) \right), \quad (4.15)$$

where n_q is the number of quadrature points, and (x_k, t_k) and w_k are the quadrature locations and weights. Since the point (x_k, t_k) is on S_i it satisfies the relation

$$x_k = \tilde{x}_i(t^n) + v_i(t_k - t^n). \quad (4.16)$$

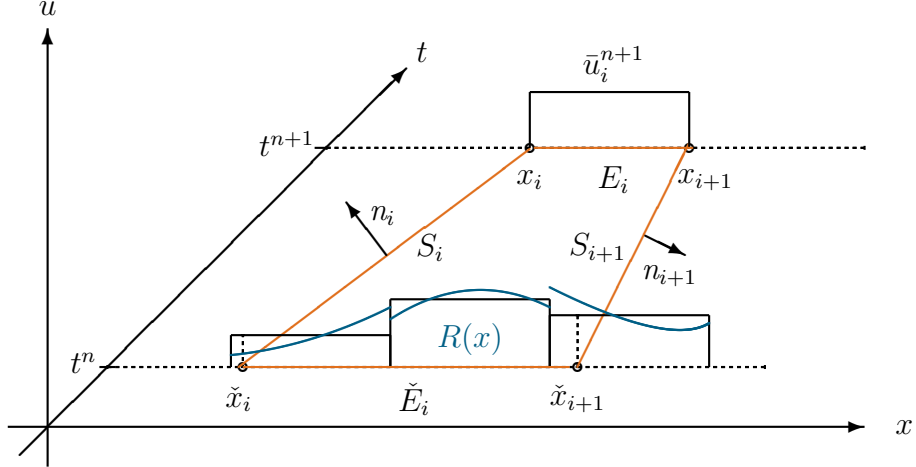


Figure 4.3: **Flux Corrected EL-WENO Scheme.** An illustration of a time step in a flux corrected EL-WENO scheme. The grid element average \bar{u}_i^{n+1} at the advanced time t^{n+1} is computed from the reconstructed solution, $R(x)$, over the interval $[\check{x}_i, \check{x}_{i+1}]$ at the time t^n , and a correction for flux values across the trace lines.

The evolution of u along S_i is computed using a NCE Runge-Kutta scheme that uses the differentiation technique presented for the IWENO scheme.

This method can be extended to two spatial dimensions by using a Strang splitting technique where the conservation equation

$$u_t + f_1(u)_x + f_2(u)_y = 0, \quad \text{for } (x, y) \in \mathbb{R}^2, t > 0, \quad (4.17)$$

$$u(x, y, 0) = u_0(x, y), \quad \text{for } (x, y) \in \mathbb{R}^2, \quad (4.18)$$

is solved by decoupling to a series of one dimensional problems. We do not provide details for this sort of algorithm, instead in Chapter 5 this flux correction technique is combined with the QEL-WENO scheme of Chapter 3 to provide a method for the nonlinear problem in multiple spatial dimensions.

4.2.1 Mass Reconstruction

There are two choices of methods to compute the mass integral

$$\int_{\check{E}_i} u(x, t^n) dx. \quad (4.19)$$

The first is the QEL-WENO scheme given in Chapter 3. The second choice, which is the one implemented in the code for the numerical results presented in this chapter, is the standard EL-WENO method. The reason for this choice is it allows direct comparison of the computation of the flux term for the two different techniques. In the paper [48] developing the EL-WENO technique, Arbogast and Huang provide a detailed explanation of the mass integration. In this section I will briefly outline their method and give the details to implement it for schemes of 3rd and 5th order.

Trace regions on a given element are explicitly defined as $\check{E}_j^n = [\check{x}_j, \check{x}_{j+1}]$. They are viewed as a union of subintervals of grid elements E_i of the form

1. $[x_i, x_{i+1}]$, when the whole E_i is contained in \check{E}_j^n ;
2. $[x_i, x_{i+\alpha}]$, $0 < \alpha < 1$, when \check{x}_{j+1}^n intersects E_i but not \check{x}_j^n ;
3. $[x_{i+\beta}, x_{i+1}]$, $0 < \beta < 1$, when \check{x}_j^n intersects E_i but not \check{x}_{j+1}^n ;
4. $[x_{i+\beta}, x_{i+\alpha}]$, $0 < \beta < \alpha < 1$, when \check{E}_j^n is contained in E_i .

High order reconstruction of u over these types of subintervals can be computed as integrals over type 1 and type 2 subintervals by noticing that type 3 and 4

subintervals are combinations of type 1 and 2 subintervals. Type 3 subintervals are

$$[x_{i+\beta}, x_{i+1}] = [x_i, x_{i+1}] - [x_i, x_{i+\beta}] \quad (4.20)$$

and type 4 subintervals are

$$[x_{i+\beta}, x_{i+\alpha}] = [x_i, x_{i+\alpha}] - [x_i, x_{i+\beta}]. \quad (4.21)$$

This gives a way of breaking the integral over any subinterval into a sum of integrals over type 1 and type 2 subintervals. Since the integral over a type 1 subinterval is $\Delta x \bar{u}_i$, it suffices to demonstrate how to compute a high order reconstruction of the integral over a type 2 subinterval.

To get the high order reconstruction of the integral over a type 2 subinterval a WENO scheme is used. Define $m+1$ polynomials, p_0, \dots, p_m , of degree m such that

$$\int_{I_i} p_k dx = \Delta x \bar{u}_i, \quad i = -m + k, \dots, k, \quad (4.22)$$

as shown in Figure 2.1. The value of m is 1 or 2 for the 3rd and 5th order schemes respectively. We denote P_k as the integral of p_k over a type 2 subinterval for the 3rd order scheme these polynomials are

$$P_0 = \frac{\alpha \Delta x}{2} (\bar{u}_{i-1}(1 - \alpha) + \bar{u}_i(1 + \alpha)), \quad (4.23)$$

$$P_1 = \frac{-\alpha \Delta x}{2} (\bar{u}_i(-3 + \alpha) + \bar{u}_{i+1}(1 - \alpha)), \quad (4.24)$$

and for the 5th order scheme they are

$$P_0 = \frac{\alpha \Delta x}{6} (\bar{u}_{i-2}(-1 + \alpha^2) + \bar{u}_{i-1}(5 - 3\alpha - 2\alpha^2) + \bar{u}_i(2 + 3\alpha + \alpha^2)), \quad (4.25)$$

$$P_1 = \frac{\alpha \Delta x}{6} (\bar{u}_{i-1}(2 - 3\alpha + \alpha^2) + \bar{u}_i(5 + \alpha - 2\alpha^2) + \bar{u}_{i+1}(-1 + \alpha^2)), \quad (4.26)$$

$$P_2 = \frac{\alpha \Delta x}{6} (\bar{u}_i(11 - 6\alpha + \alpha^2) + \bar{u}_{i+1}(-7 + 9\alpha - 2\alpha^2) + \bar{u}_{i+2}(2 - 3\alpha + \alpha^2)). \quad (4.27)$$

Linear weighting coefficients are chosen such that

$$\sum_{k=0}^m c_k P_k - \int_{x_i}^{x_{i+\alpha}} u(x) dx = O(\Delta x^{2m+1}). \quad (4.28)$$

For the 3rd order scheme the linear coefficients are

$$c_0 = (2 - \alpha)/3 \quad \text{and} \quad c_1 = (1 + \alpha)/3. \quad (4.29)$$

For the fifth order scheme the linear coefficients are

$$c_0 = (-3 + \alpha)(-2 + \alpha)/20, \quad (4.30)$$

$$c_1 = (3 - \alpha)(2 + \alpha)/10, \quad (4.31)$$

$$c_2 = (1 + \alpha)(2 + \alpha)/20. \quad (4.32)$$

The nonlinear weighting polynomials $w_k(x)$ are defined as a function of the linear coefficients and of a set of smoothness indicators IS as

$$w_k = \frac{\sigma_k}{\sum_{i=0}^m \sigma_i} \quad \text{with} \quad \sigma_i = \frac{c_i}{(\epsilon + IS_i)^p} \quad \text{for} \quad k = 0, \dots, m. \quad (4.33)$$

Here $p \geq 1$ is an integer and $0 < \epsilon \ll 1$ is to avoid dividing by zero. The standard smoothness indicator for a WENO scheme are chosen, these are defined as

$$\sum_{l=1}^m \int_I h^{2l-1} \left(p_k^{(l)} \right)^2 dx. \quad (4.34)$$

The smoothness indicators for the third order scheme are

$$IS_0 = (\bar{u}_{i-1} - \bar{u}_i)^2, \quad (4.35)$$

$$IS_1 = (\bar{u}_i - \bar{u}_{i+1})^2, \quad (4.36)$$

and for the fifth order scheme they are

$$IS_0 = \frac{13}{12}(\bar{u}_{i-2} - 2\bar{u}_{i-1} + \bar{u}_i)^2 + \frac{1}{4}(\bar{u}_{i-2} - 4\bar{u}_{i-1} + 3\bar{u}_i)^2, \quad (4.37)$$

$$IS_1 = \frac{13}{12}(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1})^2 + \frac{1}{4}(\bar{u}_{i-1} - \bar{u}_{i+1})^2, \quad (4.38)$$

$$IS_2 = \frac{13}{12}(\bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2})^2 + \frac{1}{4}(\bar{u}_i - 4\bar{u}_{i+1} + 3\bar{u}_{i+2})^2. \quad (4.39)$$

The high order approximation of the mass in a type 2 subinterval is

$$\int_{x_i}^{x_i+\alpha} u(x) dx \approx \sum_{k=0}^m w_k P_k, \quad (4.40)$$

which is computed using equations (4.23)-(4.39). This completes the details for the high order mass integration over the trace region \check{E} .

