THESIS

ENTROPY STABILITY FOR A FOURTH-ORDER ACCURATE FINITE-VOLUME METHOD FOR BURGERS' EQUATION

Submitted by

Noah Meisner

Department of Mechanical Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2019

Master's Committee:

Advisor: Xinfeng Gao

Stephen Guzik
Jiangguo Liu

ABSTRACT

ENTROPY STABILITY FOR A FOURTH-ORDER ACCURATE FINITE-VOLUME METHOD FOR BURGERS' EQUATION

Computational fluid dynamics (CFD) algorithms need efficiency, accuracy, and robustness to be useful to engineers. Faster computers improve the effective speed of a given method, and larger memories allow higher grid resolution, improving accuracy. However, robustness cannot be achieved through advancements in computer hardware. Improvements in this area require a fundamental understanding of the mathematical and physical aspects of the algorithm being investigated. For high-order numerical algorithms, the stability can easily be aggravated with the presence of strong gradients. Many methods in CFD incorporate some kind of numerical limiter to suppress spurious oscillations and handle nonlinear instabilities for flows with strong discontinuities. However, these limiters often lack a basis in the physics that governs the fluid flow. For this reason, the present research employs a limiting method that is based on the second law of thermodynamics to achieve numerical robustness for a higher order code in solving flows with strong discontinuities.

The aim of this work is to address the question of robustness for a high-order finite-volume method (FVM) by extending the entropy stability strategy developed by Marshal L. Merriam [1] for a second-order FVM. Unlike generic limiters or artificial viscosity, the approach explored in this thesis provides a physical, quantitative explanation for artificial viscosity or limiters in the form of entropy. The mathematical derivation of the entropy stability method is presented in detail, shortcomings of the method by Merriam are explored,

and a more robust approach to deriving an entropy stable limiting method was carried out for the low-order methods. As a first step, this study focuses on the application to Burgers' equation for both a first- and second-order accurate solution to a problem with the onset of shocks. Then, a cell entropy fix for the fourth-order discretization scheme is derived and applied to Burgers' equations. Although the oscillations near the discontinuities can be mitigated, the logical conditions associated with ensuring the entropy constraints become impractical to implement for high-order discretization schemes. Through this research, it is deemed that the entropy stability method proposed by Merriam may not be a viable solution to effectively suppress oscillations near strong discontinuities of problems governed by systems of nonlinear equations, particularly, for high-order schemes.

ACKNOWLEDGEMENTS

This research would not have been possible without all of the people who surround me. First and foremost I must thank Dr. Xinfeng Gao for providing insight into this research, and encouraging me to pursue an advanced degree in Computational Fluid Dynamics. The assistance in class, in meetings, and advice about my professional career is the driving force behind the quality of this work. I would also like to thank my committee; Dr. Stephen Guzik and Dr. Jiangguo Liu for their assistance throughout my graduate studies while at Colorado State University.

I would also like to thank my colleagues in the CFD lab for all of their technical assistance with this research. I truly could not have done it without the advice and support from them. I must also thank my fiancé, Torrey Fernandez, for all of the support in my move accross the country, dealing with my long hours, and always being a voice of encouragement and support. Finally, I would like to thank my parents, and older brother for their unwavering support of my passions and assistance in getting me to where I am today.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

Many numerical Methods in Computation Fluid Dynamics (CFD) incorporate an artificial dissipation scheme and/or some type of numerical limiter to suppress spurious oscillations to cope with the nonlinear instabilities for flows with strong discontinuities. For higher-order numerical algorithms, the stability can be easily aggravated with the presence of strong gradients. These sharp gradients have the potential to contaminate the solution so much that the results could become unphysical. To get rid of these unphysical oscillations researches attacked them head on, producing schemes which, at least for scalar equations in one dimension were guaranteed not to have any "wiggles." The concept of Total Variation Diminishing (TVD) schemes was formalized by Harten [2] and implemented for gasdynamics, which was effective in getting rid of these annoying oscillations. These limiting schemes came in two different forms. Flux limiters, which acted on system fluxes, and slope limiters, which acted of a system of states (like pressure, velocity, etc.). These limiters were effective in reducing the magnitude of the spurious oscillations in the discontinuous regions of a flowfield, resulting in solutions that were smooth, and stable. These schemes do not necessarily follow a physical law, and often times were developed from a strictly numerical point of view.

However, Lax referenced proofs [3, 4] that although the physically accurate solution has the TVD property, it does not follow that all TVD schemes converge to a physical solution. Schemes have been introduced which satisfy the TVD condition and yet produce unphysical solutions. To make schemes stable and rid of oscillations, some researchers have ignored the entropy inequality [5] that is fundamental to getting a correct solution. Motivated by the point of view from the second law of thermodynamics, the present work examines the entropy production in a finite-volume scheme and aims to develop stability requirements in a high-order scheme for nonsmooth flows using Burgers' equation as a test case.

The uniqueness of the second law of thermodynamics provides information about whether a process is feasible. To be more specific, the production of entropy can be evaluated for a computational cell or the entire domain in a numerical method. If the entropy production is negative, then potential oscillations may occur in the cell or throughout the domain. Accordingly, certain stabilization methods need to be applied to smooth or suppress the oscillations while strictly adhering to the second law of thermodynamics.

Numerical methods require speed, accuracy, and robustness to be useful. Faster computers improve the effective speed of a given method, and larger and more efficiently organized memories allow for more grid points, thereby improving accuracy. Robustness, on the other hand, cannot be achieved through improvements in computer hardware. The aim of this thesis is to address the robustness of the solution based on the use of the second law of thermodynamics and the ensurance of positive entropy production. Framing a solution methodology with respect to the second law appears to be a robust approach guaranteed to produce physically accurate results.

Examining the entropy stability of a fourth-order accurate finite-volume method is the interest of this work. Finite-volume methods are well suited for problems with discontinuities. High-order finite-volume methods can produce solutions to smooth flows much faster than low-order schemes, to the same level of accuracy. Unfortunately, instability issues often occur when solving nonsmooth flows because of the fourth-order stencil operations involved in these schemes. For example, in our fourth-order scheme, the primitive variables and their gradients on the cell-face are approximated using a fourth-order center-differencing method.

These face values need to be limited or reconstructed in nonsmooth flows in order to get rid of potential oscillations.

Work has been done on finite-volume schemes and finite-difference schemes which satisfy an entropy inequality [1, 5, 6]. It concerns a numerical approximation to the entropy inequality on a discrete basis. In these schemes, the local oscillations which still occur offer evidence of local violations of the second law, even though it is globally satisfied. The goal of the present work is to satisfy the second law of thermodynamics on a cell-by-cell basis. Efforts are made to verify the solution procedure as outlined by Merriam [1], and examine its effectiveness by applying it to solve the Burgers' equation.

1.1. Literature Review

First, the existing literature is reviewed to cover the original development and groundwork of entropy stability in section 1.1.1. The reference work is some of the earliest in the pursuit of an algorithm which adheres to the second law of thermodynamics, while also maintaining accuracy and consistency. In section 1.1.2 some recent work on entropy stable methods is outlined. Finally, section 1.2 emphasizes the motivation of the development and application of entropy stable methods in the present thesis.

1.1.1. Marshal L. Merriam 1989.

Entropy stability is a constraint placed on (in most cases) a system of nonlinear equations whose numerical solution obtained by a numerical method respects the second law of thermodynamics either globally, or locally. These schemes were originally developed alongside the advent of modern CFD algorithms for solving discontinuous flows. Since the problem first arose of nonlinear stability researches have been attempting to develop algorithms that are entropy stable and accurate. This has proved to be a difficult task, and one that is not universally agreed upon [5–7]. Marshal L. Merriam proposed a method for obtaining entropy stability in 1989 which consisted of utilizing entropy as a non-traditional limiter. This method replaces a numerical limiter, or artificial dissipation by satisfying a cell entropy inequality for a FVM on a cell-by-cell basis. It does so by applying linear adjustments in a limiting agent $\boldsymbol{\phi}$, which is a variable or vector depending on the governing equations. This limiting agent is applied at the each cell face, and using an iterative process to find entropy stable values of the conservative variables at each cell face. The limited face state dependent on $\boldsymbol{\phi}_{j+\frac{1}{2}}$ is given as

$$\boldsymbol{q}_{j+\frac{1}{2}} = \boldsymbol{q}_j + \frac{1}{2}\boldsymbol{\phi}_{j+\frac{1}{2}}(\boldsymbol{q}_{j+1} - \boldsymbol{q}_j), \tag{1}$$

where $\boldsymbol{q}$ is any solution variable or vector depending on the governing equations, and j, $j+1$ and $j+\frac{1}{2}$ denote the cell index, adjacent cell index, and cell interface, respectively in one-dimension.

Other entropy stable schemes range from changing the conservation equations to include an entropy component while still making use of a traditional numerical limiter [8], to satisfying an entropy condition for the conservation laws based on a generalized summation-by-parts property [5]. Entropy stability has also been the subject of some criticism. Merriam himself criticizes the TVD connection to this method saying that overall, “it appears that satisfaction of a cell entropy inequality is sufficient to produce a stable scheme... On the bad side, it evidently isn’t sufficient to avoid the unphysical oscillations in momentum and it has a severe time step limitation.” [1]

The satisfaction of a cell entropy inequality is the crux of the research done by Merriam. Providing a numerical scheme in which the entropic process is forced to run in the correct direction for each discrete cell guarantees that the entropy throughout the domain will be conserved. As a first step Merriam applies his proposed entropy stable scheme to Burgers' equation. Burgers' equation is used frequently in research such as this because a shock can be induced without the necessity for a system of equations. It can be illuminating to research the behavior of a limiter on a single variable, which is exactly what this thesis is focused on.

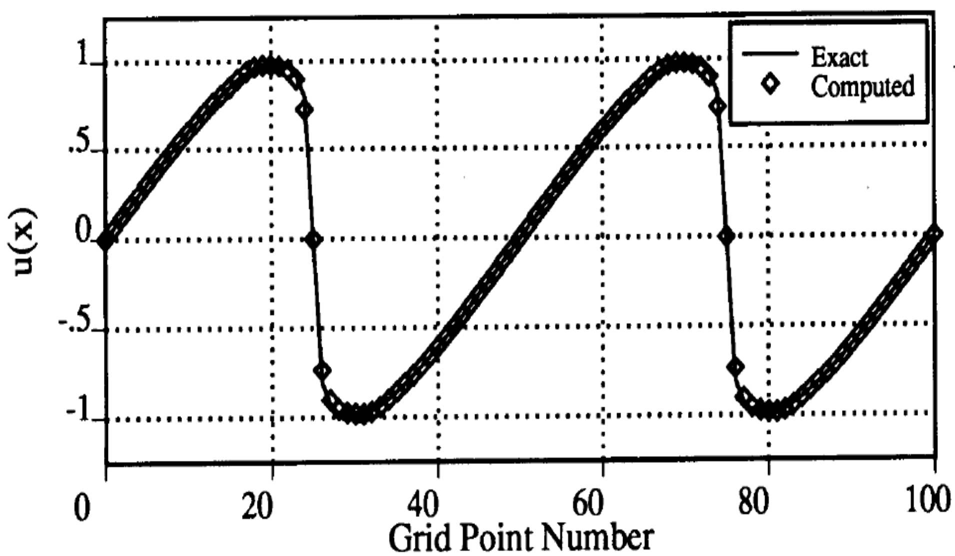

FIGURE 1.1. Onset of the shock for Burgers' equation with an initial condition of a sin wave [1]

The derivation and criteria placed on this limiting agent will be explored in further detail in Chapter 2. For illustration, this is briefly discussed here to show that the solution to this limiting agent that provides positive entropy generation is clearly much more straightforward for Burgers' equation than for a system of equations. It is convenient to test this methodology on Burgers' equation, because when the flow is initialized to a sin wave, shocks will be produced. The solution provided by Merriam with this limiting method applied in a first-order accurate spatial discretization scheme is shown in Fig. 1.1.

Finding values for the limiting agent, ϕ, which is a coefficient that determines the entropy stable scheme involves finding the stability region with respect to entropy production for

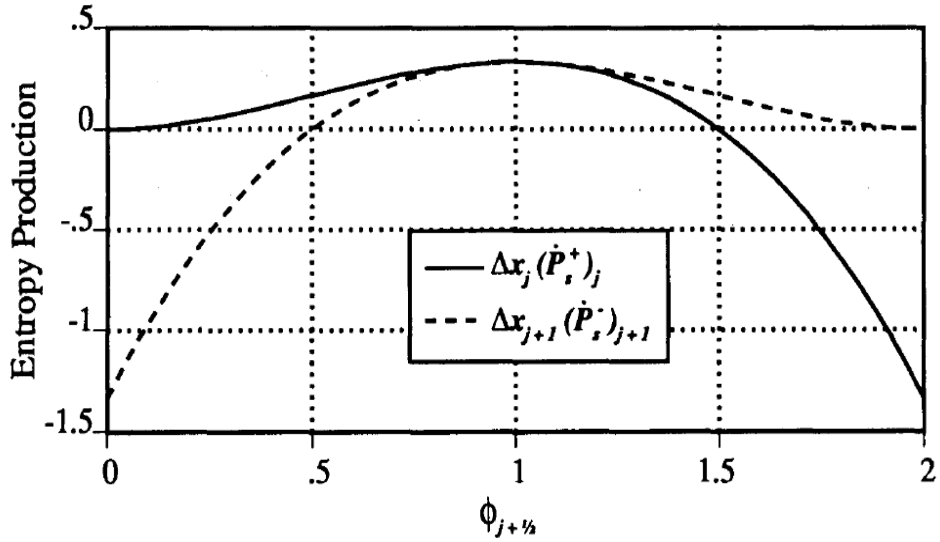

FIGURE 1.2. Entropy stability region for a sonic expansion for Burgers' equation [1]

each specific flow. The values of $\boldsymbol{\phi}$ are constrained to lie between zero and two. For the face interpolation scheme given by Eq. 1, these values correspond to the scheme changing between first and second-order accurate. When $\boldsymbol{\phi}_{j+\frac{1}{2}} = 0$ then $\boldsymbol{q}_{j+\frac{1}{2}} = \boldsymbol{q}_j$, recovering a first-order approximation. Alternatively, when $\boldsymbol{\phi}_{j+\frac{1}{2}} = 2$ then $\boldsymbol{q}_{j+\frac{1}{2}} = \boldsymbol{q}_{j+1}$. When $\boldsymbol{\phi}_{j+\frac{1}{2}} = 1$ then $\boldsymbol{q}_{j+\frac{1}{2}} = \frac{\boldsymbol{q}_j + \boldsymbol{q}_{j+1}}{2}$, recovering a second-order average. Details of this can be found in Chapter 6 of the reference work [1]. For example, Fig. 1.2 shows the stability region (for Burgers' equation, intersection point) for the value of $\boldsymbol{\phi} = 1$ that satisfies the entropy constraints (intersection between two lines) and produces positive entropy at a sonic point.