4.2.2 Flux Correction

The flux correction is computed by solving equation (4.15). The value of u changes along the side S in time according to the derivative

$$\frac{du(x(t), t)}{dt} = \frac{\partial u}{\partial x} \frac{dx}{dt} + \frac{\partial u}{\partial t}. \quad (4.41)$$

Where along S the relation between x and t is given by equation (4.16) and $\frac{dx}{dt} = v$. Also $u_t = -f(u)_x$ substituting these gives

$$\frac{du(x(t), t)}{dt} = u_x v - f(u)_x. \quad (4.42)$$

This equation is solved numerically using a NCE Runge-Kutta scheme. The value of u is used to evolved along S according to

$$u((\tilde{x}(t^n), t^n) + \theta \Delta t(v, 1)) = u(\tilde{x}(t^n), t^n) + \Delta t \sum_{i=1}^v b_i(\theta) g^{(i)} \quad (4.43)$$

$$g^{(i)} = F \left((\tilde{x}(t^n), t^n) + c_i \Delta t(v, 1), u(\tilde{x}(t^n), t^n) + \Delta t \sum_{j=1}^v a_{ij} g^{(j)} \right) \quad (4.44)$$

$$F((x, t), u(x, t)) = (u_x v - f_x(u)). \quad (4.45)$$

The matrix a and the vector $b(\theta)$ for the third order scheme are

$$b_1(\theta) = -1/2\theta^2 + \theta, \quad b_2(\theta) = 1/2\theta^2 \quad (4.46)$$

and

$$a = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}. \quad (4.47)$$

For the fifth order scheme they are

$$b_1(\theta) = 2/3\theta^3 - 3/2\theta^2 + \theta, \quad b_2(\theta) = -2/3\theta^3 + \theta^2, \quad (4.48)$$

$$b_3(\theta) = b_2(\theta), \quad b_4(\theta) = 2/3\theta^3 - 1/2\theta^2, \quad (4.49)$$

and

$$a = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (4.50)$$

To use this Runge-Kutta scheme the values of u are first reconstructed on a local stencil. For the trace point \tilde{x} the stencil is $\tilde{x} + \gamma\Delta x$ where for the 3rd order scheme

$$\gamma = -1/2, 0, 1/2 \quad (4.51)$$

and for the 5th order scheme

$$\gamma = -1/2, -1/4, 0, 1/4, 1/2. \quad (4.52)$$

At each point on the stencil the concentration $u(\tilde{x} + \gamma\Delta x, t^n)$ is evaluated using an IWENO reconstruction over the elements E_{k-m}, \dots, E_{k+m} where $\tilde{x} \in E_k$, here m is the degree of the lower order polynomial used in the reconstruction.

In order to compute u_x and $f(u)_x$ from these points a nodal variation of the IWENO scheme is used. The following explains the computation of u_x and it is straightforward to use the points $f(u)$ in the same processes to get $f(u)_x$. Polynomials $P_k(x)$ are constructed such that

$$P_k(x_i) = u_i, \quad i = -m + k, \dots, k + 1, \quad (4.53)$$

and linear coefficients $c_k(x)$ are found optimizing the value of the reconstruction at the point $x \in E_k$

$$\sum_{k=0}^m c_k(x) P_k(x) = u(x) + O(\Delta x^{2m+1}). \quad (4.54)$$

The results in chapter 2 guarantee the existence of these polynomials and coefficients and provides an algorithm for computing them. Since this is a nodal based interpolation instead of a cell average based interpolation these

polynomials are different than those presented in Chapter 2, for completeness they are listed at the end of this section. Following the IWENO strategy we take the derivative of the reconstruction

$$\sum_{k=0}^m c'_k(x)P_k(x) + c_k(x)p_k(x) = u'(x) + O(\Delta x^{2m}) \quad (4.55)$$

where $p_k(x) = P'_k(x)$. The weighting coefficients are modified as in (4.33) with smoothness indicators

$$\sum_{l=1}^m \int_I h^{2l-1} \left(p_k^{(l)} \right)^2 dx. \quad (4.56)$$

This gives the reconstruction

$$u'(x) \approx \sum_{k=0}^m w'_k(x)P_k(x) + w_k(x)p_k(x) \quad (4.57)$$

which is evaluated at all the points on the stencil $\tilde{x} + \gamma\Delta x$. These values are used to advance the Runge-Kutta Scheme to the next stage where the same procedure is applied.

The polynomials, linear coefficients, and smoothness indicators for the 3rd order scheme are

$$P_0 = (1 + \alpha)u_i - \alpha u_{i-1}, \quad p_0 = (u_i - u_{i-1})/h, \quad (4.58)$$

$$P_1 = (1 - \alpha)u_i + \alpha u_{i+1}, \quad p_1 = (u_{i+1} - u_i)/h, \quad (4.59)$$

$$c_0 = (1 - \alpha)/2, \quad c'_0 = -1/2, \quad (4.60)$$

$$c_1 = (\alpha + 1)/2, \quad c'_1 = 1/2, \quad (4.61)$$

$$IS_0 = (u_i - u_{i-1})^2, \quad \text{and} \quad IS_1 = (u_{i+1} - u_i)^2. \quad (4.62)$$

For the fifth order scheme they are

$$P_0 = \frac{1}{2}(\alpha(\alpha + 1)u_{i-2} - 2\alpha(\alpha + 2)u_{i-1} + (\alpha + 1)(\alpha + 2)u_i), \quad (4.63)$$

$$P_1 = \frac{1}{2}(-2(\alpha^2 - 1)u_i + (\alpha - 1)\alpha u_{i-1} + \alpha(\alpha + 1)u_{i+1}), \quad (4.64)$$

$$P_2 = \frac{1}{2}((\alpha - 2)((\alpha - 1)u_i - 2\alpha u_{i+1}) + (\alpha - 1)\alpha u_{i+2}), \quad (4.65)$$

$$p_0 = \frac{1}{2h}((2\alpha + 1)u_{i-2} - 4(\alpha + 1)u_{i-1} + (2\alpha + 3)u_i), \quad (4.66)$$

$$p_1 = \frac{1}{2h}((2\alpha - 1)u_{i-1} + -4\alpha u_i + (2\alpha - 1)u_{i+1}), \quad (4.67)$$

$$p_2 = \frac{1}{2h}((2\alpha - 3)u_i - 4(\alpha - 1)u_{i+1} + (2\alpha - 1)u_{i+2}), \quad (4.68)$$

$$c_0 = \frac{1}{12}(\alpha - 2)(\alpha - 1), \quad c'_0 = (2\alpha - 3)/(12h), \quad (4.69)$$

$$c_1 = -\frac{1}{6}(\alpha - 2)(\alpha + 2), \quad c'_1 = -\alpha/(3h), \quad (4.70)$$

$$c_2 = \frac{1}{12}(\alpha + 1)(\alpha + 2), \quad c'_2 = (2\alpha + 3)/(12h), \quad (4.71)$$

$$IS_0 = \frac{13}{12}(\bar{u}_{i-2} - 2\bar{u}_{i-1} + \bar{u}_i)^2 + \frac{1}{4}(\bar{u}_{i-2} - 4\bar{u}_{i-1} + 3\bar{u}_i)^2 \quad (4.72)$$

$$IS_1 = \frac{13}{12}(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1})^2 + \frac{1}{4}(\bar{u}_{i-1} - \bar{u}_{i+1})^2 \quad (4.73)$$

$$IS_2 = \frac{13}{12}(\bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2})^2 + \frac{1}{4}(\bar{u}_i - 4\bar{u}_{i+1} + 3\bar{u}_{i+2})^2. \quad (4.74)$$

The 3rd order scheme uses $\alpha = -1, 0, 1$ and $h = \Delta x/2$; the 5th order scheme uses $\alpha = -2, -1, 0, 1, 2$ and $h = \Delta x/4$.

4.2.3 The Relaxed CFL Condition

For the nonlinear problem there are 2 natural choices for the approximate trace velocity \tilde{v} . The first is $\tilde{v} = f'(u)$ for some fixed value of u . Since u is constant along characteristics, this choice should give a small error in the flux correction step. The nonlinear problem can admit solutions where the characteristics collide and if no CFL constraint is applied the scheme will collapse. The standard CFL constraint for a fixed mesh method limits the time step to the time when characteristics at grid point x_i arrives at the next grid point x_{i+1} or x_{i-1} this time step is given as $\Delta t \leq \frac{\Delta x}{\max |f'(u)|}$. The related problem $(u_t + (vu)_x) + (f(u) - vu)_x$ has a relaxed CFL condition

$$\Delta t \leq \Delta t_{\text{CFL}} := \frac{h}{\max |f'(u) - v|} \quad (4.75)$$

as illustrated in Figure 4.4 This CFL condition was identified by Stockie Mackenzie and Russell [73] where they used a velocity perturbation to solve their Moving Mesh Method. Later used by Arbogast and Huang in [46] note that this is the same condition needed for the flux corrected Eulerian-Lagrangian schemes. This condition states that the time step is small enough that waves from neighboring cells don't intersect across tracelines.

A second choice for \tilde{v} is the particle velocity field $f(u)/u$ where again $u(x, t)$ is fixed. This minimized the number of particles flowing across the boundary, and again should lead to a small flux correction.

The choice of \tilde{v} depends on the solution of the given problem, however for the problems we ran numerical tests on there was no significant difference.

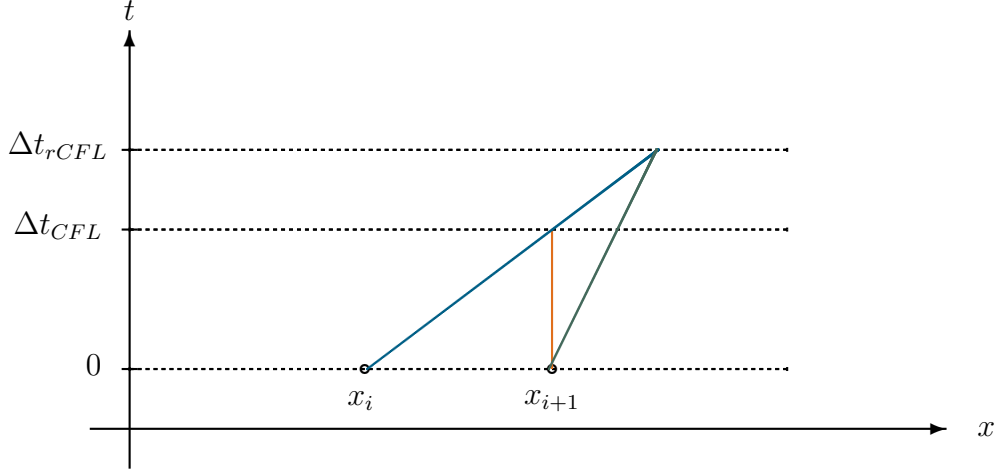


Figure 4.4: **Relaxed CFL Constraint** An illustration of relaxed CFL condition, orange line is fixed grid location, blue line has slope $\max(f'(u))^{-1}$, green line has slope v^{-1} . The fixed grid CFL condition is when the wave reaches the next grid point. The relaxed CFL condition describes the time when the wave reaches the Lagrangian traceline.

The numerical results for this dissertation are presented with $\tilde{v}_i = f'(u_i)$.

4.3 Numerical Results

We follow the numerical examples provided by Arbogast and Huang in [46] since this method differs only in the computation of the flux integral it provides the natural comparison for studying the changes to the scheme. In their paper they demonstrate that without computing the flux term, inaccurate particle tracing leads to both a lack of accuracy and poor performance at capturing shocks. In this section we will not reproduce results of this type, we will simply analyze the performance of this algorithm in comparison with their scheme. Numerical results will first be presented for the linear problem, followed by the non-linear problem. Error results are presented with discrete L^1 and L^∞ norms as given in equations (3.24) and (3.25).

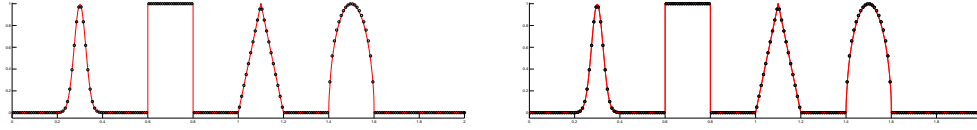


Figure 4.5: **Shu's Linear Test** Numerical solutions for Shu's Linear Test at time $t = 2$ with 4 time steps. On the left is the 3rd order scheme, and on the right is the 5th order scheme.

We first present results to modeling the linear transport problem. This is useful for the following reasons, first it illustrates the scheme in a context that is simple to interpret. The second reason is that exactly computing tracers while taking large time steps can be an expensive procedure. As a way to minimize the computational effort one can trace through the approximated velocity field and then compute the flux integral provided in this chapter.

4.3.0.1 Constant Speed Transport

We first test our scheme on the linear problem $u_t + u_x = 0$ with analytic solution $u(x, t) = u_0(x - t)$. Since trace points are simply $x - \Delta t$ we randomly perturb these points with a uniform distribution in $[-0.2\Delta x, 0.2\Delta x]$, so that the flux across the tracerline is nonzero. We apply this to the initial condition given by Shu's Linear Test as described in Section 3.2.4.4. The test is run on a mesh with 200 elements. The results at $T = 2$ using 4 time steps shown in Figure 4.5 show no visible oscillations or diffusion.

N	Error L^1	Order	Error L^∞	Order
3 rd order scheme				
80	1.74 E-1	-	8.33 E-1	-
160	2.06 E-3	6.40	6.97 E-3	6.90
320	4.36 E-5	5.56	5.43 E-4	3.68
640	6.63 E-6	2.71	1.22 E-4	2.15
1280	4.73 E-7	3.81	1.20 E-5	3.34
2560	2.48 E-8	4.26	4.45 E-7	4.76
5120	1.50 E-9	4.05	1.39 E-8	5.00
10240	9.72 E-11	3.95	1.40 E-9	3.30
LR		4.21		3.96
5 th order scheme				
40	2.25 E-3	-	1.77 E-3	-
80	4.08 E-5	5.79	3.20 E-5	5.79
160	6.80 E-7	5.91	5.34 E-7	5.90
320	1.08 E-8	5.96	8.58 E-9	5.96
640	1.73 E-10	5.98	1.36 E-10	5.98
1280	3.48 E-12	5.63	2.62 E-11	2.37

Table 4.1: **Time Dependent Transport Test 1D** Results for the flux corrected EL-WENO method with $v = \sin(t)$.

4.3.0.2 Time Dependent Velocity

This example uses the velocity $v = \sin(t)$. The initial condition is again $u_0(x) = \sin(\pi x)$ for $x \in [0, 2]$. The analytical solution to this problem is $u(x, t) = u_0(x + 1 + \cos(t))$. The trace points are evaluated using RK4 with a single time step. The error results at $t = 4$ using $\Delta t = 20\Delta x$ are listed in Table 4.1 show high convergence rates. The 5th order results here are comparable to those of the EL-WENO method of Arbogast and Huang who also see a converge rate at around 6 for this problem.

4.3.0.3 Space Dependent Velocity

For this test the velocity is given as $v = \sin(x)$ with an initial condition that $u_0(x) = 1$ for $x \in [0, 2\pi]$. The analytical solution to this problem is

$$u(x, t) = \frac{\sin(2 \arctan(e^{-t} \tan(x/2)))}{\sin(x)}. \quad (4.76)$$

These results are presented in Table 4.2 at $t = 1$ with $\Delta t = 5\Delta x$. The trace points are evaluated using a single time step of RK4. Numerical results confirm formally high order convergence. The results of the 5th order scheme are similar in magnitude to the method of Arbogast and Huang.

4.3.0.4 Burgers' Equation

The final numerical example we give is for Burgers' equation $u_t + (1/2u^2)_x = 0$. We first solve this equation for the initial condition

$$u_0(x) = \begin{cases} 1.0, & x \in [0.6, 1.4], \\ 0.5, & \text{otherwise} \end{cases} \quad (4.77)$$

which develops shock and rarefaction waves. Numerical results are shown in Figure 4.6 on a grid with 100 points at time $t = 0.5$ with $\Delta t = .02$. The 5th order scheme has better resolution near the rarefaction wave than the 3rd order scheme.

The second initial condition we use is $u(x, 0) = u_0(x) = 0.75 + 0.25 \sin(\pi x)$ for $x \in [0, 2]$. The mass forms a shock at time $t = 4/\pi$. Numerical results in Table 4.3 are computed before the shock forms at $t = 1$ with $\Delta t = 4\Delta x$. The analytical solution is computed numerically by solving the fixed point problem

$$u(x, t) - (0.75 + 0.25 \sin(\pi \tilde{x})) = 0 \quad (4.78)$$

N	Error L^1	Order	Error L^∞	Order
3 rd order scheme				
20	9.63 E-3	-	5.11 E-2	-
40	5.25 E-3	0.87	4.41 E-3	.212
80	1.08 E-3	2.28	1.86 E-3	1.25
160	1.75 E-4	2.62	5.65 E-4	1.72
320	2.61 E-5	2.74	1.52 E-4	1.89
640	3.33 E-6	2.97	3.77 E-5	2.01
1280	4.23 E-7	2.98	7.54 E-6	2.32
2560	5.28 E-8	3.00	9.25 E-7	3.02
5 th order scheme				
20	9.40 E-4	-	6.94 E-4	-
40	1.47 E-4	2.67	1.60 E-4	2.11
80	1.01 E-5	3.87	1.75 E-5	3.19
160	4.25 E-7	4.57	7.45 E-7	4.55
320	1.51 E-8	4.81	2.61 E-8	4.83
640	5.05 E-10	4.91	8.64 E-10	4.92
1280	1.65 E-11	4.94	2.79 E-11	4.95

Table 4.2: **Spatially Dependent Transport Test 1D** Results for the flux corrected EL-WENO method with $v = \sin(x)$.

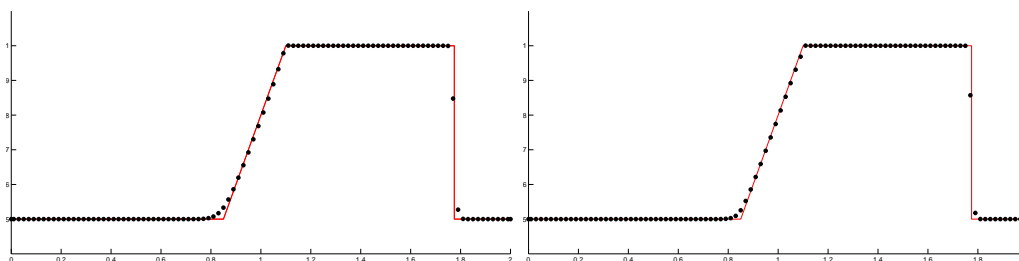


Figure 4.6: **Burgers' Equation: Example 1** Numerical solutions for Burgers' equation at time $t = 0.5$. On the left is the 3rd order scheme, and on the right is the 5th order scheme. The red is the analytical solution.

N	Error L^1	Order	Error L^∞	Order
3 rd order scheme				
40	8.34 E-4	-	7.43 E-3	-
80	1.51 E-4	2.47	1.56 E-3	2.25
160	1.91 E-5	2.95	2.96 E-4	2.40
320	2.22 E-6	3.13	6.79 E-5	2.13
640	2.90 E-7	2.93	1.09 E-5	2.64
5 th order scheme				
40	2.13 E-4	-	1.89 E-3	-
80	8.34 E-6	4.67	1.12 E-4	4.08
160	6.99 E-7	3.58	1.35 E-5	3.05
320	2.87 E-8	4.60	7.34 E-7	4.21
640	1.17 E-9	4.62	3.21 E-8	4.51

Table 4.3: **Burgers' Equation** Results for the flux corrected EL-WENO method.

for the trace location \tilde{x} using a Newton iteration. This is solved at quadrature points x on the element which are then used to compute the average value of u used in the error equation. The code implements a 5 point Gaussian quadrature rule for this computation. The code is then tested after the shock formation at $t = 2$ with $\Delta t = 2\Delta x$ the results in Figure 4.7 show that the method captures a sharp front at the shock.

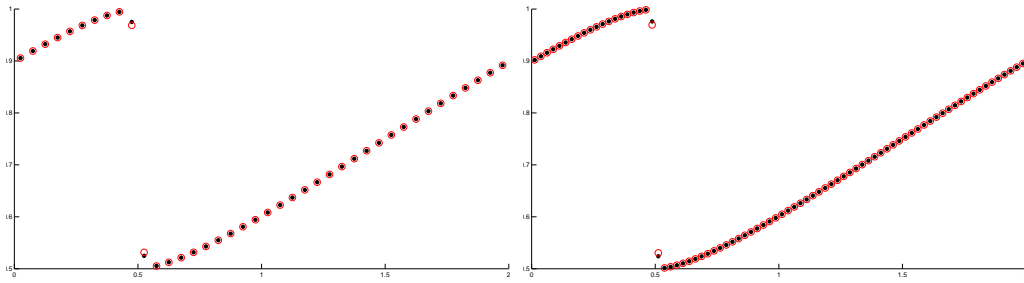


Figure 4.7: **Burgers' Equation: Example 2** Numerical solutions for Burgers' equation at time $t = 1$. On the left is a grid with 40 elements and on the right there are 80 elements. The black dots are the solution of the 3rd order scheme and the red circles are for the 5th order scheme.

Chapter 5

QEL-WENO for Nonlinear Advection in 2D

In this chapter we will combine the ideas of the previous chapters in order to solve the 2 dimensional nonlinear conservation problem

$$u_t + \nabla \cdot f(u) = 0, \quad (5.1)$$

$$u(x, y, t = 0) = u_0(x, y), \quad (5.2)$$

where $f(u) = (f_1(u), f_2(u))$.