1.1.2. FISHER AND CARPENTER 2013. Numerical solutions to problems with shocks are considerably more difficult to obtain than smooth flows because of the strong gradients. Solution methods for these problems are typically hybrid [9, 10] or high-order adaptive [11, 12] schemes, or highly dissipative low-order methods. Many methods have been devised that attempt to balance accuracy, added dissipation, and efficiency. Work done by Fisher and Carpenter [5] seeks a numerical method that is based on nonlinear analysis that is entropy stable. Satisfying an entropy inequality is an uncommon property for conservative, high-order methods. Recent advances in entropy stability theory for the compressible Euler equations now facilitate the development of conservative and entropy conservative high-order formulations. Entropy conservative schemes are constructed by Tadmor and others [13–15] for second-order finite-volume methods. An extension to high-order formulations on periodic domains is given by LeFloch and Rhode [16]. These schemes are made computationally tractable for the Navier-Stokes equations through the work of Ismail and Roe [8]. A methodology for constructing entropy stable schemes satisfying a cell entropy inequality and capable of simulating flows with shocks in periodic domains is developed by Fjordholm et al. [17]. However, in the work developed by Fisher and Carpenter, a generalized approach to entropy stability is developed based on operators that naturally extend to high-order methods. First, the general considerations sufficient for achieving entropy conservation are developed based on a generalized summation-by-parts property. The entropy condition for the conservation law is found by integrating the second law of thermodynamics over a finite cell given by

$$\frac{d}{dt}\int_{x_L}^{x_R} S dx + F \geq 0, \tag{2}$$

where x_L and x_R are the left and right of the integration domain respectively, S is the specific entropy, and F is the entropy flux. The existence of an entropy function is in general not guaranteed for an arbitrary nonlinear system of equations due to the large number of constraints it must satisfy. This is potentially a problem in developing a general entropy stability criteria, and is one of the reasons why so many independent approaches to entropy stability can be found in literature. Next, the entropy conservative schemes for conservation laws developed by Tadmor and others [13] are extended to high-order on finite domains including boundary closures. Finally, high-order narrow-stencil discrete operators are derived for the viscous terms and are implemented in an entropy stable manner.

1.2. Motivation

State-of-the-art limiting methods provide stable numerical solutions at sufficient accuracy. Current limiting methods such as artificial viscosity or artificial dissipation, while robust, lack basis in the fundamental physics that govern fluid flow. These are both numerical schemes that mimic the behavior of observed fluid flow properties. Additionally, these limiting methods often do not take into account the second law of thermodynamics, and can unintentionally violate entropy conservation either locally, and/or globally. A limiting method based on entropy gives engineers working on complex problems the ability to determine areas of design where improvement is imperative, as well as confidence that their solutions are accurate and robust. The entropy limiting methods investigated in this thesis respect the second law of thermodynamics, and are designed to achieve stable and accurate numerical solutions when strong discontinuities occur in a flow problem. These approaches ensure that entropy is conserved locally, on a cell-by-cell basis, which in turn ensures that positive entropy generation occurs locally, and globally. Some entropy stable methods only apply entropy nonnegativity to the entire domain, which leaves the possibility that the second law can be violated locally in the domain which is unphysical for a closed system. The methods outlined in this thesis will provide some insight into the ability to use entropy stable methods through its effective enforcement of entropy generation. The entropy stable methods are exercised using a fourth-order finite-volume method for solving the one dimensional Burgers' equation.

1.3. Thesis Organization

The thesis is organized as follows. In Chapter 2, an overview of the entropy stability method is provided as outlined by Merriam. Application to Burgers' equation is explored, and the results in the reference work are reproduced, and compared with a set of logical conditions that are derived by satisfying the cell entropy inequality requirements in a more robust approach. Both of these methods are applied to Burgers' equation for a first-, and second-order accurate spatial discretization scheme. Chapter 3 outlines the extension of this "new" limiting method to a fourth-order accurate spatial discretization. This "new" method essentially follows the same entropy requirements proposed by Merriam, however instead of making assumptions about the flow, each criteria is solved mathematically resulting in specific entropy criteria that must be adhered to. Details will be provided in Chapter 2, and Chapter 3. Chapter 4 provides results for this fourth-order accurate method as well as a hybrid method that is entropy stable. Finally, Chapter 5 draws conclusions and suggests furture work. The appendix includes the Maxima [18] code used for derivation of the entropy criteria.

CHAPTER 2

Limitations in the Existing Entropy Stability Methods

In this chapter, the fundamental mathematical formulation of entropy stability is presented. Section 2.1 presents the underlying physics associated with the criteria that must be met for a scheme to be entropy stable. Next, section 2.2 details the existing first-order entropy stability method by Merriam, and the new method derived by strictly following the entropy stability criteria. Both the existing and the new methods are applied to Burgers' equation. Section 2.4 outlines the extension of both methods for a second-order scheme. Finally, the limitations of both methods are summarized.

2.1. Second Law of Thermodynamics

The second law of thermodynamics describes direction of possible state changes, and guards against the impossible. Therefore, the second law of thermodynamics is one of the most robust ways to approach a stabilization method by enforcing the production of entropy when solving flows with discontinuities. Deriving a semi-discrete version of the second law allows us to develop criteria that must be enforced in order to be both entropy stable and accurate. Finite-volume methods are often used to solve discontinuous flows on discrete domains. The basic concept of these is to satisfy the integral form the the conservation law to some degree of approximation for each of many contiguous control volumes which cover the domain of interest. The integral equation for smooth flows describing a control volume that can be identified by a single index, j is given below in Eq. 3. Additionally, the surface bounding the j^{th} volume is described by ∂V_j, and is shown in Fig. 2.1 using one dimension as an example.

$$\int_{V_j} \boldsymbol{q}[t+\Delta t]dv - \int_{V_j} \boldsymbol{q}[t]dv + \int_t^{t+\Delta t} \oint_{\partial V_j} \boldsymbol{f} \cdot \boldsymbol{n} dA d\tau = 0 \tag{3}$$

In this equation, $\boldsymbol{q}$ is a solution variable, or vector, $\boldsymbol{f}$ is a flux variable, or vector, depending on the governing equation(s), and $\boldsymbol{n}$ is the outward facing unit normal.

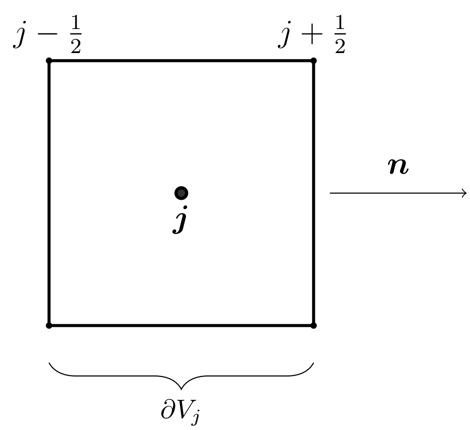

FIGURE 2.1. Control volume of a discrete cell in one dimension

In constructing the cell interface values as such, we can use the following definitions

$$\boldsymbol{q}_{j\pm\frac{1}{2}} = \boldsymbol{q}(x_{j\pm\frac{1}{2}}), \tag{4}$$

$$\boldsymbol{f}_{j\pm\frac{1}{2}} = \boldsymbol{f}(\boldsymbol{q}_{j\pm\frac{1}{2}}). \tag{5}$$

Considering integration of the governing equations over a time sufficiently small that $\boldsymbol{q}_j$ is essentially constant, it is important to be sure that reconstruction of $\boldsymbol{q}(x)$ from $\boldsymbol{q}(j)$ is done in such a way that entropy is not destroyed. From this point of view, the safest assumption is the piecewise constant assumption given by

$$\boldsymbol{q}(x) = \bar{\boldsymbol{q}}_j \quad x_{j-\frac{1}{2}} \leq x \leq x_{j+\frac{1}{2}}. \tag{6}$$

The cell average solution values are defined as

$$\langle \boldsymbol{q}_j \rangle \equiv \frac{1}{V_j} \int_{V_j} \boldsymbol{q}_j dV_j. \tag{7}$$

A finite-volume method requires the construction of a cell-average value, as well as the values of the flux, which further need solution variables or vector at the cell interface. Throughout the rest of this work, the brackets from the cell averaged value will be dropped, and the solution variables at each cell will represent cell-average values with respect to the definition given above in Eq. 7. With these definitions for the cell averaged values, the integral form becomes

$$\frac{d}{dt}\boldsymbol{q}_j + \frac{\boldsymbol{f}_{j+\frac{1}{2}} - \boldsymbol{f}_{j-\frac{1}{2}}}{\Delta x} = 0, \tag{8}$$

which is the semi-discrete version of the conservation law defined in Eq. 3. This equation is exact, as written.

We can use a similar approach to obtain the semi-discrete form of the second law of thermodynamics. The second law in integral form can be written as

$$\int_t^{t+\Delta t} \int_V \dot{P}_s dv d\tau \equiv \int_V S[t+\Delta t] dv - \int_{V_j} S[t] dv + \int_t^{t+\Delta t} \oint_{\partial V_j} F \cdot \boldsymbol{n} dA d\tau \geq 0. \tag{9}$$

Eq. 9 must be greater than or equal to zero for all processes according the the physics. Processes for which $\dot{P}_s$ are less than zero are never observed, and violate the second law. Adhering to this inequality is the foundation for the existing limiting method proposed by Merriam, and the new method outlined in this chapter. In the equation above, $\dot{P}_s$ defines the entropy production per unit volume, S is the specific entropy per unit volume, defined as

$S = \rho s$, F is the entropy flux per unit area, defined as $F = \rho u s$, and s is the nondimensional specific entropy. The second law of thermodynamics is not constrained for one equation, a system of equations, or an ideal gas. It is applied consistently to all continuum mechanics. Using the same definitions above for the cell interface values and the flux values given by Eq. 4, the semi-discrete version of the second law can be written as

$$(P_s)_j \equiv \frac{d}{dt} S_j + \frac{F_{j+\frac{1}{2}} - F_{j-\frac{1}{2}}}{\Delta x} \geq 0. \tag{10}$$

Using the piecewise constant approximation of Godunov, this provides the maximum entropy in each cell, consistent with the known values of $\boldsymbol{q}_j$. Since $\boldsymbol{q}(x)$ is continuous, any analysis valid for the integral equations is also valid for the semi-discrete equations. The definitions of both S and F are not always unique, and can change depending on the application. For example, Harten and Lax [19] derived a unique entropy pair for the Euler equations. These must be chosen in such a way as to satisfy the mathematical theory of entropy, given as:

$$\frac{\partial^2 S}{\partial \boldsymbol{q}^2} < 0, \tag{11}$$

$$\frac{\partial F}{\partial \boldsymbol{q}} = \frac{\partial S}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}. \tag{12}$$

Eq. 11, also known as the convexity condition of entropy, forces irreversible processes to run the correct direction and produce entropy. The derivation of the semi-discrete form of entropy production was necessary in order to construct a scheme that is entropy stable and accurate. Due to the piecewise constant assumption, the domain integrals can easily be carried out as cell-by-cell sums in the form of

$$\sum_{j}^{J}\left(\frac{\partial S}{\partial t}\right)+F_{j+\frac{1}{2}}-F_{j-\frac{1}{2}}\geq 0. \tag{13}$$

Eq. 13 directly implies that in each cell, the entropy generation must be greater than or equal to zero. This equation can be satisfied with the cellwise condition

$$(\dot{P}_s)_j \geq 0, \tag{14}$$

where $(\dot{P}_s)_j$ is explicitly defined as

$$(\dot{P}_s)_j = -(\frac{\partial S}{\partial \boldsymbol{q}})_j(\boldsymbol{f}_{j+\frac{1}{2}}-\boldsymbol{f}_{j-\frac{1}{2}})+(F_{j+\frac{1}{2}}-F_{j-\frac{1}{2}}). \tag{15}$$

For convenience, throughout the remainder of this work the dot is removed from the entropy generation term, from $(\dot{P}_s)_j$ to $(P_s)_j$. The definition remains the same. This conclusion is the crux of the framework of the entropy stability method. Satisfying this constraint guarantees entropy stability for a given flow. Next, this general criteria followed by Merriam and this work will be demonstrated on Burgers' equation. The solutions from the existing method and the new method are compared and analyzed. Although both methods follow the entropy stability criteria, their effects are different near the discontinuities.

2.2. Application to Burgers Equation

Often times simple examples can be more illuminating than the more complicated ones. For this reason, the application of the entropy stability method will be proposed here with respect to Burgers' equation rather than for the Euler system of equations to demonstrate the methodology. Burgers' equation is given in the form

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \tag{16}$$

where f is any nonlinear flux function of u given by

$$f = \frac{1}{2}u^2. \tag{17}$$

Burgers' equation is an ideal governing equation to apply the entropy stability method, because a suitable entropy function is explicitly expressed as

$$S = -u^2, \tag{18}$$

$$F = -\frac{2}{3}u^3, \tag{19}$$

which satisfies the entropy convexity condition given in the previous section by Eq. 11. This entropy pair also satisfies the compatibility condition, $\frac{\partial F}{\partial u} = \frac{\partial S}{\partial u}\frac{\partial f}{\partial u}$. An expression for the constraint (P_s) that must be satisfied was derived in the previous section, but is explored here in the context of a first-order accurate method. In order to guarantee an entropy stable first-order method as proposed by Merriam, the entropy generation constraints given by

$$(P_S)_j = (P_S^+)_j + (P_S^-)_j \geq 0, \tag{20}$$

must be met on a cell-by-cell basis. These terms are depicted in Fig. 2.2 as the right and left half cell entropy production values in each discrete cell.

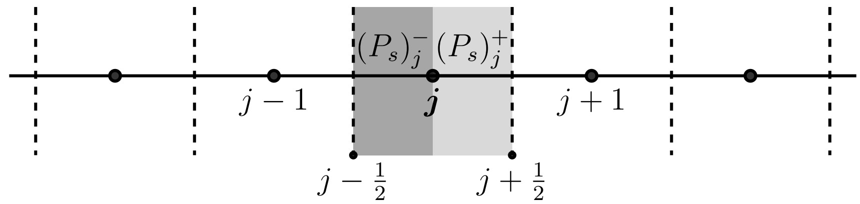

FIGURE 2.2. Right and left half cell entropy generation

A first-order spatial discretization scheme requires that each term, $(P_S^+)_j$ and $(P_S^-)_j$, be individually positive. These are defined as

$$\Delta x_j (P_s^+)_j = -(\frac{\partial S}{\partial u})_j (f_{j+\frac{1}{2}} - f_j) + (F_{j+\frac{1}{2}} - F_j), \tag{21}$$

$$\Delta x_j (P_s^-)_j = -(\frac{\partial S}{\partial u})_j (f_j - f_{j-\frac{1}{2}}) + (F_j - F_{j-\frac{1}{2}}). \tag{22}$$

If a process is reversible, then in accordance with the second law the entropy generated in a closed system is equal to zero. However, practical processes are irreversible, therefore the entropy generation must be positive. Satisfaction of the cell entropy inequality requires the selection of fluxes that depend on one limiting constant, ϕ at each cell interface. In principal, the limiting agent, $\phi_{j+\frac{1}{2}}$, can always be chosen in such a way as to make $(P_S^+)_j$ and $(P_S^-)_j$ nonnegative.