5.1 The Space-Time Region, $\mathcal{E}(v)$

We define the approximate velocity field by taking an approximate characteristic tracing at the boundary nodes. To solve this we define an approximated velocity field $\tilde{v}(x, y, t)$, that is specific to the flux function $f(u)$. We apply a characteristic tracing, at the nodes of the grid x_i, y_j through the chosen approximate velocity field $\tilde{v}(x, y, t)$:

$$(\xi_1, \xi_2)_t = (\tilde{v}_1, \tilde{v}_2), \quad (5.3)$$

$$\xi(t^{n+1}) = (x_i, y_j). \quad (5.4)$$

This tracing defines the traceback points $(\tilde{x}_i, \tilde{y}_j) = \xi(t^n)$. We define the velocity v at the nodes as the constant value $v_{i,j} = \left(\frac{x_i - \tilde{x}_i}{\Delta t}, \frac{y_j - \tilde{y}_j}{\Delta t} \right)$. These are

used to define the velocity field over the element $E_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ as the bilinear interpolation

$$\begin{aligned} v(x_{i+\alpha}, y_{j+\beta}) &= (1 - \beta)(1 - \alpha)v_{i,j} + (1 - \beta)\alpha v_{i+1,j} \\ &\quad + \beta(1 - \alpha)v_{i,j+1} + \alpha\beta v_{i+1,j+1} \end{aligned} \quad (5.5)$$

for $(\alpha, \beta) \in [0, 1]^2$. The velocity field on E_{ij} has two components $v(x, y) = (a_1(x, y), a_2(x, y))$. This defines the space-time region $\mathcal{E} = \mathcal{E}(v)$ and traceback region $\check{E} = \check{E}(v)$ as a function of the velocity field.

5.2 The Flux Corrected Problem

We now solve the related problem

$$(u_t + \nabla \cdot (a_1 u, a_2 u)) + \nabla \cdot (f_1(u) - a_1 u, f_2(u) - a_2 u) = 0, \quad (5.6)$$

$$u(x, y, t = 0) = u_0(x, y). \quad (5.7)$$

Integrating over \mathcal{E} gives

$$\int_{\mathcal{E}} u_t + \nabla \cdot \begin{pmatrix} a_1 u \\ a_2 u \end{pmatrix} dV + \int_{\mathcal{E}} \nabla \cdot \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \end{pmatrix} dV = 0. \quad (5.8)$$

Applying the divergence theorem over the space time region this equation becomes

$$\int_E u^{n+1} dA - \int_{\check{E}} u^n dA + \oint_{\partial\mathcal{E}} \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \nu_x \\ \nu_y \\ \nu_t \end{pmatrix} dS \quad (5.9)$$

The mass on cell E at t^{n+1} is given by

$$\int_E u^{n+1} dA = \int_{\check{E}} u^n dA - \oint_{\partial\mathcal{E}} \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \nu_x \\ \nu_y \\ \nu_t \end{pmatrix} dS. \quad (5.10)$$

In chapter 3 we gave the algorithm to solve the integral over \check{E} using a quadrature scheme. We remark that since the velocity field at every point is constant we do not need to apply an ODE solver to compute the trace points as they are simply $(\check{x}, \check{y}) = (x, y) - v(x_{i+\alpha}, y_{j+\beta})\Delta t$ for $(x, y) \in E_{i,j}$. The Jacobian can be computed directly using the linear trace and

$$\frac{\partial v}{\partial x} = \frac{1}{\Delta x} [(1 - \beta)(v_{i+1,j} - v_{i,j}) + \beta(v_{i+1,j+1} - v_{i,j+1})], \quad (5.11)$$

$$\frac{\partial v}{\partial y} = \frac{1}{\Delta y} [(1 - \alpha)(v_{i,j+1} - v_{i,j}) + \alpha(v_{i+1,j+1} - v_{i+1,j})] \quad (5.12)$$

to solve for the components $\frac{\partial \check{x}}{\partial y}$, $\frac{\partial \check{x}}{\partial x}$, $\frac{\partial \check{y}}{\partial y}$ and $\frac{\partial \check{y}}{\partial x}$ directly.

This leaves a method to solve for

$$\oint_{\partial \mathcal{E}} \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \nu_x \\ \nu_y \\ \nu_t \end{pmatrix} dS = \sum_{i=0}^3 \int_{S_i} \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \nu_x \\ \nu_y \\ \nu_t \end{pmatrix} dS \quad (5.13)$$

which will use the technique detailed in Chapter 4. However we will be explicit about this extension to higher dimensions. We note that the faces E and \check{E} have outward normals $\pm(0, 0, 1)$ and do not contribute to this sum, which is over the 4 remaining surfaces.

5.2.1 Surface Parameterization

The goal of this section is to define a parameterization of the Surface S_i of the form

$$r(\gamma, t) = x(\gamma, t) \hat{\mathbf{i}} + y(\gamma, t) \hat{\mathbf{j}} + t \hat{\mathbf{k}}, \quad (5.14)$$

where $(\gamma, t) \in [0, 1] \times [t^n, t^{n+1}] = D$. This allows us compute (5.13) as

$$\int_D \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \nu_{x(\gamma, t)} \\ \nu_{y(\gamma, t)} \\ \nu_t \end{pmatrix} \|r_\gamma \times r_t\| dA. \quad (5.15)$$

The outward unit normal on a surface with a parameterization $r(\gamma, t)$ is given by

$$\nu = \frac{r_\gamma \times r_t}{\|r_\gamma \times r_t\|}, \quad (5.16)$$

which gives the flux integral over D as

$$\int_D \begin{pmatrix} f_1(u) - a_1 u \\ f_2(u) - a_2 u \\ 0 \end{pmatrix} \cdot (r_\gamma \times r_t) dA. \quad (5.17)$$

We will solve this using quadrature for both γ and t .

This parameterization is defined on each surface independently. Let (x_p, y_p) and (x_q, y_q) be adjacent nodes of $E_{i,j}$. We will consider the face described by these 2 points and their traceback points $(\check{x}_p, \check{y}_p)$ and $(\check{x}_q, \check{y}_q)$. The surface $r(\gamma, t) = (x(\gamma, t), y(\gamma, t), t)$ is described by

$$\begin{aligned} x(\gamma, t) &= x_p(1 - \gamma) + x_q \gamma \\ &\quad - \left(\frac{(x_p - \check{x}_p)}{\Delta t} (1 - \gamma) + \frac{(x_q - \check{x}_q)}{\Delta t} \gamma \right) (t^{n+1} - t), \end{aligned} \quad (5.18)$$

$$\begin{aligned} y(\gamma, t) &= y_p(1 - \gamma) + y_q \gamma \\ &\quad - \left(\frac{(y_p - \check{y}_p)}{\Delta t} (1 - \gamma) + \frac{(y_q - \check{y}_q)}{\Delta t} \gamma \right) (t^{n+1} - t). \end{aligned} \quad (5.19)$$

The partial derivatives of r with respect to γ and t are

$$\begin{aligned}
r_\gamma &= \frac{\partial x(\gamma, t)}{\partial \gamma} \hat{\mathbf{i}} + \frac{\partial y(\gamma, t)}{\partial \gamma} \hat{\mathbf{j}} + \frac{\partial t}{\partial \gamma} \hat{\mathbf{k}} \\
&= (x_q - x_p) + \left(\frac{(x_p - \check{x}_p)}{\Delta t} - \frac{(x_q - \check{x}_q)}{\Delta t} \right) (t^{n+1} - t) \hat{\mathbf{i}} \\
&\quad + (y_q - y_p) + \left(\frac{(y_p - \check{y}_p)}{\Delta t} - \frac{(y_q - \check{y}_q)}{\Delta t} \right) (t^{n+1} - t) \hat{\mathbf{j}} \\
&\quad + 0 \hat{\mathbf{k}}
\end{aligned} \tag{5.20}$$

and

$$\begin{aligned}
r_t &= \frac{\partial x(\gamma, t)}{\partial t} \hat{\mathbf{i}} + \frac{\partial y(\gamma, t)}{\partial t} \hat{\mathbf{j}} + \frac{\partial t}{\partial t} \hat{\mathbf{k}} \\
&= \left(\frac{(x_p - \check{x}_p)}{\Delta t} (1 - \gamma) + \frac{(x_q - \check{x}_q)}{\Delta t} \gamma \right) \hat{\mathbf{i}} \\
&\quad + \left(\frac{(y_p - \check{y}_p)}{\Delta t} (1 - \gamma) + \frac{(y_q - \check{y}_q)}{\Delta t} \gamma \right) \hat{\mathbf{j}} \\
&\quad + 1 \hat{\mathbf{k}}.
\end{aligned} \tag{5.21}$$

Using these the flux integral over the surface is the integral

$$\int_D (f_1(u) - a_1 u)(r_\gamma \cdot (0, 1, 0)) + (f_2(u) - a_2 u)(r_\gamma \cdot (-1, 0, 0)) dA. \tag{5.22}$$

5.2.2 Quadrature

We solve (5.22) with a quadrature rule in space γ and time t

$$\begin{aligned}
&\sum_{k=1}^{n_q} \tilde{w}_k \sum_{l=1}^{n_q} w_l \left((f_1(u) - a_1 u)(r_\gamma \cdot (0, 1, 0)) \right. \\
&\quad \left. + (f_2(u) - a_2 u)(r_\gamma \cdot (-1, 0, 0)) \right),
\end{aligned} \tag{5.23}$$

where the function f and the mass u and the vector r are functions of the quadrature points γ_k and t_l . To compute this we need to know the value of $u(x(\gamma_k, t_l), y(\gamma_k, t_l), t_l)$. We compute this term using the strategy in Chapter 4. We note that for a fixed γ , the mass u changes in time along the line $(x(t), y(t), t)$ as

$$\begin{aligned} \frac{du}{dt} &= \frac{\partial u}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial u}{\partial t} \\ &= u_x a_1 + u_y a_2 - f_1(u)_x - f_2(u)_y \end{aligned} \quad (5.24)$$

This is solved using the NCE Runge Kutta Scheme in Chapter 4. The details of the IWENO scheme are presented in Chapter 2, taking the tensor product of the 1D polynomials given in Chapter 4. The smoothness indicators are defined in the standard way.