Choosing values for $u_{j+\frac{1}{2}}$ in such a way as to make $(P_s^+)_j$ and $(P_s^-)_j$ individually positive requires that

$$(P_s^+)_j \geq 0 \quad \& \quad (P_s^-)_{j+1} \geq 0. \tag{23}$$

This is because each face value determines the entropy generation on the left and right side of the face. The choice for $u_{j+\frac{1}{2}}$ that satisfies these constraints that is dependent on the limiting agent can be written as

$$u_{j+\frac{1}{2}} = u_j + \frac{1}{2}\phi_{j+\frac{1}{2}}(u_{j+1} - u_j), \tag{24}$$

where the value for $\phi_{j+\frac{1}{2}}$ takes on values between zero and two as $u_{j+\frac{1}{2}}$ takes on values between u_j and u_{j+1}. It is worth mentioning here that Godunov's Order Barrier Theorem [20] necessarily reverts this scheme to first-order accurate in the neighborhood of shocks. It is not the interest of this work to include a detailed derivation of the theorem provided by Godunov, but it is noteworthy in that the limiting schemes presented both by Merriam, and in this research, obey the property that limiting schemes for solving partial differential equations (PDE's) having the property of not generating new extrema can at most be first-order accurate. The face interpolation scheme given here by Eq. 24 strictly adheres to this property.

For the first-order scheme, a few simple graphs will be used to illustrate the choice of $\phi_{j+\frac{1}{2}}$ that must be made to satisfy the cell entropy inequality, given by Eq. 23. If the two cell values, u_j and u_{j+1} are equal, then there exists no flux through the cell interface and a choice of $\phi_{j+\frac{1}{2}} = 1$ is made. If the two values are different, but both have the same sign, then the choice

$$\phi_{j+\frac{1}{2}} = 1 - (u_j), \tag{25}$$

will always satisfy Eq. 23. This is demonstrated graphically by Fig. 2.3.

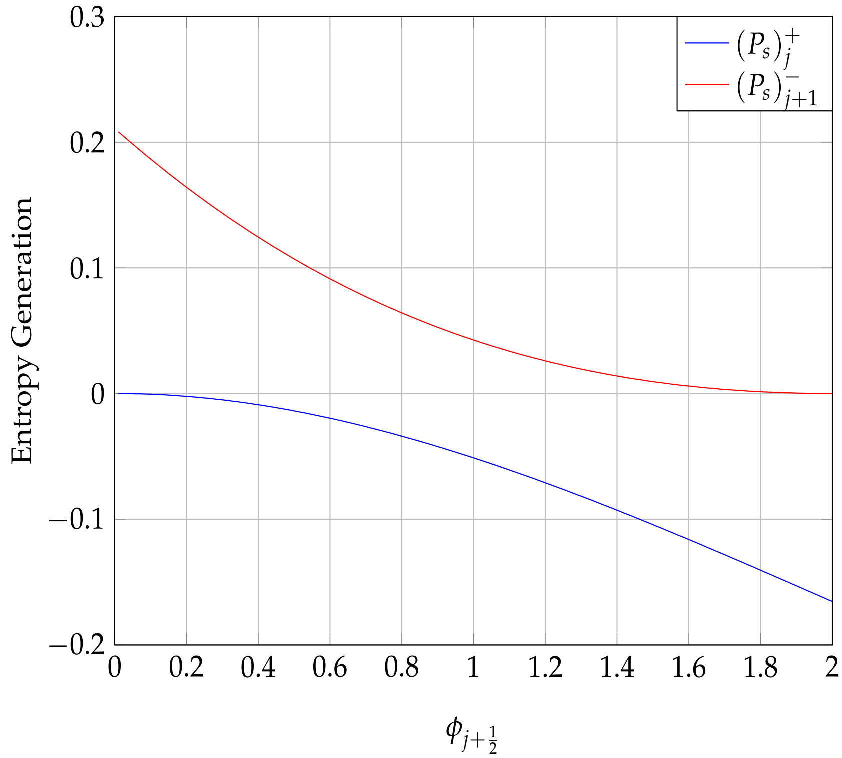

FIGURE 2.3. Cell entropy production rates for Burgers' equation during a strong compression

If the two values have opposite signs, the curves will contain extrema. In Fig. 2.4 below, recreated from work by Merriam uses cell values are chosen to be $u_j = -1$ and $u_{j+1} = 1$. These values for each cell indicate the presence of a sonic point. This is important later on because this specific choice of cell values results in some limitations of that method. This choice is made specifically so that $u_{j+\frac{1}{2}} = 0$, which may not always be the case if the cell values have opposite signs. This is a situation where neither of the endpoints of $\phi_{j+\frac{1}{2}}$ satisfy Eq. 23. For this case, it can clearly be seen graphically that $\phi_{j+\frac{1}{2}} = 1$, satisfies these constraints.

For Burgers' equation, this choice amounts to

$$\phi_{j+\frac{1}{2}} = -\frac{2u_j}{u_{j+1} - u_j}. \tag{26}$$

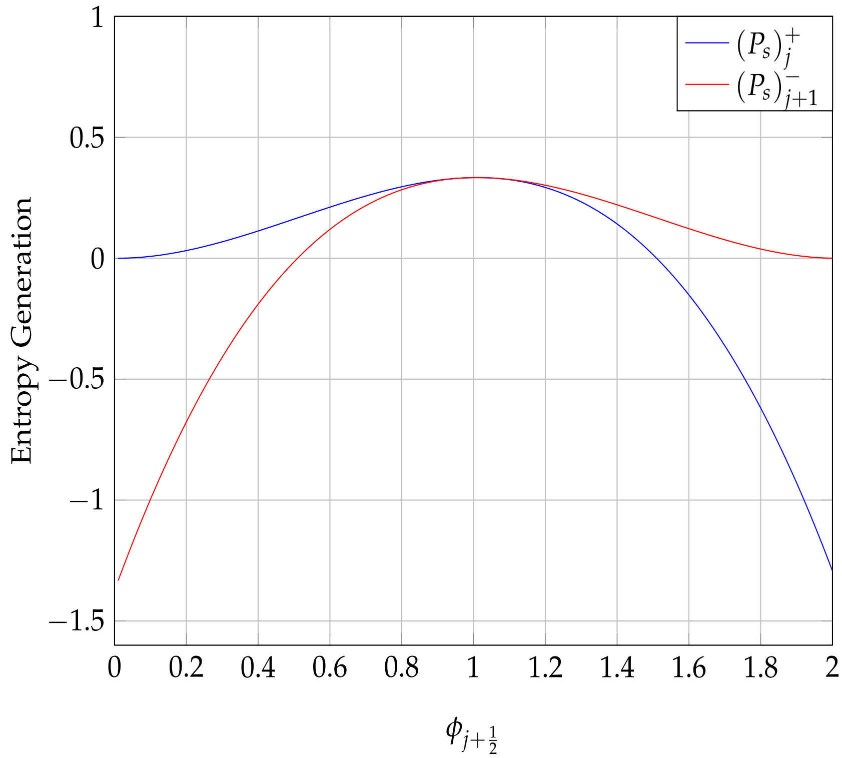

FIGURE 2.4. Cell entropy production rates for Burgers' equation during a sonic expansion

The next figure depicts criteria from the existing limiting method for a shock, which is given by $u_j = 1$ and $u_{j+1} = -0.45$. This choice for $\phi_{j+\frac{1}{2}}$ is given as

$$\phi_{j+\frac{1}{2}} = 1 - \left(\frac{u_j + u_{j+1}}{2}\right), \tag{27}$$

which is depicted below in Fig. 2.5.

In summary, a first-order scheme proposed by Merriam which satisfies a cell entropy inequality is defined as

$$\phi_{j+\frac{1}{2}} = \begin{cases} -\frac{2u_j}{u_{j+1}-u_j} & \text{Sonic Points}(u_j \leq 0 < u_{j+1}) \\ 1 - \left(\frac{u_j+u_{j+1}}{2}\right) & \text{Shocks}(u_j \geq 0 > u_{j+1}) \\ 1 - (u_j) & \text{Elsewhere}(u_j u_{j+1} \geq 0). \end{cases} \tag{28}$$

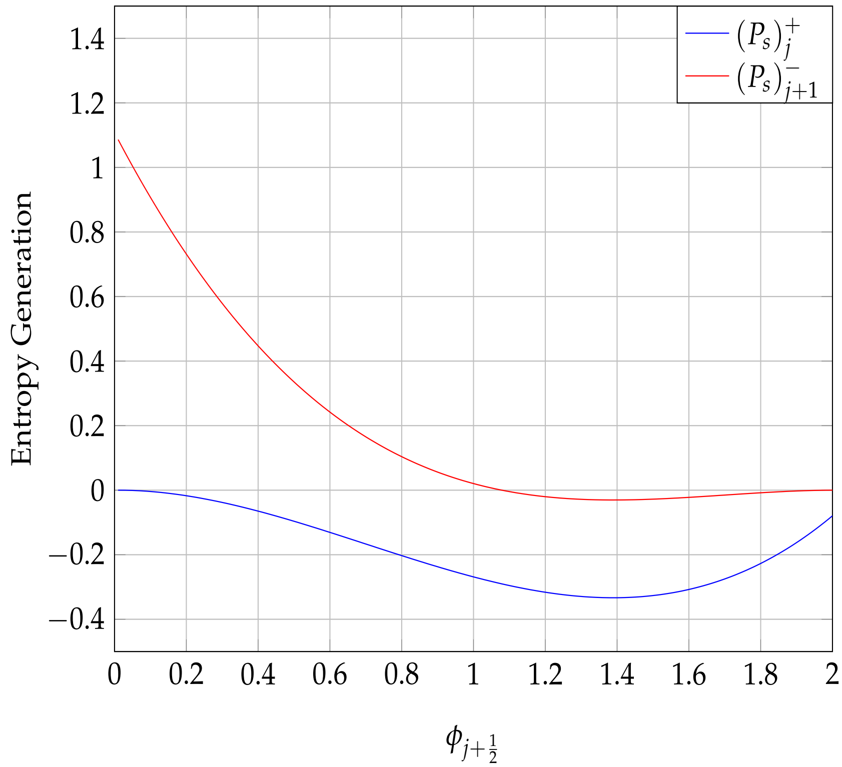

FIGURE 2.5. Cell entropy production rates for Burgers' equation through a moving shock

It is here that we must explore a few limitation in the derivation of this first-order method proposed by Merriam. In order to satisfy the general condition, Eq. 23, no assumptions must be made about the cell values until after an expression for $\phi_{j+\frac{1}{2}}$ is fully derived. Only then can we place specific constraints on the cell values that ultimately determine the values of $\phi_{j+\frac{1}{2}}$. This can be done explicitly, algebraically, by finding the value for $\phi_{j+\frac{1}{2}}$ that satisfies

$$\Delta x_j (P_s^+)_j = -(\frac{\partial S}{\partial u})_j (f_{j+\frac{1}{2}} - f_j) + (F_{j+\frac{1}{2}} - F_j) \geq 0, \tag{29}$$

as well as $(P_s^-)_j \geq 0$ shifted by one cell index, as to solve for a $\phi_{j+\frac{1}{2}}$ value for one respective face. This is given as $(P_s^-)_{j+1}$ and is defined as

$$\Delta x_j (P_s^-)_{j+1} = -(\frac{\partial S}{\partial u})_{j+1} (f_{j+1} - f_{j+\frac{1}{2}}) + (F_{j+1} - F_{j+\frac{1}{2}}) \geq 0. \tag{30}$$

Both Eq. 29 and Eq. 30 depend only on the single free parameter $\phi_{j+\frac{1}{2}}$, that ultimately determine $f_{j+\frac{1}{2}}$ and $F_{j+\frac{1}{2}}$. Since this is the case, we can simply substitute, and solve for what values of $\phi_{j+\frac{1}{2}}$ make both equations greater than or equal to zero, and then impose restrictions on the cell values. The coupling of these constraints is a more robust approach than solving for each inequality individually. Theoretically, this methodology allows a general entropy stability approach to be applied to any high-order schemes as well as a system of equations. However, it will be shown later on that solving these constraints, and the implementation of the resulting logical conditions can be rather impractical.

This substitution was done in Maxima, and the result for $(P_s^+)_j \geq 0$ was finalized as

$$\frac{-\phi_{j+\frac{1}{2}}^2 (u_{j+1} - u_j)^2 (\phi_{j+\frac{1}{2}} u_{j+1} - \phi_{j+\frac{1}{2}} u_j + 3u_j)}{12} \geq 0. \tag{31}$$

Subsequently, the result for $(P_s^-)_{j+1} \geq 0$ was finalized as

$$\frac{(\phi_{j+\frac{1}{2}} - 2)^2 (u_{j+1} - u_j)^2 (\phi_{j+\frac{1}{2}} u_{j+1} + u_{j+1} - \phi_{j+\frac{1}{2}} u_j + 2u_j)}{12} \geq 0. \tag{32}$$

Solving for $\phi_{j+\frac{1}{2}}$ results in two criteria that must be simultaneously met in order to satisfy the cell entropy inequality. These are

$$\phi_{j+\frac{1}{2}} (u_{j+1} - u_j) \leq -3u_j, \tag{33}$$

$$\phi_{j+\frac{1}{2}} (u_{j+1} - u_j) \geq -u_{j+1} - 2u_j. \tag{34}$$

In deriving these two equations that must be satisfied for the cell entropy inequality to hold, we have made no assumptions, and thus the result here is exact. This will be true for all cell values. Now, the task is to develop a choice for $\phi_{j+\frac{1}{2}}$ which is dependent on the cell values. The following cases will illustrate these choices.

Case 1: $(u_j = u_{j+1})$

Then as stated previously, there is no flux at the cell interface, and a choice of $\phi_{j+\frac{1}{2}} = 1$ is made.

Case 2: $(u_{j+1} - u_j) > 0$

This results in a change in the two criteria for given by Eq. 33, and Eq. 31 respectively as

$$1 - \frac{3u_j}{u_{j+1} - u_j} \leq \phi_{j+\frac{1}{2}} \leq \frac{-3u_j}{u_{j+1} - u_j}, \tag{35}$$

$$2 - \frac{3u_j}{u_{j+1} - u_j} \leq \phi_{j+\frac{1}{2}} \leq \frac{-3u_j}{u_{j+1} - u_j}. \tag{36}$$

– **Sub-case 2.1** $(u_j > 0)$

Then the right hand side of Eq. 35 is negative. Since we know that $\phi_{j+\frac{1}{2}}$ must be greater than zero, a choice to set $\phi_{j+\frac{1}{2}}$ equal to zero is made here, $\phi_{j+\frac{1}{2}} = 0$.

– **Sub-case 2.2** $(u_j < 0)$

$\phi_{j+\frac{1}{2}} = \frac{-3u_j}{u_{j+1}+u_j}$.