5.3 Numerical Results

5.3.0.1 Example: Swirling Deforming Flow

Two dimensional problem with a swirling and deforming velocity field as described in section 3.2.5.3 is applied. A Runge-Kutta scheme using a single time step is used to trace the element nodes. Figure 5.1 shows the numerical results for 10 times steps on a grid with 60 elements in both directions. The error results listed in Table 5.1 show that for coarse grids we get the similar error results as the linear QEL-WENO method. The formal convergence rates are not seen here possibly due to rounding error accumulation or a subtle bug in the code.

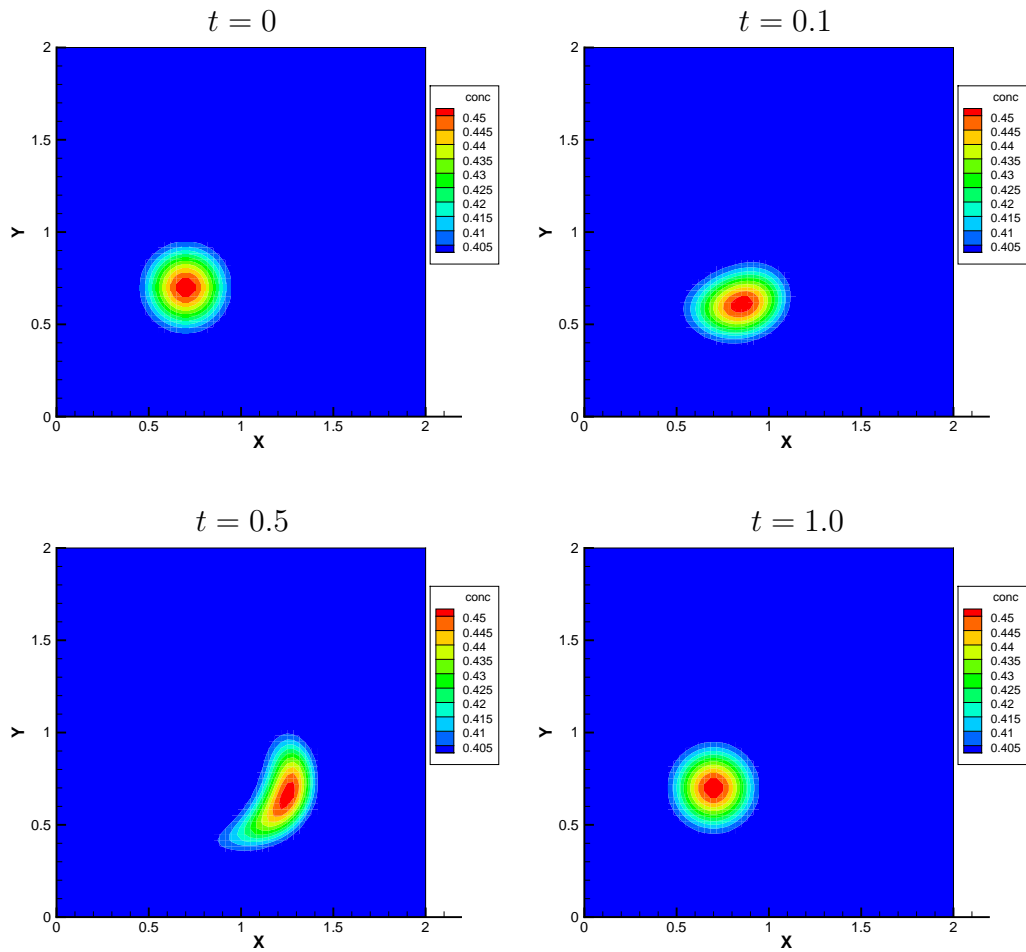


Figure 5.1: **Flux Corrected Swirling and Deforming Flow** Numerical results for a radial bump function in a swirling and deforming velocity field at various times with $\Delta t = 0.1$.

N	Error L^1	Order	Error L^∞	Order
10	3.20 E-3	-	2.64 E-2	-
20	1.03 E-3	1.63	8.44 E-3	1.64
40	1.10 E-4	3.22	9.06 E-4	3.21
60	3.47 E-5	2.84	6.41 E-4	0.85
80	3.70 E-5	-0.22	1.78 E-3	-3.54

Table 5.1: **Flux Corrected Swirling and Deforming Flow.** Error and convergence order for transport with a time and spatially dependent velocity field at time $t = 1$ with $\Delta t = 0.1$.

5.3.0.2 Example: Burgers' Equation

The final example is the two dimensional Burgers' Equation

$$u_t + (u^2/2)_x + (u^2/2)_y = 0, \quad (x, y) \in [0, 2]^2, t > 0, \quad (5.25)$$

with the initial condition

$$u(x, y, 0) = \begin{cases} \sin^2(\pi x) \sin^2(\pi y), & \text{for } (x, y) \in (0, 1)^2 \\ 0, & \text{otherwise.} \end{cases} \quad (5.26)$$

The problem is solved using the wave velocity $v_{ij} = u_{ij}$ for the approximate tracing. The solution with 40 grid elements in each direction and $\Delta t = \Delta x$ is shown in Figure 5.2 at times $t = 0, 1, 2$ and 3. The solution shows the method captures the sharp front of the shock.

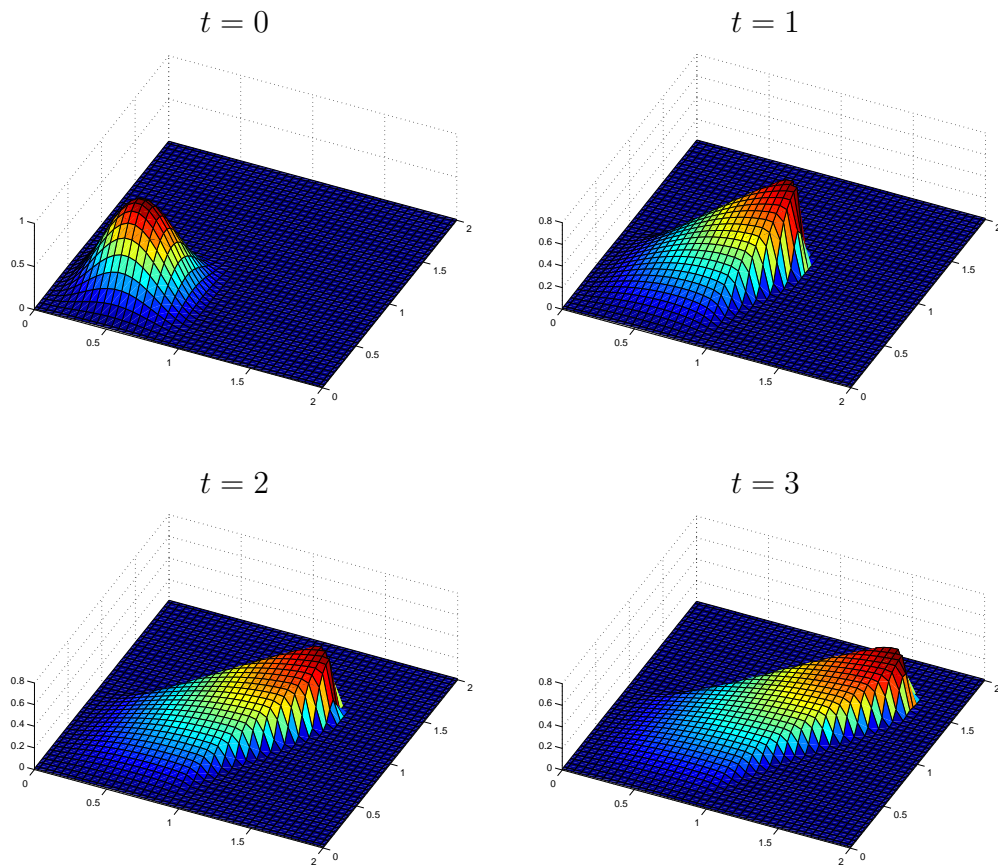


Figure 5.2: **Flux Corrected 2-D Burgers' Solution** at times $t = 0, 1, 2$ and 3 using 40 grid elements in each direction and $\Delta t = \Delta x$.

Chapter 6

Two Phase Flow

One application of the Quadrature Eulerian Lagrangian method is to simulate the transport part of the incompressible two phase flow problem. There is a broad set of literature [24, 57, 66, 67, 69, 74, 80, 81] developing the physical equations and analyzing numerical methods to solve flow and transport in porous media. This chapter will provide an overview of the two-phase flow problem and include details of our implementation.

Let $\alpha = w, n$ refer to the wetting and non-wetting phases of the fluid. The two-phase flow problem for the Darcy velocity u_α , pressure p_α and saturation s_α is described by the partial differential equations

$$u_\alpha = -K_a \lambda_{r\alpha}(s_\alpha)(\nabla p_\alpha - \rho_\alpha g) \quad \text{in } \Omega \times J, \quad (6.1)$$

$$\partial_t(\phi s_\alpha) + \nabla \cdot u_\alpha = q_a \quad \text{in } \Omega \times J, \quad (6.2)$$

$$u_\alpha \cdot \nu = 0 \quad \text{on } \partial\Omega \times J, \quad (6.3)$$

$$s_n(x, 0) = s_n^0(x) \quad \text{in } \Omega, \quad (6.4)$$

where Ω is the spatial domain, J is the time domain, K_a is the medium absolute permeability, Let $k_{r\alpha}$ is the relative permeability, μ_α is the viscosity and $\lambda_{r\alpha}$

is the relative mobility defined as

$$\lambda_{r\alpha} = \frac{k_{r\alpha}}{\mu_\alpha}, \quad (6.5)$$

q_α is the external source/sink, ν is the normal, ρ_α is the density of the fluid and g is the gravitational acceleration.

$$s_w + s_n = 1. \quad (6.6)$$

p is the pressure, We also consider the effects of capillary pressure, which is modeled as

$$p_c(s_w) = p_n - p_w. \quad (6.7)$$

6.1 Global Pressure Formulation

The global pressure formulation of the above flow equation was introduced for incompressible two phase flow in [1, 21], and extended as a method to solve three phase flow in [22, 23].

6.1.1 Pressure Equation

To minimize the coupling between the pressure and saturation, we use a global pressure formulation. This global pressure is

$$p = p(s_w) = p_0 - \int_0^{s_w} \frac{\lambda_{rw}}{\lambda_{rt}} p'_c ds. \quad (6.8)$$

We also define a total velocity, volumetric injection rate and mobility as

$$u_t = u_n + u_w, \quad (6.9)$$

$$q_t = q_n + q_w, \quad (6.10)$$

$$\lambda_{rt} = \lambda_{rn} + \lambda_{rw}. \quad (6.11)$$

By summing each phase in Darcy's Law (6.1) and the Mass Conservation (6.2) we get an elliptic pressure equation in mixed form

$$\nabla \cdot u_t = q_t \quad \text{in } \Omega \times J, \quad (6.12)$$

$$u_t = -K_a \lambda_{rt} \left(\nabla p - \frac{\lambda_{rw} \rho_w + \lambda_{rn} \rho_n}{\lambda_{rt}} g \right) \quad \text{in } \Omega \times J. \quad (6.13)$$

6.1.2 Saturation Equation

We solve for the wetting phase $\alpha = w$ and use (6.6) to solve for the non-wetting phase. Combining the mass conservation equation (6.2) with Darcy's Law (6.1) we derive the saturation equation as

$$\partial_t(\phi s_w) + \nabla \cdot \left(\frac{\lambda_{rw}}{\lambda_{rt}} u_t \right) + \nabla \cdot \left[a \frac{\lambda_{rn} \lambda_{rw}}{\lambda_{rt}} (\nabla p_c + (\rho_w - \rho_n) g) \right] = q_w. \quad (6.14)$$

6.2 Discretization

To discretize the problem we choose p , u_t , and s_w as the primary variables for solving equations (6.1)-(6.4). We use an implicit pressure explicit saturation (IMPES) scheme for solving the two phase problem.

6.2.1 Discretization of Flow

Mortar mixed finite elements are a well-known technique used to solve the flow problem. In [58, 59] a fully implicit formulation is used to solve a non-linear interface problem. This work was extended in [11, 37, 38, 83, 84] where preconditioning techniques are developed to aid in solving the problem. We use these types of techniques, and use code developed by Hailong Xiao; details of his method and can be found in [82].

For the examples provided we use Raviart-Thomas [64] spaces of lowest order (RT0). The space (V_h, W_h) is used to approximate

$$\left((K_a \lambda_{rt})^{-1} u_t, v \right) - (p, \nabla \cdot v) = - \left(\frac{\lambda_{rw} \rho_w + \lambda_{rn} \rho_n}{\lambda_{rt}} g, v \right) \quad \forall v \in V_h, \quad (6.15)$$

$$(\nabla \cdot u_t, w) = (q_t, w) \quad \forall w \in W_h. \quad (6.16)$$

6.2.2 Discretization of Transport

As discussed in Chapter 1, the two terms in the transport problem have very different behaviors. In order to solve this equation over a time step we use an operator splitting technique to solve for the hyperbolic part (advection) and parabolic part (diffusion) separately. We solve

$$\partial_t(\phi s_w) + \nabla \cdot \left(\frac{\lambda_{rw}}{\lambda_{rt}} u_t \right) = 0 \quad (6.17)$$

using the QEL-WENO techniques provided in Chapters 5. Then we use the solution as an initial condition to solve for

$$\partial_t(\phi s_w) + \nabla \cdot \left[a \frac{\lambda_{rn} \lambda_{rw}}{\lambda_{rt}} (\nabla p_c + (\rho_w - \rho_n) g) \right] = q_w. \quad (6.18)$$

In order to deal with the degenerate diffusion coefficient $\alpha(s) = K_a \lambda_{rn} \lambda_{rw} / \lambda_{rt}$ we use a mixed finite element method [10, 25, 68] computed with a rectangle quadrature rule, or cell centered finite differencing. This method is given as

$$(\phi s^{n+1}, w) + (\Delta t \nabla \cdot \gamma, w) = (\phi s^n, w), \quad (6.19)$$

$$(\gamma, v)_Q = (\alpha(s)[p'_c(s) \hat{\gamma} + (\rho_w - \rho_n)g \nabla z], v)_Q, \quad (6.20)$$

$$(\hat{\gamma}, \omega)_Q = -(s, \nabla \cdot \omega)_Q. \quad (6.21)$$

6.3 Numerical Example

We run a simulation for 7 days taking time steps of 0.02 [days]. The computational domain is $(x, y) \in [0, 35] \times [0, 35]$ [m] solved on a uniform mesh with 35 elements in each spatial dimension. The initial condition reflects gravitation equilibrium with initial saturation set at 0.4 at a depth 35 [m]. The boundary has a no flow condition. The inflow source term is on the elements with $(x, y) \in [0, 1] \times [0, 32]$ [m] and the outflow is on $(x, y) \in [31, 32] \times [0, 32]$ [m] with respective flow rates of ± 10 [/day]. The rock porosity is constant at 0.2, and the permeability is given in Figure 6.1. This figure also shows the initial pressure and velocity field for the problem. The Brooks-Corey model [17, 18] is used to generate the capillary pressure and relative permeability curves

$$p_c = p_e s_w^{-1/\lambda}, \quad (6.22)$$

$$k_{rw} = s_w^{(2+3\lambda)/\lambda}, \quad (6.23)$$

$$k_{rn} = (1 - s_w)^2 (1 - s_w^{(2+\lambda)/\lambda}), \quad (6.24)$$

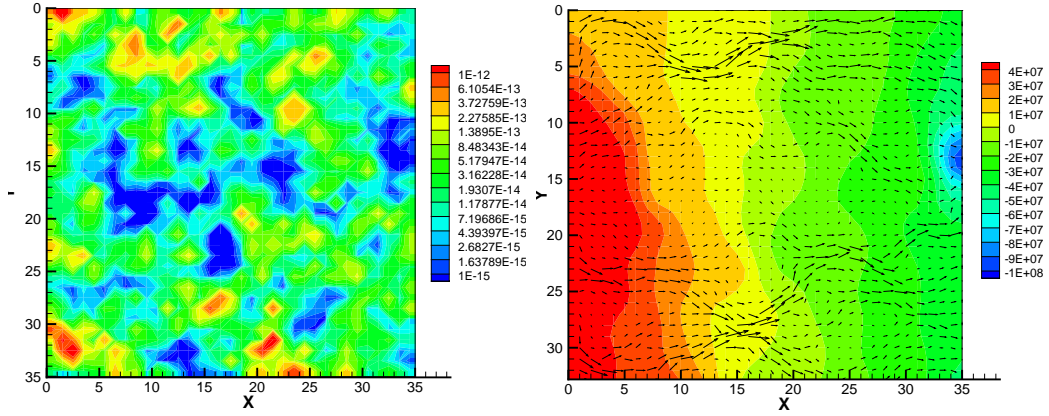


Figure 6.1: **Permeability and Flow** On the left is the permeability. On the right is the initial pressure and velocity.

where p_e is the entry pressure and λ is the pore-size distribution index. For all of the numerical examples provided we take $p_e = 0.5[\text{psi}]$ and $\lambda = 2$ as shown in Figure 6.2. The water and oil viscosities are $0.5[\text{cp}]$ and $2.5[\text{cp}]$ respectively. The densities of water and oil are respectively $1[\text{g}/\text{cm}^3]$ and $0.7[\text{g}/\text{cm}^3]$.

The numerical results of this simulation are shown in 6.3. These results show the fluid moving through the channels. The solution is not monotone, suggesting that a flux splitting technique might be needed for this problem. There are two improvements we suggest for future investigation. The first is to improve the velocity model to include a cross flow. This can be done using the higher order Raviart Thomas space RT1 or the Brezzi Douglas Marini space BDM1 [16]. The higher order velocity field will increase the accuracy of the particle tracing in the QEL-WENO scheme and allow for larger time steps to

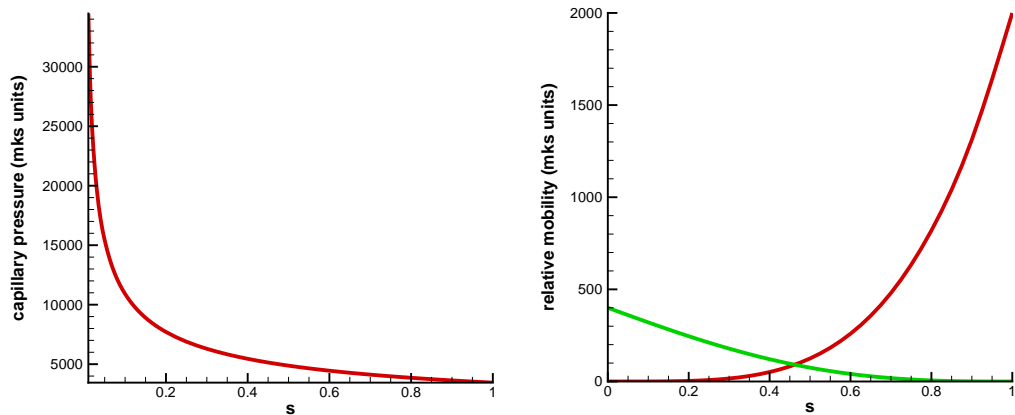


Figure 6.2: **Brooks-Corey Model** Capillary pressure is on left and relative mobilities are on the right.

be taken.

The second improvement is to change the implementation of the source term. Currently the source term is implemented in the diffusion equation. This is a not a physical interpretation of the injection process. The source term should be included with the advection solve and implemented in the method described in the VCCMM where they trace the fluid forward out of the injection wells. The current implementation is subject to local oscillation around the wells since the QEL-WENO method can transport the fluid a larger spatial distance than where the injected fluid can reach with the diffusion solver.