– **Sub-case 2.3** $(0 > u_{j+1} > u_j)$

Then the only choice is $\phi_{j+\frac{1}{2}} = 2$ because the right hand side (RHS) of Eq. 33 is greater than 2.

Case 3: $(u_{j+1} - u_j) < 0$

The only possible choice for $\phi_{j+\frac{1}{2}}$ here is $\phi_{j+\frac{1}{2}} = 1 - \frac{u_{j+1}+u_j}{2}$ because the left hand side (LHS) of Eq. 33 is greater than 2.

It's clear here that this approach is much more thorough and general than that of the reference work. These constraints on $\phi_{j+\frac{1}{2}}$ provide a general solution to the cell entropy inequality constraint that must be satisfied. Results for these two methods are compared in the next section.

2.3. First-Order Comparison of Methods

In this section some results will be provided in order to compare the two first-order entropy stable methodologies as detailed above. In order to accurately compare the two methods, the same problem setup will be used. The initial condition is a two-period sin wave of unit amplitude. This problem contains sonic points and will develop shocks. The domain length specified for this problem is 4π. In Fig. 2.6 the initial condition is plotted, along with the solution at time $t = 1.0s$ without the application of any limiter. This is done to illustrate the effect of a discontinuous region on the solution that produces oscillations.

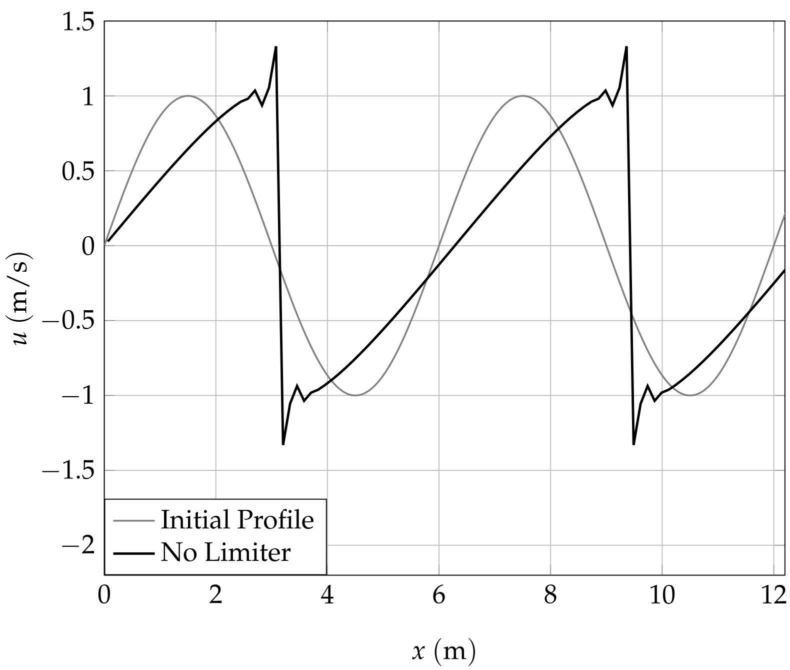

Figure 2.6. Initial condition of Burgers' equation and onset of shock

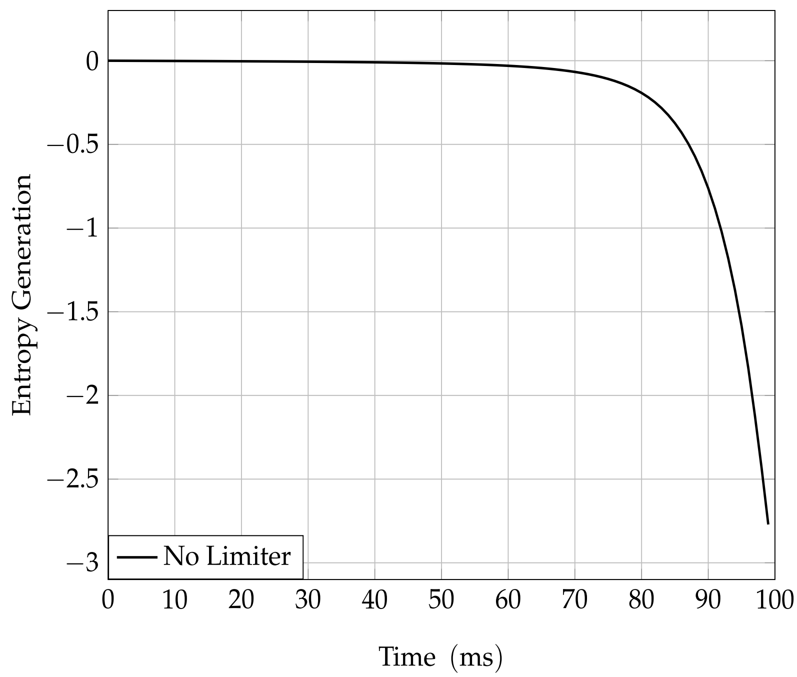

FIGURE 2.7. Entropy generation over time

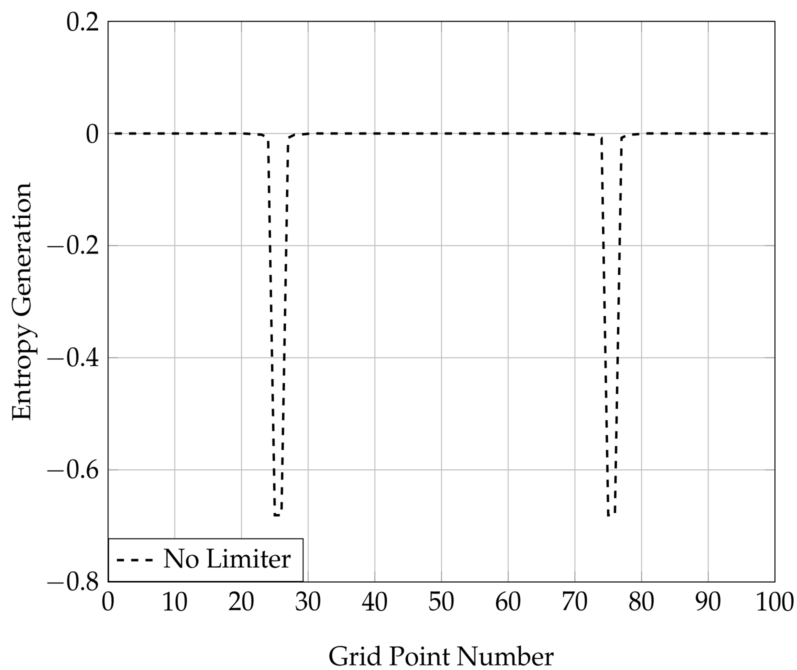

FIGURE 2.8. Entropy generation in domain

Additionally, Fig. 2.7 shows the entropy generated in the domain over time. It can clearly be seen that without the use of a limiter that enforces entropy production to move in the correct direction, the second law of thermodynamics is violated throughout the domain. The second law tells us that we should expect entropy to increase as time moves forward, however, without a limiter that enforces this criteria, we are guaranteed no such behavior. Fig 2.8 shows the entropy generated in each discrete cell at the final time. The entropy generated in the domain is zero in the smooth regions, and should be a positive quantity around the shock. However, without the enforcement of an entropy stability criteria, the entropy generation around the shock is negative. It is the aim of this work to ensure positive entropy production in the discontinuous region consistent with the second law of thermodynamics.

Fig. 2.9 shows the solution when the existing first-order limiter is applied. It can be seen that the solution here smooths out the region affected by the shocks that form. In Fig. 2.10 the proposed limiter is also compared to the existing limiter, and the solutions match up nicely.

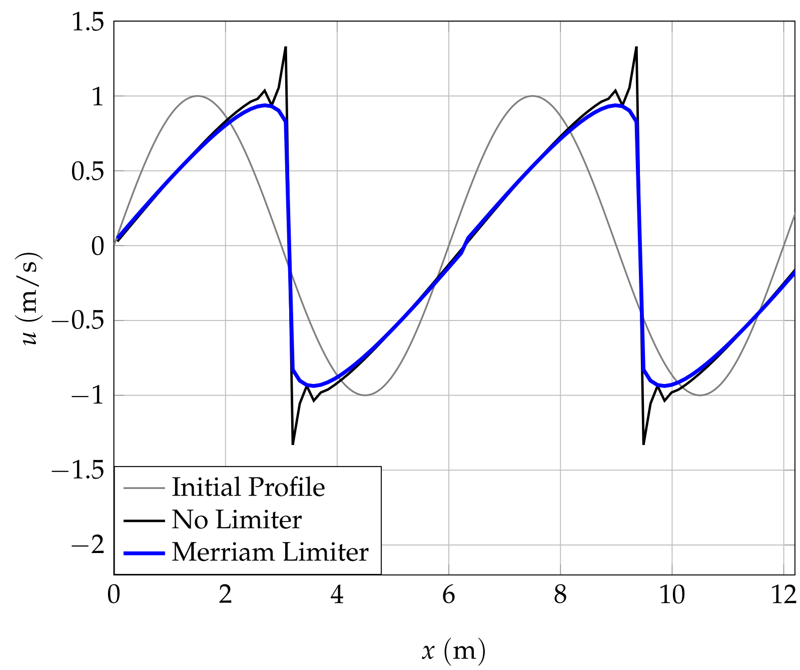

FIGURE 2.9. Oscillations near shocks are suppressed by the existing method

Fig. 2.11 shows the comparison of the entropy generated throughout the domain between the existing method, and the new method devised by the present work at the final

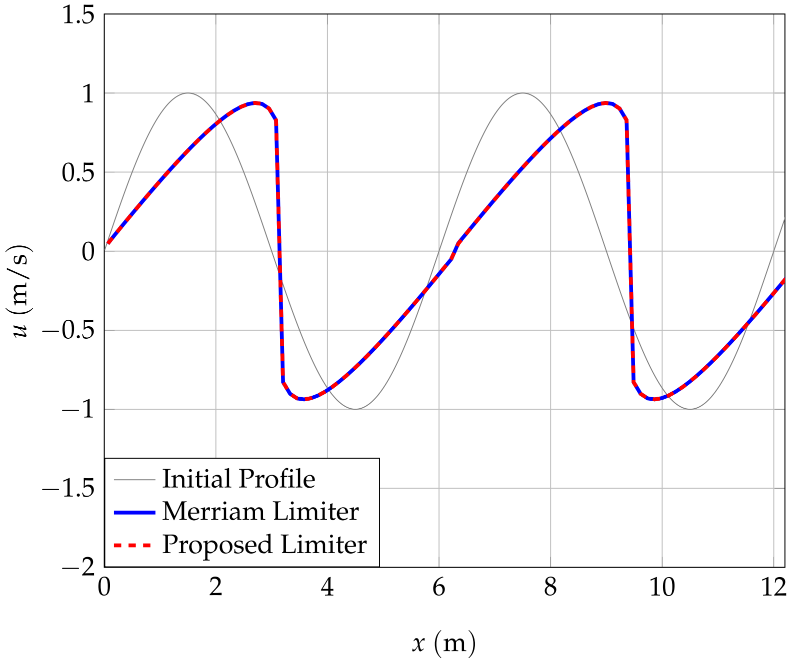

FIGURE 2.10. Comparison of the numerical solutions limited by the existing method and the new method

time step. Large discontinuities occur where the shock interface occurs in the solution as expected. The entropy generated around the shock interface is positive, and the only place in the domain where entropy is produced. This is both consistent with the second law of thermodynamics, and provides insight into the areas of the domain that could benefit the most from optimization.

Fig. 2.12 shows the comparison of total entropy in the domain over the duration of the simulation. This clearly demonstrates entropy flowing in the correct direction, and is consistent with what we expect from the physics of the second law. Entropy increases when the onset of shocks occur in the domain, and grows until the simulation is completed.
The next figure shows the comparison of the $\phi_{j+\frac{1}{2}}$ throughout the domain at the time $t = 3.50s$ for both of the limiting methods.

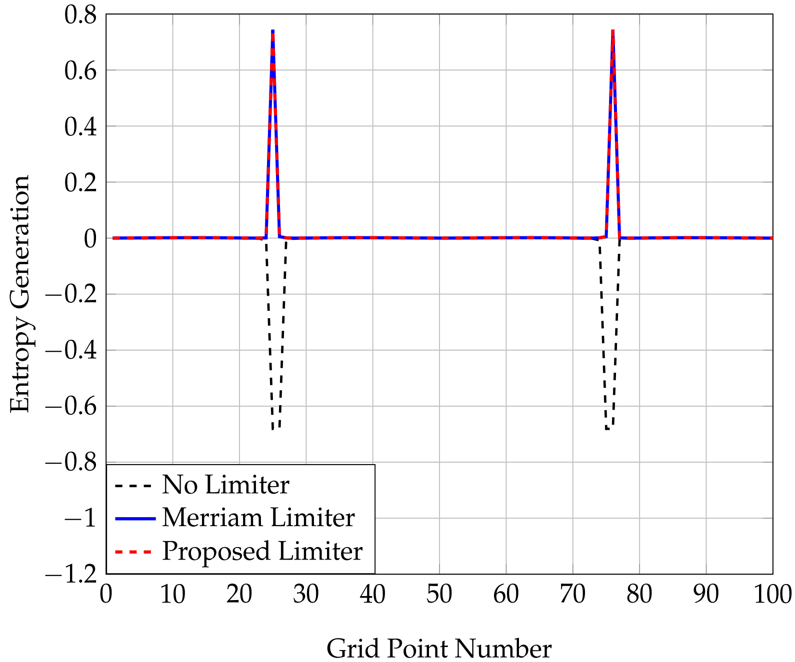

FIGURE 2.11. Comparison of the entropy generated in the domain by existing method, the new method, and without a limiter applied

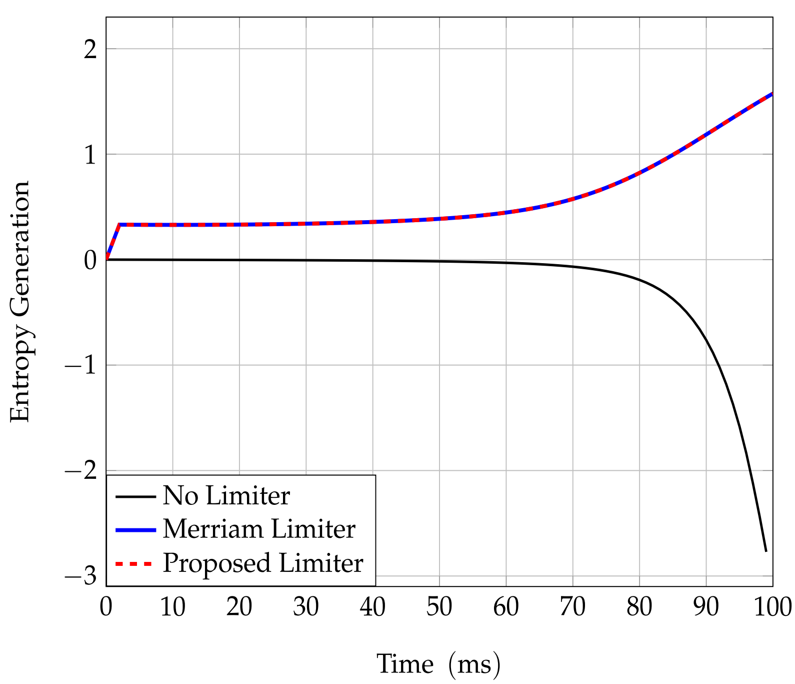

FIGURE 2.12. Comparison of the entropy generated over time by the existing method, the new method, and without a limiter applied