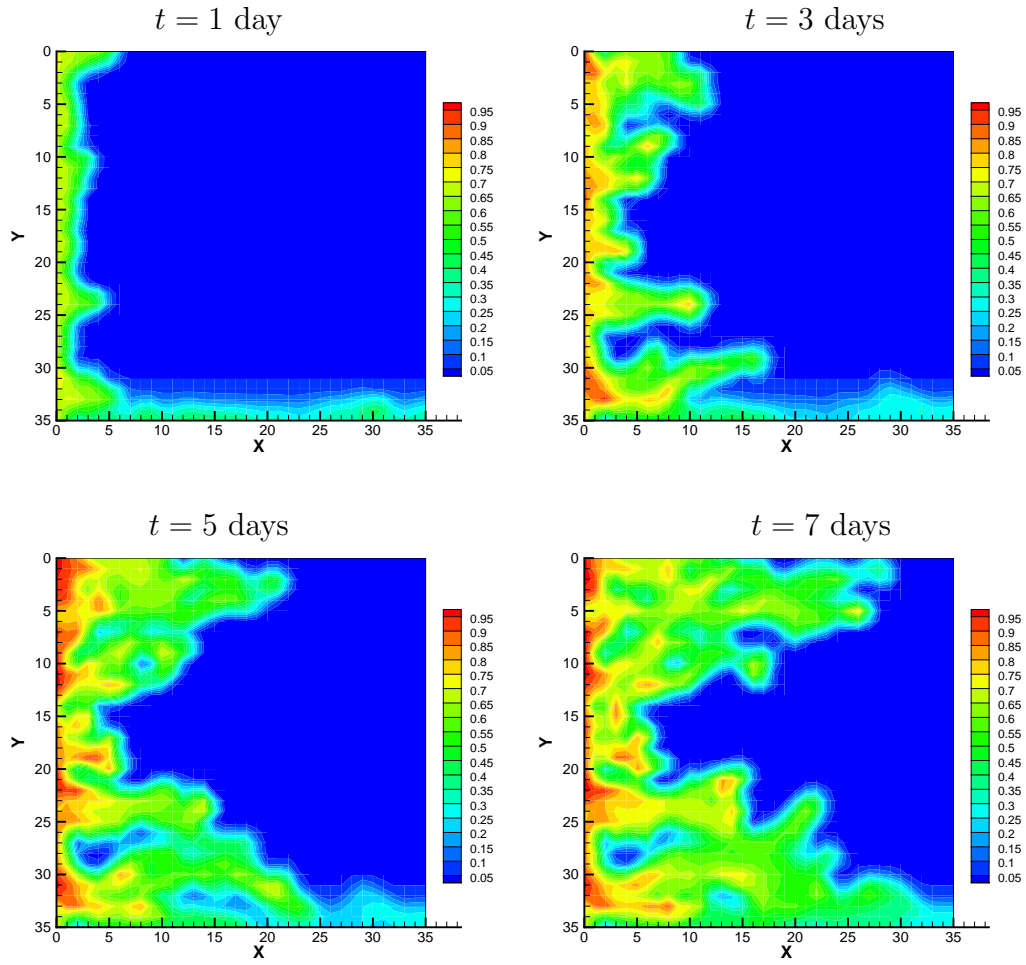


Figure 6.3: **Two-Phase Transport Solution** The solution to the two-phase flow problem using the Flux Corrected QEL-WENO Scheme.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Eulerian-Lagrangian algorithms are numerical methods solving advection equations. The goal in the development of these types of schemes is to utilize time steps significantly larger than the CFL limited time step of fixed mesh methods. The computational time of tracing the characteristics is mitigated since the problems naturally can be decomposed to solve different cell blocks on multiple processors. These schemes have recently been combined with the reconstruction techniques of WENO schemes which provide a high order accuracy.

In order to aid in the development of these schemes we have presented the new IWENO interpolation technique. This interpolation procedure allows for a high order reconstruction of the function at any point in the rectangular domain. The key idea in this method is to reconstruct the primitive function and the differentiate. These polynomials are weighted in the standard WENO way so that the interpolation avoids using a stencil that contains a shock in the data. These schemes achieve the theoretical order of accuracy.

This interpolation technique aided the development of two new strate-

gies for solving the advection problem. The first is to solve the linear advection problem by using a quadrature rule to integrate the traceback region. This allows for reconstruction of the mass in the traceback region without resolving its boundary. This method fails to be either mass or volume conservative, however criteria was developed to choose a sufficient number of quadrature points to ensure that the volume error was within an acceptable tolerance. Numerical results demonstrate the formally high order convergence rate for smooth problems.

The second application of IWENO is for the nonlinear problem which is solved using a flux correction technique. The IWENO technique is used to compute the integral of the flux across the approximated boundary. This technique was implemented in one spatial dimension. Numerical results were presented demonstrating its performance. The numerical results are very similar to the previous technique of Arbogast and Huang. The compact stencil improves the ability for the scheme to be run across multiple processors.

The final addition of this thesis is to combine the quadrature scheme with the flux-correction scheme in higher spatial dimensions. This is done through approximate characteristic tracing at element nodes to define a bilinear velocity field which is used in the quadrature scheme. Numerical results were presented.

7.2 Future Work

7.2.1 Conservation

The major limitation of the QEL-WENO method is that it fails to be conservative. There is an underlying algebraic system to the scheme $\bar{u}^{n+1} = M\bar{u}^n$, where M is a matrix with coefficients that depend on the scheme. It has been shown that the scheme is volume conservative if the column sum of $M = 1$, and mass conservative if the row sum of $M = 1$. Future research would include an investigation of methods to adjust the resulting algebraic system in a way to improve the mass conservation of the scheme.

7.2.2 CWENO3

Central WENO schemes are important fixed mesh methods since they allow for mass transport without an artificial flux splitting technique. However due to technical reasons these schemes did not exist for odd orders until recently due to challenges in reconstructing the midpoint. The IWENO technique presented here gives a way around this problem. In addition the idea of using a compact stencil to compute the flux across the edge has the potential of providing a less diffusive solution.

7.2.3 Applications

There are multiple applications in reservoir simulation such as unstable miscible displacement that the QEL-WENO algorithm can be applied to. Another direction for future research is testing the performance on systems of

coupled conservation laws such as the Euler Equations.

Bibliography

- [1] S. N. Antoncev and V. N. Monahov. Three-dimensional problems of time dependant two-phase filtration in nonhomogeneous anisotropic porous media. In *Soviet Math. Dokl, Vol. 19*, pages 1354–1358, 1978.
- [2] T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov. Mixed finite element methods on non-matching multiblock grids. *SIAM. J. Numer. Anal.*, 37:1295–1315, 2000.
- [3] T. Arbogast and Ch.-S. Huang. A fully mass and volume conserving implementation of a characteristic method for transport problems. *SIAM. J. Sci. Comput.*, 28(6):2001–2022, 2006.
- [4] T. Arbogast and Ch.-S. Huang. A fully conservative Eulerian-Lagrangian method for a convection-diffusion problem in a solenoidal field. *J. Comput. Phys.*, 229(9):3415–3427, 2010. DOI 10.1016/j.jcp.2010.01.009.
- [5] T. Arbogast, Ch.-S. Huang, and Ch.-H. Hung. A fully conservative Eulerian-Lagrangian stream-tube method for advection-diffusion problems. *SIAM. J. Sci. Comput.*, 34(4):B447–B478, 2012.
- [6] T. Arbogast, Ch.-S. Huang, and T. F. Russell. A locally conservative Eulerian-Lagrangian method for a model two-phase flow problem in a

- one-dimensional porous medium. *SIAM. J. Sci. Comput.*, 34(4):A1950–A1974, 2012.
- [7] T. Arbogast, M. Juntunen, J. Pool, and M. F. Wheeler. A discontinuous Galerkin method for two-phase flow in a porous medium enforcing $H(\text{div})$ velocity and continuous capillary pressure. *Comput. Geosci.*, 17(6):1055–1078, 2013.
- [8] T. Arbogast and Wen-Hao Wang. Stability, monotonicity, maximum and minimum principles, and implementation of the volume corrected characteristic method. *SIAM. J. Sci. Comput.*, 33(4):1549–1573, 2011.
- [9] T. Arbogast and M. F. Wheeler. A characteristics-mixed finite element method for advection dominated transport problems. *SIAM. J. Numer. Anal.*, 32:404–424, 1995.
- [10] T. Arbogast, M. F. Wheeler, and I. Yotov. Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences. *SIAM J. Numer. Anal.*, 34:828–852, 1997.
- [11] T. Arbogast and Hailong Xiao. Two-level mortar domain decomposition preconditioners for heterogeneous elliptic problems. *Comput. Methods Appl. Mech. Engrg.*, 292:221–242, 2015.
- [12] I. Babuška, G. Caloz, and J. E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31:945–981, 1994.

- [13] S. Balay, J. Brown, , K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.
- [14] D. S. Balsara and C.-W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comput. Phys.*, 160:405–452, 2000.
- [15] J. B. Bell, C. N. Dawson, and G. R. Shubin. An unsplit higher-order Godunov scheme for scalar conservation laws in two dimensions. *J. Comp. Physics*, 74:1–24, 1988.
- [16] Franco Brezzi, Jim Douglas Jr, and L Donatella Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2):217–235, 1985.
- [17] R. H. Brooks. *Hydraulic properties of porous media*. PhD thesis, Colorado State University, Fort Collins, Colorado, 1965.
- [18] R. H. Brooks and A. T. Corey. Hydraulic properties of porous media. *Hydrology Papers*, March 1964.
- [19] E. Carlini, R. Ferretti, and G. Russo. A weighted essentially nonoscillatory, large time-step scheme for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 27(3):1071–1091, 2005.

- [20] M. A. Celia, T. F. Russell, I. Herrera, and R. E. Ewing. An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Advances in Water Resources*, 13:187–206, 1990.
- [21] G. Chavent. A new formulation of diphasic incompressible flows in porous media. In A. Dold and B. Eckman, editors, *Applications of Methods of Functional Analysis to Problems in Mechanics*, number 503 in Lecture Notes in Math., pages 258–270. Springer Berlin, Heidelberg, 1976.
- [22] G. Chavent. A fully equivalent global pressure formulation for three-phases compressible flows. *Applicable Analysis*, 88(10–11):1527–1541, 2009.
- [23] G. Chavent and J. Jaffré. *Mathematical models and finite elements for reservoir simulation*. Elsevier Science Publishers, New York, 1986.
- [24] Z. Chen, G. Huan, and Y. Ma. *Computational Methods for Multiphase Flows in Porous Media*, volume 2 of *Computational Science and Engineering Series*. SIAM, Philadelphia, 2006.
- [25] Zhangxin Chen. Expanded mixed finite element methods for linear second order elliptic problems I. Technical Report 1219, Institute for Mathematics and its Applications, University of Minnesota, 1994.
- [26] H. K. Dahle, R. E. Ewing, and T. F. Russell. Eulerian-Lagrangian localized adjoint methods for a nonlinear advection-diffusion equation. *Comput. Methods Appl. Mech. Engrg.*, 122(3–4):223–250, 1995.

- [27] C. N. Dawson. Godunov-mixed methods for advection-diffusion equations in multidimensions. *SIAM J. Numer. Anal.*, 30:1315–1332, 1993.
- [28] C. N. Dawson, T. F. Russell, and M. F. Wheeler. Some improved error estimates for the modified method of characteristics. *SIAM J. Numer. Anal.*, 26:1487–1512, 1989.
- [29] J. Douglas, Jr. and C.-S. Huang. The convergence of a locally conservative Eulerian-Lagrangian finite difference method for a semilinear parabolic equation. *BIT*, 41:480–489, 2001.
- [30] J. Douglas, Jr., C.-S. Huang, and F. Pereira. The modified method of characteristics with adjusted advection. *Numer. Math.*, 83:353–369, 1999.
- [31] J. Douglas, Jr., F. Pereira, and L.-M. Yeh. A locally conservative Eulerian-Lagrangian numerical method and its application to nonlinear transport in porous media. *Comput. Geosci.*, 4:1–40, 2000.
- [32] J. Douglas, Jr. and T. F. Russell. Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures. *SIAM J. Numer. Anal.*, 19:871–885, 1982.
- [33] Y. Efendiev, J. Galvis, and Xiao-Hui Wu. Multiscale finite element methods for high-contrast problems using local spectral basis functions. *J. Comput. Phys.*, 230(4):937–955, 2011.

- [34] M. W. Farthing, C. E. Kees, T. F. Russell, and C. T. Miller. An ELLAM approximation for advective-dispersive transport with nonlinear sorption. *Advances in Water Resources*, 29:657–675, 2006.
- [35] Erwin Fehlberg. Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems. Technical Report 315, NASA, 1969.
- [36] P. Frolkovič. Flux-based methods of characteristics for coupled transport equations in porous media. *Comput. Visual. Sci.*, 6(4):173–184, 2004.
- [37] B. Ganis, K. Kumar, G. Pencheva, M. F. Wheeler, and I. Yotov. A global jacobian method for mortar discretizations of a fully implicit two phase flow model. in preparation, 2013.
- [38] B. Ganis, G. Pencheva, M.F. Wheeler, T. Wildey, and I. Yotov. A frozen jacobian multiscale mortar preconditioner for nonlinear interface operators. multiscale modeling and simulation. *Multiscale Model. Simul.*, 10(3):853–873, 2012.
- [39] Oscar Gonzalez and Andrew M Stuart. *A first course in continuum mechanics*. Cambridge University Press, 2008.
- [40] Sigal Gottlieb, Julia S. Mullen, and Steven J. Ruuth. A fifth order flux implicit WENO method. *J. Sci. Comput.*, 27(1–3):271–287, 2006.

- [41] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high-order accurate essentially nonoscillatory schemes III. *J. Comput. Phys.*, 71(2):231–303, 1987.
- [42] A. Harten and S. Osher. Uniformly high-order accurate nonoscillatory schemes I. *SIAM J. Numer. Anal.*, 24(2):279–309, 1987.
- [43] Y. Hasbani, E. Livne, and M. Bercovier. Finite elements and characteristics applied to advection-diffusion equations. *Computers and Fluids*, 11:71–83, 1983.
- [44] R. W. Healy and T. F. Russell. Treatment of internal sources in the finite-volume ELLAM. In Bentley et al., editors, *Computational Methods in Water Resources XIII, Vol. 2*, pages 619–622, Rotterdam, 2000. A. A. Balkema.
- [45] F. B. Hildebrand. *Introduction to numerical analysis*. Dover, New York, 2 edition, 1987.
- [46] Ch.-S. Huang and T. Arbogast. An Eulerian-Lagrangian WENO scheme for nonlinear conservation laws. *Submitted*, 2015.
- [47] Ch.-S. Huang, T. Arbogast, and Ch.-H. Hung. A re-averaged WENO reconstruction and a third order CWENO scheme for hyperbolic conservation laws. *J. Comput. Phys.*, 262:291–312, 2014.

- [48] Ch.-S. Huang, T. Arbogast, and Jianxian Qiu. An Eulerian-Lagrangian WENO finite volume scheme for advection problems. *J. Comput. Phys.*, 231(11):4028–4052, 2012. DOI 10.1016/j.jcp.2012.01.030.
- [49] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.
- [50] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, Basel, second edition, 1992.
- [51] D. Levy, G. Puppo, and G. Russo. Central WENO schemes for hyperbolic systems of conservation laws. *Math. Model. Numer. Anal.*, 33:547–571, 1999.
- [52] D. Levy, G. Puppo, and G. Russo. Compact central WENO schemes for multidimensional conservation laws. *SIAM J. Sci. Comput.*, 22(2):656–672, 2000.
- [53] D. Levy, G. Puppo, and G. Russo. A third order central WENO scheme for 2D conservation laws. *Appl. Numer. Math.*, 33:415–421, 2000.
- [54] X. D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200–212, 1994.
- [55] Y.-Y. Liu, C.-W. Shu, and M.-P. Zhang. On the positivity of linear weights in WENO approximations. *Acta Mathematicae Applicatae Sinica*, 25:503–538, 2009.

- [56] T. Neubauer and P. Bastian. On a monotonicity preserving Eulerian-Lagrangian localized adjoint method for advection-diffusion equations. *Adv. Wat. Res.*, 28(12):1292–1309, 2005.
- [57] J. C. Parker. Multiphase flow and transport in porous media. *Reviews of Geophysics*, 27:311–328, 1989.
- [58] M. Peszyńska, Q. Lu, and M. F. Wheeler. Coupling different numerical algorithms for two phase fluid flow. In *The Mathematics of Finite Elements and Applications X*, pages 205–214. Brunel University, 1999. MAFELAP.
- [59] M. Peszyńska, M. F. Wheeler, and I. Yotov. Mortar upscaling for multiphase flow in porous media. *Comput. Geosci.*, 6:73–100, 2002.
- [60] J. Qiu and C.-W. Shu. On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. *J. Comput. Phys.*, 183:187–209, 2002.
- [61] J.-M. Qiu and A. Christlieb. A conservative high order semi-Lagrangian WENO method for the Vlasov equation. *J. Comput. Phys.*, 229:1130–1149, 2010.
- [62] J.-M. Qiu and C.-W. Shu. Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow. *J. Comput. Phys.*, 230(4):863–889, 2011.

- [63] J.-M. Qiu and C.-W. Shu. Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation. *Communications in Comput. Phys.*, 10:979–1000, 2011.
- [64] R. A. Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. In I. Galligani and E. Magenes, editors, *Mathematical Aspects of Finite Element Methods*, number 606 in Lecture Notes in Math., pages 292–315. Springer-Verlag, New York, 1977.
- [65] M. Restelli, L. Bonaventura, and R. Sacco. A semi-lagrangian discontinuous Galerkin method for scalar advection by incompressible flows. *J. Comput. Phys.*, 216(1):195–215, 2006.
- [66] M. E. Rose. Numerical methods for flows through porous media II. *Computers and Math. Appl.*, 20:99–122, 1980.
- [67] M. E. Rose. Numerical methods for flows through porous media I. *Math. Comp.*, 40:435–467, 1983.
- [68] T. F. Russell and M. F. Wheeler. Finite element and finite difference methods for continuous flows in porous media. In R. E. Ewing, editor, *The Mathematics of Reservoir Simulation*, number 1 in Frontiers in Applied Mathematics, pages 35–106, Chapter II. Society for Industrial and Applied Mathematics, Philadelphia, 1983.
- [69] A. E. Scheidegger. *The Physics of Flow Through Porous Media*, 3rd ed. University of Toronto Press, Toronto, 1974.

- [70] J. Shi, C. Hu, and C.-W. Shu. A technique of treating negative weights in WENO schemes. *J. Comput. Phys.*, 175(1):108–127, 2002.
- [71] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In B. Cockburn and A. Quarteroni, editors, *Advanced numerical approximation of nonlinear hyperbolic equations*, volume 1697 of *Lecture Notes in Mathematics*, chapter 4, pages 325–432. Springer-Verlag, Berlin, 1998.
- [72] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II. *J. Comput. Phys.*, 83:32–78, 1989.
- [73] John M. Stockie, John A. Mackenzie, and Robert D. Russell. A moving mesh method for one-dimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 22(5):1791–1813, 2001.
- [74] M. R. Todd, P. M. O’Dell, and G. J. Hirasaki. Methods for increased accuracy in numerical reservoir simulators. *Soc. Petrol. Engr. J.*, 12:515–530, 1972.
- [75] H. Wang, D. Liang, R. E. Ewing, S. L. Lyons, and G. Qin. An ELLAM approximation for highly compressible multicomponent flows in porous media. *Comput. Geosci.*, 6(3–4):227–251, 2002.
- [76] Hong Wang. An optimal-order error estimate for MMOC and MMO-CAA schemes for multidimensional advection-reaction equations. *Numer. Methods Partial Differential Equations*, 18(1):69–84, 2002.

- [77] Hong Wang and M. Al-Lawatia. A locally conservative Eulerian-Lagrangian control-volume method for transient advection-diffusion equations. *Numer. Methods Partial Differential Equations*, 22(3):577–599, 2006.
- [78] Hong Wang, Weidong Zhao, Magne S. Espedal, and Aleksey S. Telyakovskiy. A component-based Eulerian–Lagrangian formulation for multicomponent multiphase compositional flow and transport in porous media. *SIAM J. Sci. Comput.*, 35(2):B462–B486, 2013.
- [79] Hong Wang, Weidong Zhao, and R. E. Ewing. A numerical modeling of multicomponent compressible flows in porous media with multiple wells by an Eulerian-Lagrangian method. *Comput. Vis. Sci.*, 8(2):69–81, 2005.
- [80] Stephen Whitaker. Flow in porous media I: A theoretical derivation of Darcy’s law. *Transport in Porous Media*, 1:3–25, 1986.
- [81] Stephen Whitaker. Flow in porous media II: The governing equations for immiscible, two–phase flow. *Transport in Porous Media*, 1:105–125, 1986.
- [82] Hailong Xiao. *Multiscale Mortar Mixed Finite Element Methods for Flow Problems in Highly Heterogeneous Porous Media*. PhD thesis, University of Texas at Austin, 2013.
- [83] Ivan Yotov. Interface solvers and preconditioners of domain decomposition type for multiphase flow in multiblock porous media. *Advances in*

computation: Theory and practice, 7:157–167, 2001.

[84] Ivan Yotov. A multilevel newton–krylov interface solver for multiphysics couplings of flow in porous media. *Numerical linear algebra with applications*, 8(8):551–570, 2001.

[85] M. Zennaro. Natural continuous extensions of Runge-Kutta methods. *Math. Comp.*, 46:119–133, 1986.

Vita

Jamie Pool was born the daughter of Michael William Pool, M.D., and Jennifer Elizabeth Quick, M.A., in Oakland, California on January 21, 1986. She completed high school at Morgan Park Academy in Chicago, Illinois in May, 2004. She received the Bachelor of Science degree in Engineering Mechanics from the University of Illinois at Urbana-Champaign in May, 2008. She was accepted in the Mathematics department at the University of Texas at Austin and started graduate studies in August, 2008. She began her research on transport methods in January 2011.

Permanent address: 701 Keasbey Street
Austin, Texas 78751

This dissertation was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.