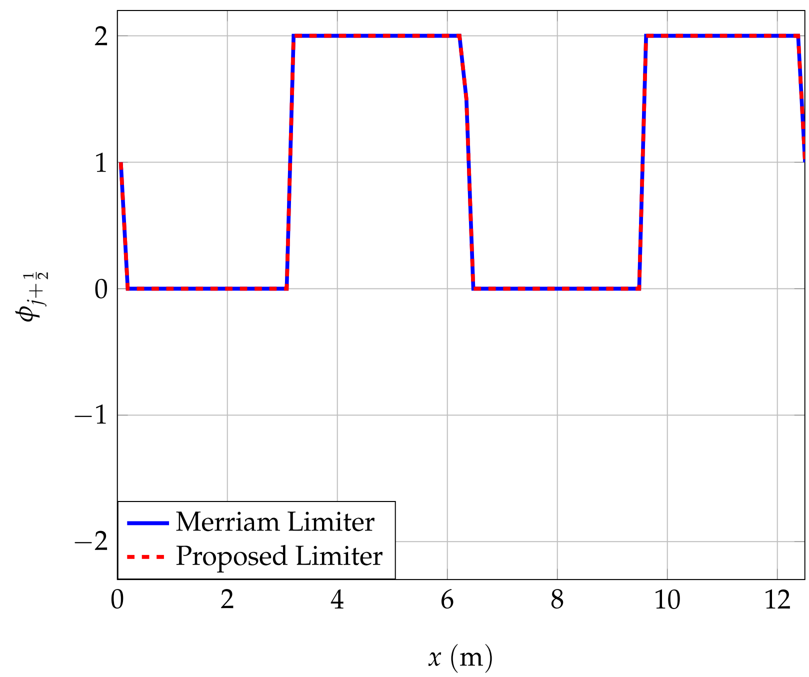

FIGURE 2.13. Comparison of $\phi_{j+\frac{1}{2}}$ values for the existing method and the new method at the onset of shocks

It is easy to tell that this scheme is first-order accurate because of the values for $\phi_{j+\frac{1}{2}}$ that the method selects. Since the values oscillate between zero and two, the method is always choosing either the left or right cell value to be used as the face value, which is exactly what is expected in the first-order accurate scheme. The two limiting methods proposed here match up nicely, as is a consequence of a first-order scheme. Larger differences will start to arise as a consequence of the limiting method moving to higher order accuracy, and will clearly be demonstrated in the second-order proposed limiting scheme.

2.4. Second-Order Entropy Stable Scheme

The scheme described in the previous section, 2.2, is consistent and stable but is only first-order accurate. This scheme is overly dissipative, and more entropy is produced than the physics of the problem requires. This section explores Merriam's approach to second-order accuracy, as well as its limitations and again, a more robust approach. While the first-order scheme required that each cell half produce entropy and be a positive quantity, the second-order scheme only requires that the entropy production in each full cell be positive. This amounts to

$$(P_s)_j = (P_s^+)_j + (P_s^-)_j \geq 0. \tag{37}$$

It may also be possible to write the entropy production in each cell as

$$(P_s)_j = \overline{(P_s^+)_j} + \overline{(P_s^-)_j}, \tag{38}$$

where the value $\overline{(P_s^+)_j}$ is computed using the linear average of u, $u_{j+\frac{1}{2}} = \frac{u_{j+1}+u_j}{2}$ rather than a value that is dependent on ϕ, as $u_{j+\frac{1}{2}} = u_j + \frac{1}{2}\phi_{j+\frac{1}{2}}(u_{j+1} - u_j)$. Under this convention, this necessitates modifying the cell entropy inequality to satisfy the following six individual constraints that can be coupled into three equations as

$$(P_s^+)_j \geq \min\left(-\overline{(P_s^-)_j}, 0\right), \tag{39}$$

$$(P_s^-)_j \geq \min\left(-\overline{(P_s^+)_j}, 0\right), \tag{40}$$

$$(P_s^-)_{j+1} \geq \min\left(-\overline{(P_s^+)_{j+1}}, 0\right). \tag{41}$$

In the previous section, it was shown that $\phi_{j+\frac{1}{2}}$ can be chosen in such a way to make both $(P_s^+)_j$ and $(P_s^-)_j$ individually positive. This is sufficient to show that in discontinuous regions, the second-order scheme reverts to first-order, just as expected. Merriam proposes these strict limitations on a second-order accurate scheme, but details on how they are explicitly satisfied are lacking. Instead, the reference work found values for $\phi_{j+\frac{1}{2}}$ that make the entropy production in each cell greater than or equal to zero. Key assumptions were made in the reference work in order to develop a scheme that is second-order accurate and entropy stable that don't necessarily cover all possible flows.

For example, in the derivation of the entropy generation term $(P_s^+)_j$, it was assumed that if $\frac{2}{3}(u_{j+\frac{1}{2}} - u_j)$ is small compared to u_j, then the term can be neglected, and in this case, $(P_s^+)_j$ for Burgers' equation reduces to

$$(P_s^+)_j \approx -\frac{1}{2}\phi_{j+\frac{1}{2}} u_j (u_{j+1} - u_j)^2. \tag{42}$$

It was further assumed that if $(P_s^-)_{j+1}$ can be approximated in the same way, and if both $u_j > 0$ and $u_{j+1} > 0$, then the value for $\phi_{j+\frac{1}{2}}$ can be approximated as

$$\phi_{j+\frac{1}{2}} \leq \min\left(1, |r_j|\right), \tag{43}$$

where $|r_j|$ is defined as

$$|r_j| = \left(\frac{u_j - u_{j-1}}{u_{j+1} - uj}\right)^2, \tag{44}$$

and if both $u_j < 0$ and $u_{j+1} < 0$, then the limiter in this case turns out to be

$$\phi_{j+\frac{1}{2}} \geq \max\left(1, 2 - \frac{1}{|r_j|}\right). \tag{45}$$

These approximations are only valid for specific values of u_{j-1}, u_j and u_{j+1}, as well as the relationship between the cell values. It is the interest of this thesis to satisfy the cell entropy constraints as proposed by Merriam explicitly and precisely derived in the same way as it was done in the first-order scheme. Since the over-line values $\left(\overline{(P_s^-)_j}, \overline{(P_s^+)_j}, \overline{(P_s^-)_{j+1}}\right)$ don't depend on the value of $\phi_{j+\frac{1}{2}}$, it is possible again to algebraically solve for the value of $\phi_{j+\frac{1}{2}}$ that satisfies these constraints.

We have already satisfied two of the above six constraints from the first-order scheme. Those are $(P_s^+)_j \geq 0$ and $(P_s^-)_{j+1} \geq 0$ which resulted in Eq. 33 and 31. We now must also satisfy the additional constraint from Eq. 37, $(P_s^-)_j \geq 0$. Solving this equation for $\phi_{j-\frac{1}{2}}$ results in the equation

$$\frac{-(\phi_{j-\frac{1}{2}} - 2)^2(u_{j-1} - u_j)(\phi_{j-\frac{1}{2}} u_{j-1} - 2u_{j-1} - \phi_{j-\frac{1}{2}} - u_j)}{12} \geq 0, \tag{46}$$

which can be factored into

$$\phi_{j-\frac{1}{2}}(u_{j-1} - u_j) \leq 2u_{j-1} + u_j. \tag{47}$$

We can now use the same technique that we used for the first-order scheme to determine which values of u_{j-1} and u_j determine the constraints of the choices for $\phi_{j-\frac{1}{2}}$.

Case 1: $(u_j = u_{j-1})$ - there is no flux at the cell interface, and a choice of $\phi_{j-\frac{1}{2}} = 1$ is made.

Case 2: $(u_{j-1} - u_j) > 0$

This results in a change in the criteria as

$$\phi_{j-\frac{1}{2}} \leq \frac{2u_{j-1} + u_j}{u_{j-1} - u_j}. \tag{48}$$

– **Sub-case 2.1** $(u_j > 0)$

Then the right hand side of Eq. 48 is positive. Since we know that $\phi_{j-\frac{1}{2}}$ must be greater than zero. If the RHS is greater than 2, then $\phi_{j-\frac{1}{2}} = 2$, otherwise,

$\phi_{j-\frac{1}{2}} = \frac{2u_{j-1}+u_j}{u_{j-1}-u_j}$.

– **Sub-case 2.2** ($u_j < 0$)

Then the value of $\phi_{j+\frac{1}{2}}$ depends on the sign of the numerator of Eq. 48. Depending on the sign, $\phi_{j-\frac{1}{2}} = 0$ or $\phi_{j-\frac{1}{2}} = \frac{2u_{j-1}+u_j}{u_{j-1}-u_j}$.

– **Sub-case 2.3** ($0 > u_{j+1} > u_j$)

Then the only choice can be $\phi_{j+\frac{1}{2}} = 0$.

Case 3: $(u_{j+1} - u_j) < 0$

Then the value of $\phi_{j+\frac{1}{2}}$ depends again, on the numerator of Eq. 48. Depending on the sign and the relationship between the two cell values, $\phi_{j-\frac{1}{2}} = 0$ or $\phi_{j-\frac{1}{2}} = \frac{2u_{j-1}+u_j}{u_{j-1}-u_j}$ or $\phi_{j-\frac{1}{2}} = 2$.

For a second-order scheme, it is not enough to simply guarantee that each cell entropy generation quantity be greater than or equal to zero. We must explicitly satisfy the following six constraints that come from Eq. 38 - 40 that can be broken up as:

$$(P_s^+)_j \geq 0 \quad \& \quad (P_s^-)_{j+1} \geq 0, \tag{49}$$

$$(P_s^+)_j \geq 0 \quad \& \quad (P_s^-)_{j+1} \geq \left(-\overline{(P_s^+)}_{j+1}\right), \tag{50}$$

$$(P_s^+)_j \geq \left(-\overline{(P_s^-)}_j\right) \quad \& \quad (P_s^-)_{j+1} \geq 0, \tag{51}$$

$$(P_s^+)_j \geq \left(-\overline{(P_s^-)}_j\right) \quad \& \quad (P_s^-)_{j+1} \geq \left(-\overline{(P_s^+)}_{j+1}\right), \tag{52}$$

$$(P_s^-)_j \geq 0, \tag{53}$$

$$(P_s^-)_j \geq \left(-\overline{(P_s^+)}_j\right). \tag{54}$$

The only remaining difficulty is satisfying the coupling of constraints that correspond to the value for $\phi_{j+\frac{1}{2}}$.

This amounts to simultaneously satisfying Eq. 49, or Eq. 50, or Eq. 51, or Eq. 52 which are explicitly dependent on $\phi_{j+\frac{1}{2}}$. Eq. 53 and 54 do not involve $\phi_{j+\frac{1}{2}}$ and so don't need to be explicitly solved for, they can simply exist as an additional check ensure the values for each face value are set appropriately. The $\phi_{j+\frac{1}{2}}$ values associated with the solutions to these inequalities involve solving a cubic inequality for each equation above. When making the substitutions to solve for $\phi_{j+\frac{1}{2}}$, Maxima can be used to rearrange the LHS of Eq. 51 which leaves us with

$$\frac{-\phi_{j+\frac{1}{2}}^2(u_{j+1}-u_j)^2(\phi_{j+\frac{1}{2}}u_{j+1}-\phi_{j+\frac{1}{2}}u_j+3u_j)}{12}+\frac{(u_{j-1}-u_j)^2(u_{j-1}+2u_j)}{12}\geq 0, \tag{55}$$

where the roots are given as

$$\phi_{j+\frac{1}{2}}^1=\frac{-\sqrt{3}\sqrt{4u_j^2-u_{j-1}^2}+u_{j-1}+2u_j}{2u_{j+1}-2u_j}, \tag{56}$$

$$\phi_{j+\frac{1}{2}}^2=\frac{\sqrt{3}\sqrt{4u_j^2-u_{j-1}^2}-u_{j-1}-2u_j}{2u_{j+1}-u_j}, \tag{57}$$

$$\phi_{j+\frac{1}{2}}^3=\frac{u_j-u_{j-1}}{2u_j-u_{j+1}}. \tag{58}$$

Eq. 54 can be substituted and rearranged as

$$\frac{-(\phi_{j-\frac{1}{2}}-2)^2(u_{j-1}-u_j)(\phi_{j-\frac{1}{2}}u_{j-1}-2u_{j-1}-\phi_{j-\frac{1}{2}}-u_j)}{12}-\frac{(u_{j+1}-u_j)^2(u_{j+1}+2u_j)}{12}\geq 0, \tag{59}$$

where the roots are

$$\phi^1_{j-\frac{1}{2}} = \frac{-\sqrt{3}\sqrt{4u_j^2 - u_{j+1}^2} - u_{j+1} - 4u_{j-1} + 2u_j}{2u_{j-1} - 2u_j}, \tag{60}$$

$$\phi^2_{j-\frac{1}{2}} = \frac{\sqrt{3}\sqrt{4u_j^2 - u_{j+1}^2} + u_{j+1} + 4u_{j-1} - 2u_j}{2u_{j-1} - 2u_j}, \tag{61}$$

$$\phi^3_{j-\frac{1}{2}} = \frac{u_{j+1} - 2u_{j-1} + u_j}{u_j - u_{j-1}}. \tag{62}$$

Finally, the RHS of Eq. 50 can be written as

$$\frac{(\phi_{j+\frac{1}{2}} - 2)^2(u_{j+1} - u_j)^2(\phi_{j+\frac{1}{2}}u_{j+1} + u_{j+1} - \phi_{j+\frac{1}{2}}u_j + 2u_j)}{12} - \frac{(u_{j+1} - u_{j+2})^2(2u_{j+1} + u_{j+2})}{12} \geq 0, \tag{63}$$

where the roots are

$$\phi^1_{j+\frac{1}{2}} = \frac{-\sqrt{3}\sqrt{4u_{j+1}^2 - u_{j+2}^2} - 2u_{j+1} + u_{j+2} + 4u_j}{2u_{j+1} - 2u_j}, \tag{64}$$

$$\phi^2_{j+\frac{1}{2}} = \frac{\sqrt{3}\sqrt{4u_{j+1}^2 - u_{j+2}^2} + 2u_{j+1} - u_{j+2} - 4u_j}{2u_{j+1} - 2u_j}, \tag{65}$$

$$\phi^3_{j+\frac{1}{2}} = \frac{-u_{j+1} - u_{j+2} + 2u_j}{u_j - u_{j+1}}. \tag{66}$$

These are the critical values for each cubic equation that define the entropy stability criteria. These provide the regions in which the values for ϕ provide positive stable entropy generation for each cell while remaining second-order accurate. These equations are under-constrained because they depend on four cell values, u_{j-1}, u_j, u_{j+1}, and u_{j+2} that should result in one value for $\phi_{j+\frac{1}{2}}$. Solving for these regions that provide stability algebraically, just as we did for the first-order criteria may not be as illuminating (and much more tedious) than a graphical solution would provide. Included below is a plot that shows the stable region for $\phi_{j+\frac{1}{2}}$ that lies between zero and two, as well as the entropy generation values for $(P_s^+)_j$ and $(P_s^+)_{j+1}$ that are greater than or equal to zero. One value of $\phi_{j+\frac{1}{2}}$ should satisfy these constraints which graphically corresponds to the intersection of these two curves as demonstrated below. Cell values that provide an intersection between these two curves are given here as $u_{j-1} = 1$, $u_j = 2.2$, $u_{j+1} = 1$, and $u_{j+2} = 0.5$ shown in Fig. 2.14.

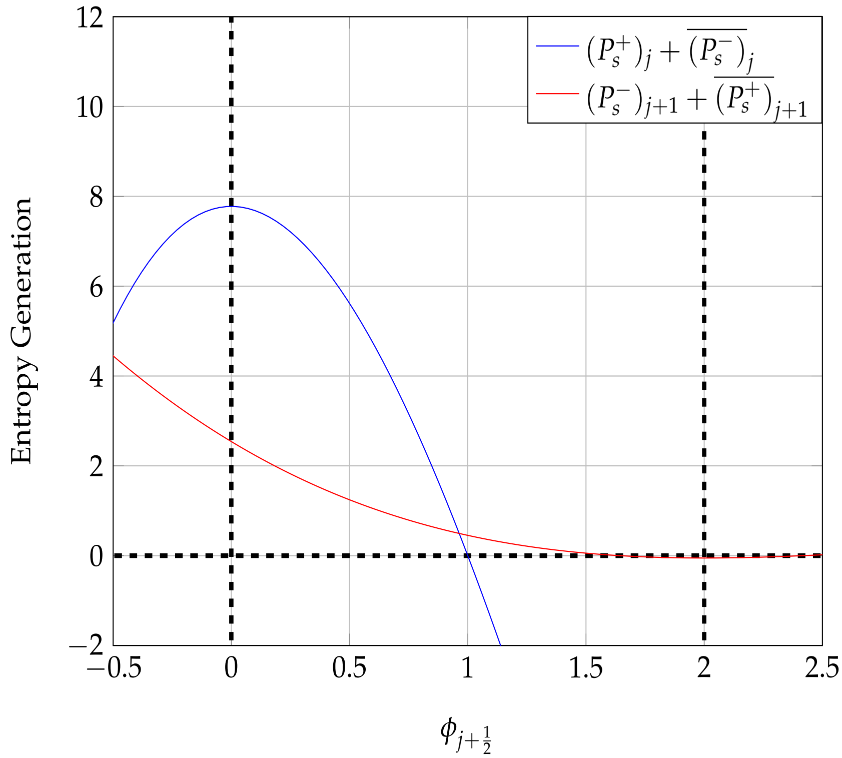

FIGURE 2.14. Stability plot demonstrating a solution for $\phi_{j+\frac{1}{2}}$

However, depending on the cell values it may not always be possible to choose a value for $\phi_{j+\frac{1}{2}}$ that provides a solution to these two equations. This can be demonstrated by instead choosing cell values that imply a strong discontinuity, and thus result in no solution for $\phi_{j+\frac{1}{2}}$. Fig. 2.15 demonstrates that since there is no explicit solution for the value of $\phi_{j+\frac{1}{2}}$, one of the endpoints must be chosen accordingly. The cell values here are given as $u_{j-1} = 0.5$, $u_j = 1$, $u_{j+1} = -4.5$, and $u_{j+2} = -2$.

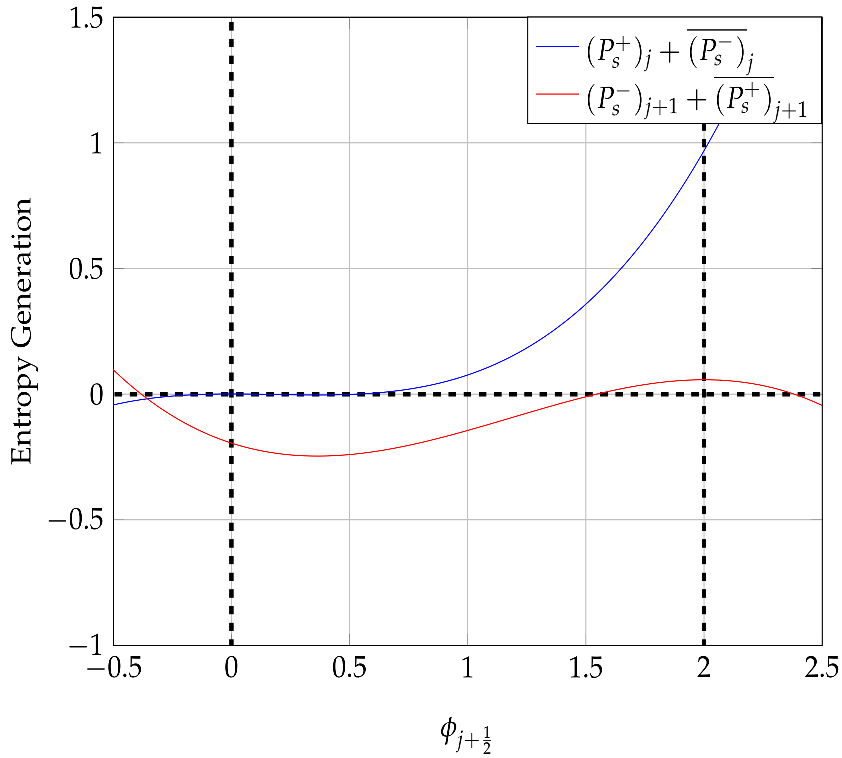

FIGURE 2.15. Stability plot demonstrating no solution for $\phi_{j+\frac{1}{2}}$

Since it was previously shown in the first-order method that $\phi_{j+\frac{1}{2}}$ can always be chosen in such a way as to make $(P_s^+)_j$ and $(P_s^+)_{j+1}$ individually positive, the only solution in this case is for the scheme to revert to first-order accurate in the discontinuous regions. In this particular case, $\phi_{j+\frac{1}{2}} = 2$ is chosen based on the flow direction.

2.5. Second-Order Comparison of Methods

In this section some results will be provided in order to compare the methodologies proposed throughout the previous sections. The graphs below show the initial condition, the second-order limited solution, and the $\phi_{j+\frac{1}{2}}$ values associated with the limited scheme for both the existing approximate second-order limiter, and the proposed limiter outlined in the previous section.

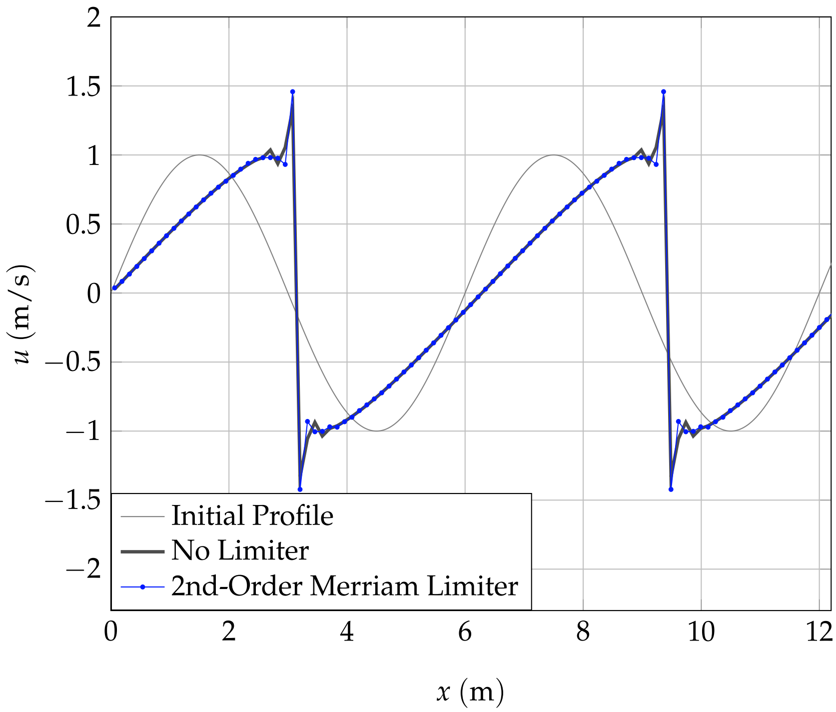

FIGURE 2.16. Existing limiting method solution plotted alongside the solution with no limiter

While the existing limiter tries to maintain second-order accuracy throughout the simulation as much as possible, the presence of oscillations still contaminate the solution. Since this is the case, the reversion to first-order that is necessary in a discontinuous region is not present in the existing limiting method. These oscillations have the potential to grow and become unstable. This can also be seen in Fig. 2.20 and the negative values of entropy generation throughout the domain in Fig. 2.21. The existing entropy generation throughout the simulation initially attempts to move in the right direction, but the onset of shocks cause the existing stability method entropy generation values to move in the negative direction, potentially as a result of the oscillations that are present in the solution.

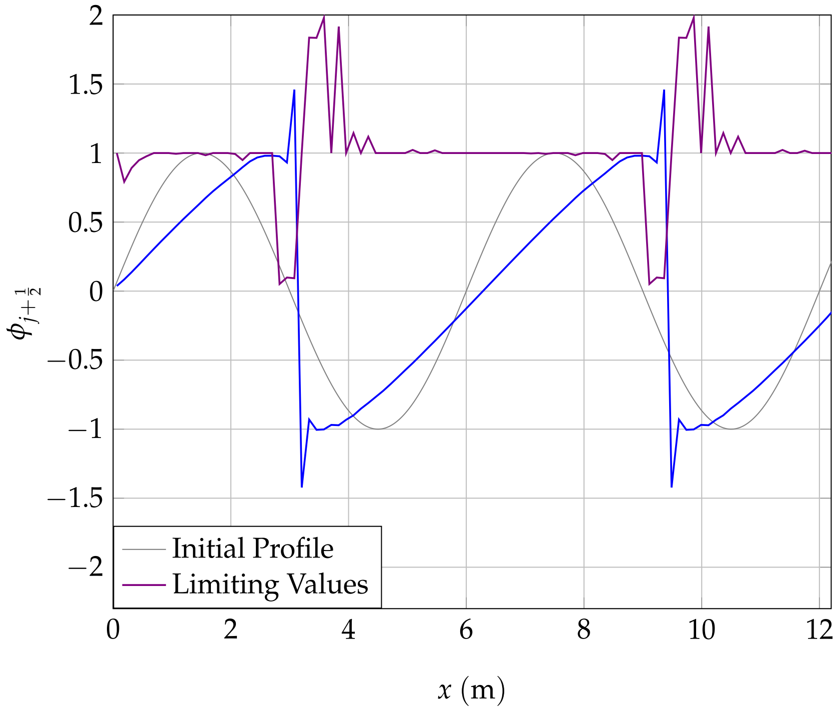

FIGURE 2.17. Existing limiting method solution plotted alongside the $\phi_{j+\frac{1}{2}}$ associated with the onset of the shock

Fig. 2.21 shows the entropy generation in each discrete cell. In the existing method, negative entropy is produced around the shock, whereas adhering to the coupling of the entropy stability constraints in the new methodology forces positive entropy production. The local negativity in the entropy production in the new method offer evidence of an imperfect limiting method. When no solution exists for a value of $\phi_{j+\frac{1}{2}}$, a choice must be made by the researcher which can perhaps be made more robust. Unavoidable, the scheme largely reverts to first-order.

As stated previously, as the order of accuracy is increased the discrepancies between both sets of logical conditions increases. It is worth noting that while the existing second-order accurate approximate limiter remains less dissipative than the method derived algebraically from the entropy inequalities in this section, it violates the second law of thermodynamics,

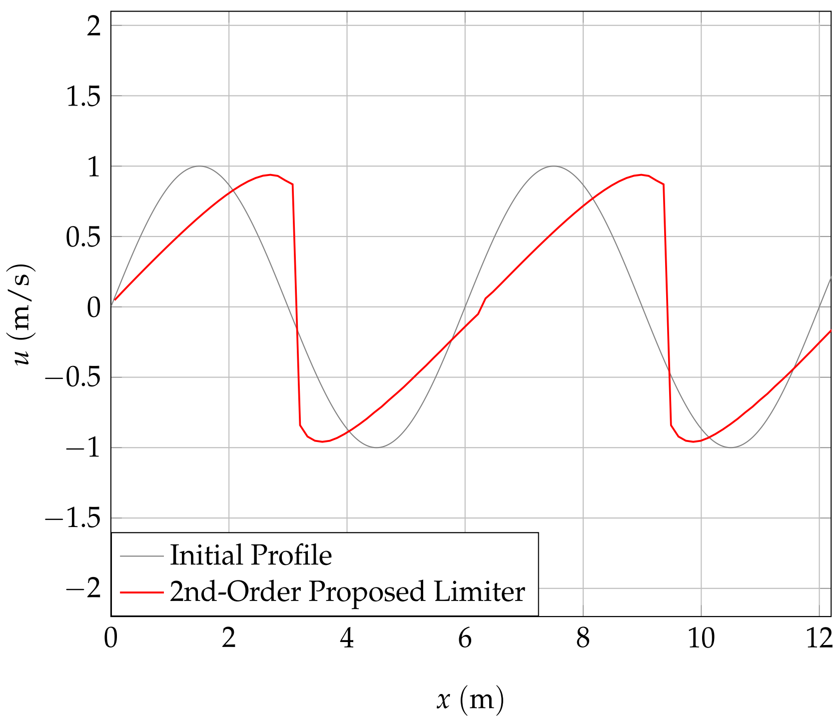

FIGURE 2.18. New limiting method solution plotted at onset of shock

the absolute crux of this limiting method. Adhering strictly to the laws of nature governed by entropy appears to degenerate the scheme to first-order accurate. Fig. 2.22 shows the new method solution plotted with the entropy generation in the domain.

It can easily be seen from the plot above that the entropy generated in the domain conforms nicely with the region where the flow develops shocks, and is zero elsewhere. From the above investigation, the new method is more restrictive than the existing method when a second-order numerical scheme is used for solving flows with discontinuities. From this exercise, both methods seem neither ideal nor viable in suppression oscillations near discontinuities while preserving the order of the underlying numerical schemes in smooth regions. Following the procedure, an application of the new method is attempted for the fourth-order numerical scheme and tested on Burgers' equation.

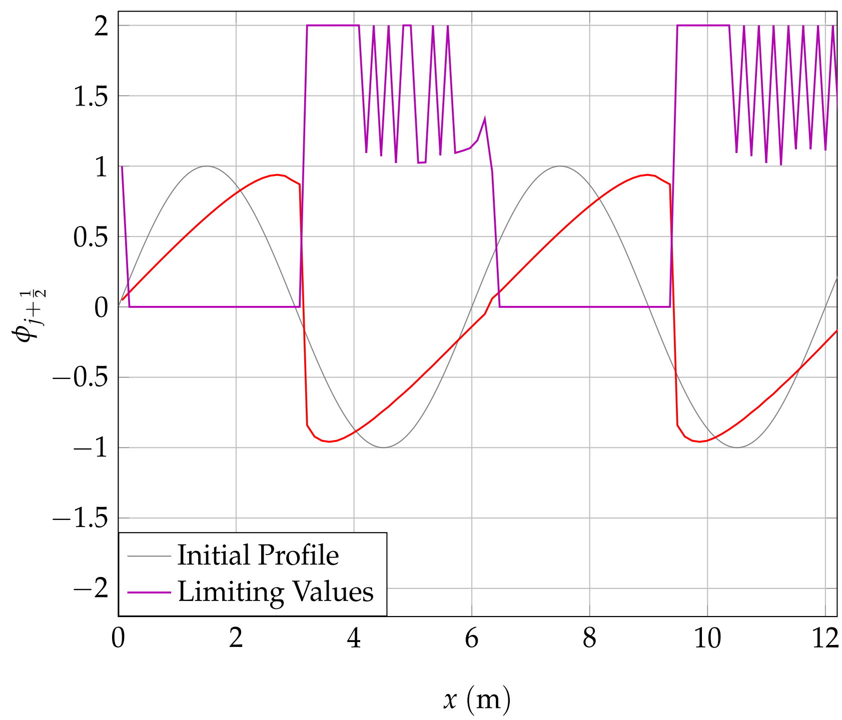

FIGURE 2.19. New limiting method solution plotted alongside the $\phi_{j+\frac{1}{2}}$ associated with the onset of the shock

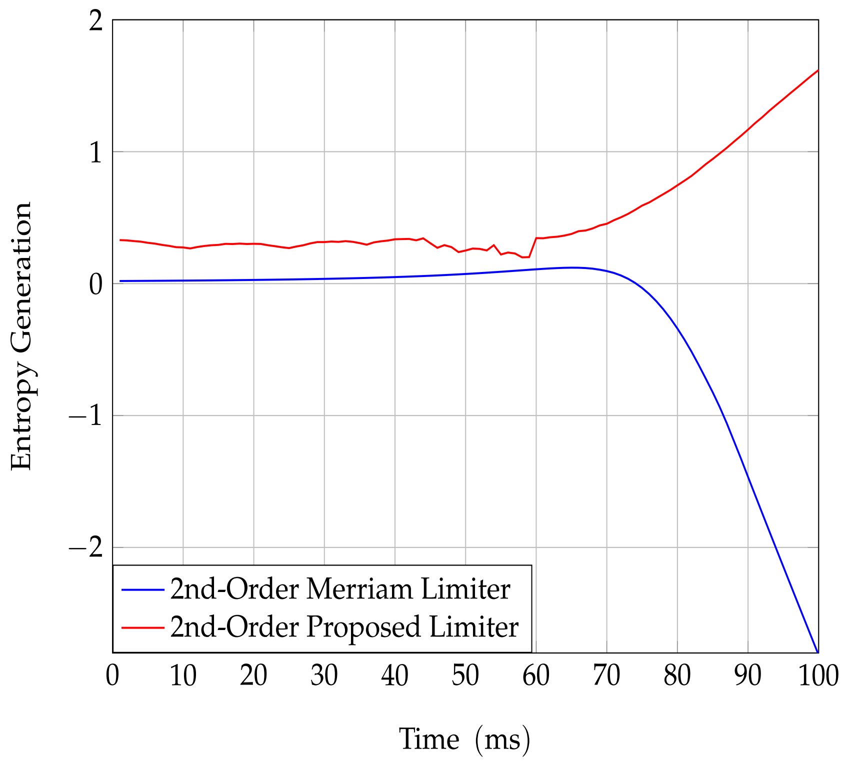

FIGURE 2.20. Comparison of entropy generation over time of both methods

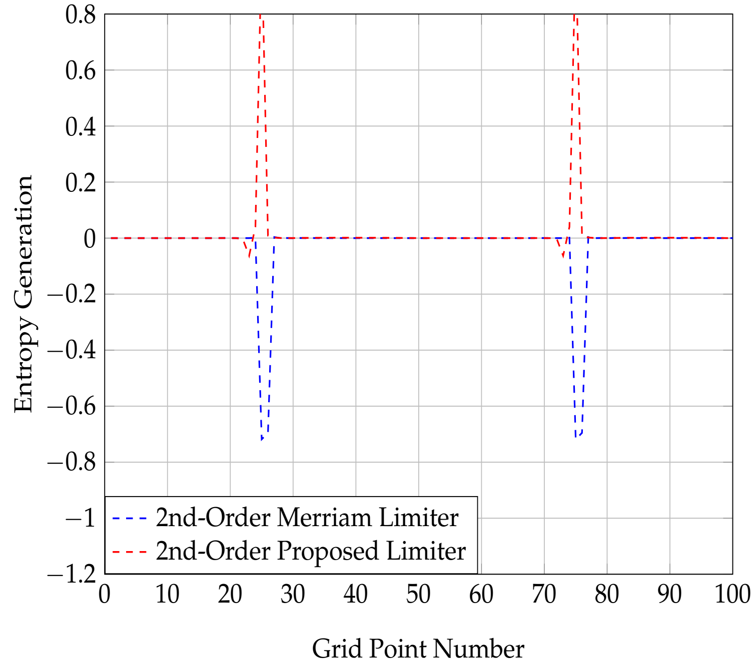

FIGURE 2.21. Comparison of entropy generation throughout the domain of both methods

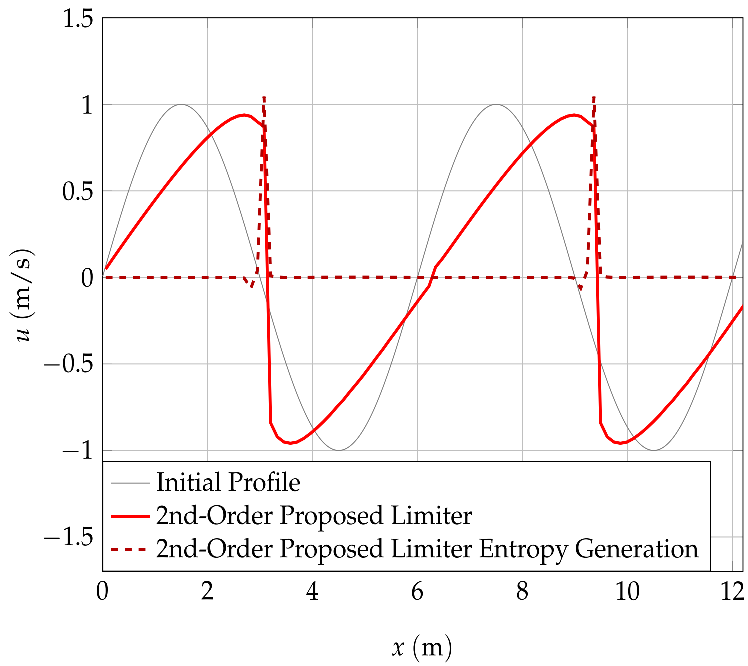

FIGURE 2.22. Solution for second-order method plotted with entropy generation in the domain

CHAPTER 3

Extension to Fourth-Order Accuracy

In the previous chapter we presented the detailed derivation of the entropy stability criteria in a second-order accurate method. This chapter will follow the same approach, but extends its application to a fourth-order scheme. A mathematical derivation of the criteria that need to be met will be shown in 3.1. A few issues with this fourth-order spatial discretization will be explored in this section as well. Section 3.2 will provide some stability regions dependent on $\phi_{j+\frac{1}{2}}$.

3.1. Fourth-Order Criteria

In the fourth-order scheme, the face value is approximated by the centered scheme given as

$$u_{j+\frac{1}{2}} = \frac{7}{12}(u_j + u_{j+1}) - \frac{1}{12}(u_{j-1} + u_{j+2}). \tag{67}$$

This interpolation gives a symmetric scheme about the $u_{j+\frac{1}{2}}$ face denoted by the dashed line below as shown in Fig. 3.1

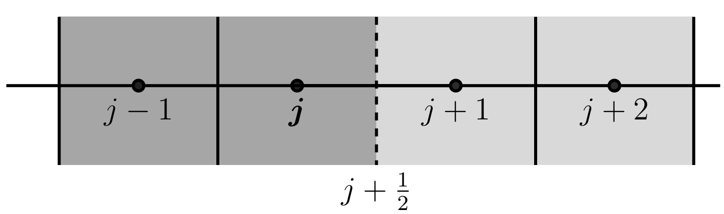

Figure 3.1. The stencil for the face interpolation in the fourth-order discretization

As stated above, this fourth-order interpolation is the symmetric average value for $u_{j+\frac{1}{2}}$. To stay consistent with the work done by Merriam, the present work assumes that the value for $\phi_{j+\frac{1}{2}}$ should still be bounded by zero and two. The face interpolation with respect to the limiting agent is given as

$$u_{j+\frac{1}{2}} = \frac{1}{2}(u_j + u_{j+1}) - \frac{1}{12}\phi_{j+\frac{1}{2}}(u_{j-1} - u_j - u_{j+1} + u_{j+2}). \tag{68}$$

This scheme will recover a weighted fourth-order stencil when $\phi_{j+\frac{1}{2}} \neq 0$ and will revert to a second-order stencil when the value for $\phi_{j+\frac{1}{2}}$ is equal to zero in the neighborhood of discontinuities. Additionally, when the value of $\phi_{j+\frac{1}{2}} = 1$, this scheme will recover the centered fourth-order discretization. This reversion to second-order will be discussed later on in this chapter.

The fourth-order approximation of $u_{j+\frac{1}{2}}$ plays a significant role in the difficulty of the satisfaction of the cell entropy inequalities. As a consequence of of this fourth-order approximation, the definition of $f(u_{j+\frac{1}{2}})$ changes to incorporate the new face interpolation, and as a result all definitions of $(P_s)_j$ change. Formally, the fourth-order cell entropy constraints that must be met are as follows.

$$(P_s^+)_j \geq \min\left(-\overline{(P_s^-)}_j, 0\right), \tag{69}$$

$$(P_s^-)_j \geq \min\left(-\overline{(P_s^+)}_j, 0\right), \tag{70}$$

$$(P_s^-)_{j+1} \geq \min\left(-\overline{(P_s^+)}_{j+1}, 0\right), \tag{71}$$

$$(P_s^+)_{j+1} \geq \min\left(-\overline{(P_s^-)}_{j+1}, 0\right). \tag{72}$$

From the fourth-order interpolation of $u_{j+\frac{1}{2}}$, the value for $\overline{(P_s)_j}$ is still determined from the value $\phi_{j+\frac{1}{2}} = 1$, which corresponds to Eq. 68. Satisfaction of these constraints can be decoupled into satisfaction of the entropy constraints that correspond to a specific face value.

For example, the constraints that depend explicitly on the face value $\phi_{j+\frac{1}{2}}$, are

$$(P_s^+)_j \geq \min\left(-\overline{(P_s^-)}_j, 0\right), \tag{73}$$

$$(P_s^-)_{j+1} \geq \min\left(-\overline{(P_s^+)}_{j+1}, 0\right). \tag{74}$$

Both of the values on the RHS of the inequality depend on $\phi_{j+\frac{1}{2}}$, while the averaged values on the LHS of the inequality strictly involve cell values. These constraints can further be decoupled into the following four combinations that must be satisfied in order for the scheme to nonnegatively produce entropy

$$(P_s^+)_j \geq 0 \quad \& \quad (P_s^-)_{j+1} \geq 0, \tag{75}$$

$$(P_s^+)_j \geq 0 \quad \& \quad (P_s^-)_{j+1} \geq -\overline{(P_s^+)}_{j+1}, \tag{76}$$

$$(P_s^+)_j \geq -\overline{(P_s^-)}_j \quad \& \quad (P_s^-)_{j+1} \geq 0, \tag{77}$$

$$(P_s^+)_j \geq -\overline{(P_s^-)}_j \quad \& \quad (P_s^-)_{j+1} \geq -\overline{(P_s^+)}_{j+1}, \tag{78}$$

just as was done for the second-order method outlined in chapter 2. If the overline values that appear on the LHS of the inequalities are greater than zero, then the RHS becomes greater than those respective values. However, if the overline values are less than zero, then the RHS must be greater than zero. Using the same procedure from the first-order and second-order methodology from chapter 2, we can develop specific constraints on the cell values that will ultimately determine the values for $\phi_{j+\frac{1}{2}}$ which satisfy the inequalities. Again, utilizing Maxima and substituting in the fourth-order interpolation for $u_{j+\frac{1}{2}}$, the result for $(P_s^+)_j \geq 0$

is given as

$$-(\phi_{j+\frac{1}{2}}u_{j+1} + 6u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j - 6u_j)^2 \frac{(\phi_{j+\frac{1}{2}}u_{j+1} + 6u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j + 12u_j)}{2592} \geq 0,$$

which can be simplified to

$$\phi_{j+\frac{1}{2}}(u_{j+1} - u_{j+2} - u_{j-1} + u_j) \leq -6u_{j+1} - 12u_j. \tag{79}$$

Similarly, a substitution for $u_{j+\frac{1}{2}}$ can be made in $(P_s^-)_{j+1} \geq 0$ which leaves

$$((\phi_{j+\frac{1}{2}}u_{j+1} - 6u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j + 6u_j)^2 \frac{(\phi_{j+\frac{1}{2}}u_{j+1} + 12u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j + 6u_j)}{2592} \geq 0,$$

which can be simplified to

$$\phi_{j+\frac{1}{2}}(u_{j+1} - u_{j+2} - u_{j-1} + u_j) \geq -6u_j - 12u_{j+1}. \tag{80}$$

We can now use both Eq. 79 and Eq. 80 to determine the constraints that need to be placed on the values of $\phi_{j+\frac{1}{2}}$ that will satisfy the cell entropy inequalities. To do this, we must again solve for $\phi_{j+\frac{1}{2}}$ while taking in multiple cases. The solution procedure for this is outlined below.

Case 1: $((u_{j+1} - u_{j+2} - u_{j-1} + u_j) = 0$

Then as stated previously, there is no flux at the cell interface, and a choice of $\phi_{j+\frac{1}{2}} = 1$ is made.

Case 2: $(u_{j+1} - u_{j+2} - u_{j-1} + u_j) > 0$

This results in a change in the two criteria for given by Eq. 79, and Eq. 80 respectively as

$$\frac{-12u_{j+1} - 6u_j}{(u_{j+1} - u_{j+2} - u_{j-1} + u_j)} \leq \phi_{j+\frac{1}{2}} \leq \frac{-12u_j - 6u_{j+1}}{(u_{j+1} - u_{j+2} - u_{j-1} + u_j)}. \tag{81}$$

– **Sub-case 2.1** $(-12u_j - 6u_{j+1} < 0)$

Then the right hand side of Eq. 81 is negative. Since we know that $\phi_{j+\frac{1}{2}}$ must be greater than zero, a choice to set $\phi_{j+\frac{1}{2}}$ equal to zero is made here, $\phi_{j+\frac{1}{2}} = 0$.

– **Sub-case 2.2** $(-12u_j - 6u_{j+1} > 0)$

if the RHS of Eq. 81 is less than two, then $\phi_{j+\frac{1}{2}} = \frac{-12u_j - 6u_{j+1}}{(u_{j+1} - u_{j+2} - u_{j-1} + u_j)}$. If the RHS of Eq. 81 is greater than two, then $\phi_{j+\frac{1}{2}} = 2$.

Case 3: $(u_{j+1} - u_{j+2} - u_{j-1} + u_j) < 0$

Then the inequalities in Eq. 81 change direction, and we are left with

$$\frac{-12u_{j+1} - 6u_j}{(u_{j+1} - u_{j+2} - u_{j-1} + u_j)} \geq \phi_{j+\frac{1}{2}} \geq \frac{-12u_j - 6u_{j+1}}{(u_{j+1} - u_{j+2} - u_{j-1} + u_j)}. \tag{82}$$

– **Sub-case 3.1** $(-12u_{j+1} - 6u_j > 0)$

Then the left hand side of Eq. 82 is negative. Again, we know that $\phi_{j+\frac{1}{2}}$ must be greater than zero, so a choice to set $\phi_{j+\frac{1}{2}}$ equal to zero is made here, $\phi_{j+\frac{1}{2}} = 0$.

– **Sub-case 3.2** $(-12u_{j+1} - 6u_j < 0)$

Then if the LHS of Eq. 82 is less than two, then $\phi_{j+\frac{1}{2}} = \frac{-12u_{j+1}-6u_j}{(u_{j+1}-u_{j+2}-u_{j-1}+u_j)}$. If the LHS of Eq. 82 is greater than two, then $\phi_{j+\frac{1}{2}} = 2$.

In order to guarantee nonnegative entropy generation, we must also check the constraints that are dependent on the averaged values of entropy generation. The following results are obtained from utilizing Maxima. For the constraint $(P_s^+)_j \geq -\overline{(P_s^-)}_j$, we are left with

$$\left[-(\phi_{j+\frac{1}{2}}u_{j+1} + 6u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j - 6u_j)^2 \right.$$

$$\left. \frac{(\phi_{j+\frac{1}{2}}u_{j+1} + 6u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j + 12u_j)}{2592} \right] \tag{83}$$

$$- \left[\frac{((u_{j+1} - 7u_{j-1} + u_{j-2} - 13u_j)(u_{j+1} - 7u_{j-1} + u_{j-2} + 5u_j)^2)}{2592} \right] \geq 0,$$

with the roots to this equation being

$$\phi^1_{j+\frac{1}{2}} = \left[-(\sqrt{3}\Big((-u^2_{j+1} + (14u_{j-1} - 2u_{j-2} + 2u_j)u_{j+1} - 49u^2_{j-1} \right.$$

$$\left. +(14u_{j-2} - 14u_j)u_{j-1} - u^2_{j-2} + 2u_ju_{j-2} + 143u^2_j)\Big)^{\frac{1}{2}} + 11u_{j+1} + 7u_{j-1} - u_{j-2} + u_j) \right]$$

$$\frac{1}{(2u_{j+1} - 2u_{j+2} - 2u_{j-1} + 2u_j)}, \tag{84}$$

$$\phi^2_{j+\frac{1}{2}} = \left[-(\sqrt{3}\Big((-u^2_{j+1} + (14u_{j-1} - 2u_{j-2} + 2u_j)u_{j+1} - 49u^2_{j-1} \right.$$

$$\left. +(14u_{j-2} - 14u_j)u_{j-1} - u^2_{j-2} + 2u_ju_{j-2} + 143u^2_j)\Big)^{\frac{1}{2}} - 11u_{j+1} - 7u_{j-1} + u_{j-2} - u_j) \right]$$

$$\frac{1}{(2u_{j+1} - 2u_{j+2} - 2u_{j-1} + 2u_j)}, \tag{85}$$

$$\phi^3_{j+\frac{1}{2}} = \frac{-7u_{j+1} + 7u_{j-1} - u_{j-2} + u_j}{u_{j+1} - u_{j+2} - u_{j-1} + u_j}. \tag{86}$$

These roots which ultimately determine the stability region depend upon cell values for u_{j-2},u_{j-1},u_j,u_{j+1}, and u_{j+2}. The constraint $(P_s^-)_{j+1} \geq -\overline{(P_s^+)}_{j+1}$ also depends on the additional cell value, u_{j+3} as illustrated below. This ultimately means that $\phi_{j+\frac{1}{2}}$ depends on six separate cell values. The inequality that must be solved for $(P_s^-)_{j+1} \geq -\overline{(P_s^+)}_{j+1}$ is given as

$$\Bigg[(\phi_{j+\frac{1}{2}}u_{j+1} - 6u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j + 6u_j)^2$$

$$\frac{(\phi_{j+\frac{1}{2}}u_{j+1} + 12u_{j+1} - \phi_{j+\frac{1}{2}}u_{j+2} - \phi_{j+\frac{1}{2}}u_{j-1} + \phi_{j+\frac{1}{2}}u_j + 6u_j)}{2592}\Bigg] \tag{87}$$

$$- \left[\frac{((5u_{j+1} + u_{j+3} - 7u_{j+2} + u_j)^2(13u_{j+1} - u_{j+3} + 7u_{j+2} - u_j))}{2592}\right] \geq 0,$$

where the roots for this inequality are

$$\phi^1_{j+\frac{1}{2}} = \Bigg[(\sqrt{3}\Big(143u_{j+1}^2 + (2u_{j+3} - 14u_{j+2} + 2u_j)u_{j+1} - u_{j+3}^2$$

$$+(14u_{j+2} - 2u_j)u_{j+3} - 49u_{j+2}^2 + 14u_ju_{j+2} - u_j^2\Big)^{\frac{1}{2}} + u_{j+1} - u_{j+3} + 7u_{j+2} + 11u_j)\Bigg]$$

$$\frac{1}{(2u_{j+1} - 2u_{j+2} - 2u_{j-1} + 2u_j)}, \tag{88}$$

$$\phi^2_{j+\frac{1}{2}} = \Bigg[(\sqrt{3}\Big(143u_{j+1}^2 + (2u_{j+3} - 14u_{j+2} + 2u_j)u_{j+1} - u_{j+3}^2$$

$$+(14u_{j+2} - 2u_j)u_{j+3} - 49u_{j+2}^2 + 14u_ju_{j+2} - u_j^2\Big)^{\frac{1}{2}} - u_{j+1} + u_{j+3} - 7u_{j+2} - 11u_j)\Bigg]$$

$$\frac{1}{(2u_{j+1} - 2u_{j+2} - 2u_{j-1} + 2u_j)}, \tag{89}$$

$$\phi^3_{j+\frac{1}{2}} = \frac{(-u_{j+1} + u_{j+3} - 7u_{j+2} + 7u_j)}{(u_{j+1} - u_{j+2} - u_{j-1} + u_j)}. \tag{90}$$

These are the critical values for each cubic function that define the entropy stability criteria for the fourth-order method. These regions, just as in the second-order scheme, define the values for $\phi_{j+\frac{1}{2}}$ that provide positive entropy generation for each cell while trying to remain fourth-order accurate as much as possible. It is clear to see that solving for these regions algebraically is impractical, if not impossible. The general solution for the stability regions will be demonstrated graphically for two situations as an example. There is one value of $\phi_{j+\frac{1}{2}}$ that should satisfy both of these inequalities because they both correspond to the same face value. However, the existence of no solution is a possibility, and in this case a choice must be made for the value of $\phi_{j+\frac{1}{2}}$. Both of these scenarios are shown below. The cell values that correspond to an intersection between these two curves that falls in the stability region are $u_{j-1} = 0.25$, $u_j = 0.5$, $u_{j+1} = 0.7$, and $u_{j+2} = 1.95$, and shown in Fig. 3.2. The cell values that correspond to an intersection between the two curves that does not fall in the stability region are $u_{j-1} = 0.5$, $u_j = 1.5$, $u_{j+1} = -2.5$, and $u_{j+2} = 1.95$, and is shown below in Fig. 3.3

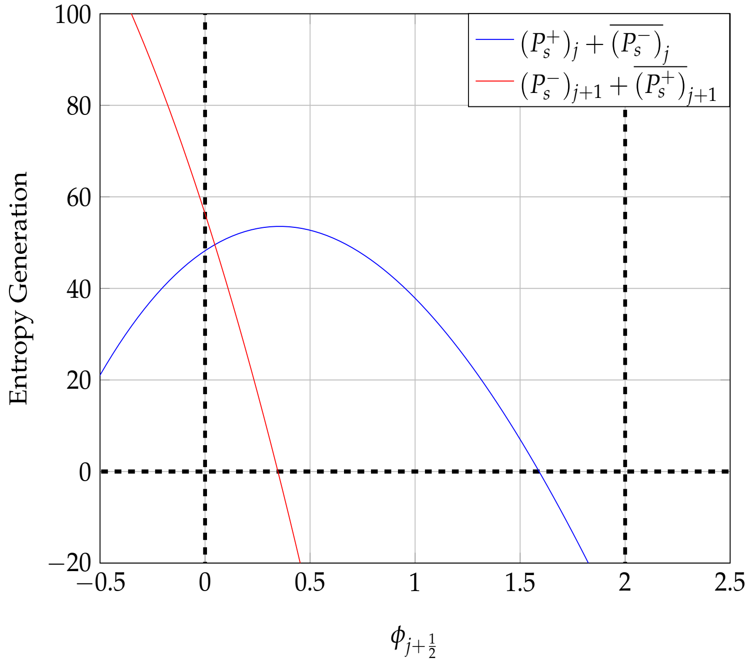

FIGURE 3.2. Fourth-order stability plot for a solution for $\phi_{j+\frac{1}{2}}$

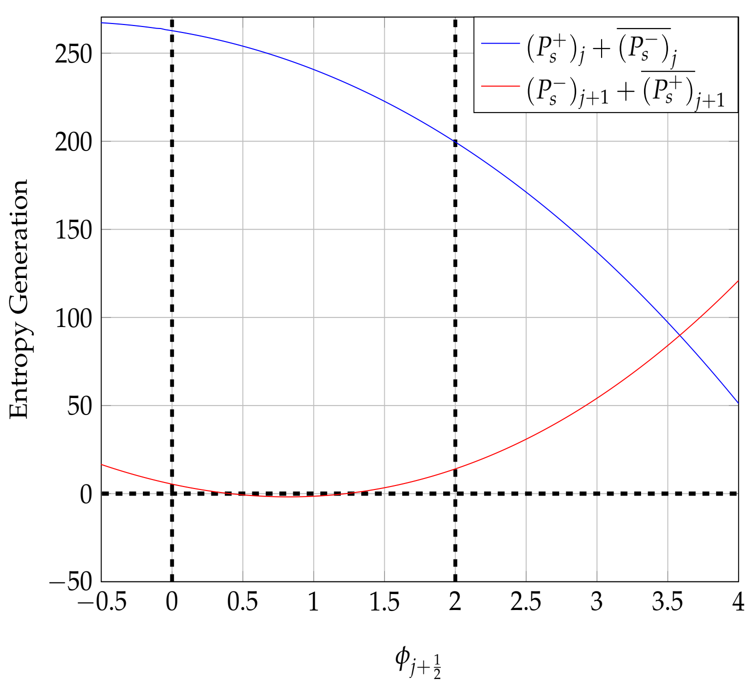

FIGURE 3.3. Fourth-order stability plot for no solution for $\phi_{j+\frac{1}{2}}$

3.2. Issues with the Stencil

While the stencil described in the previous section recovers fourth-order accuracy in the neighborhood of $\phi_{j+\frac{1}{2}}$, problems arise when the value for $\phi_{j+\frac{1}{2}}$ is at the extreme, zero or two. To illustrate this, we simply substitute zero, a typical value for $\phi_{j+\frac{1}{2}}$ around the shock, in for the face interpolation $u_{j+\frac{1}{2}}$ we are left with

$$u_{j+\frac{1}{2}} = \frac{1}{2}(u_j + u_{j+1}). \tag{91}$$

In this case of extrema this interpolation reverts to second-order. This interpolation on the face is problematic because a simple linear average in the region of a shock is not a stable scheme. Since there will be unphysical solution values around the shock, it is reasonable to assume that these values could potentially result in negative entropy production values, which is exactly what we observe. This linear approximation is sufficient to introduce oscillations, and thus violate the second law of thermodynamics.

When the second-order face interpolation is used, as is chapter 2, when $\phi_{j+\frac{1}{2}}$ is equal to zero or two, the scheme reverts to first-order depending on the flow direction, guaranteeing unconditional stability and positive entropy generation. For the fourth-order spatial discretization, we ideally want a face interpolation scheme that recovers a symmetric stencil based on flow direction when $\phi_{j+\frac{1}{2}}$ is equal to zero and two respectively. This however, is not the case with the fourth-order scheme. It was just shown that in the case of $\phi_{j+\frac{1}{2}} = 0$ that the scheme reverts to a second-order linear average. In the case of $\phi_{j+\frac{1}{2}} = 2$, this scheme becomes

$$u_{j+\frac{1}{2}} = \frac{-(u_{j+2} - 4_{j+1} + u_{j-1} - 4u_j)}{6}, \tag{92}$$

which does not revert to second-order. In order to guarantee positive entropy generation we need a scheme that reverts to first-order in the discontinuous region. It is safe to assume that this is the reason why Merriam's second-order accurate approximate limiter, while providing a nice solution with minor spurious oscillations, does not preserve entropy throughout the simulation. A truly entropy stable scheme will produce nonnegative entropy throughout the domain for the duration of the simulation.

Alternatively, in a perfect limiting scheme, there exists a face extrapolation that reverts to first-order in the case of shocks, while remaining fourth-order throughout the smooth regions of the flow. If there exists such a scheme, it should be possible to construct it based on a blending scheme between the first order method that was demonstrated in the work done by Merriam, and the fourth-order scheme given in this work.

CONJECTURE:

There exists a scheme which reverts to first order accurate in the case of shocks, while remaining fourth-order accurate in the smooth regions

GIVEN:

$$u_{j+\frac{1}{2}} = u_j + \frac{1}{2}\phi_{j+\frac{1}{2}}(u_{j+1} - u_j), \tag{93}$$

$$u_{j+\frac{1}{2}} = \frac{1}{2}(u_j + u_{j+1}) - \frac{1}{12}\phi_{j+\frac{1}{2}}(u_{j-1} - u_j - u_{j+1} + u_{j+2}). \tag{94}$$

There exists a blending scheme which recovers first-order in the case of shocks, and fourth-order in the smooth region that can be written in the form

$$u_{j+\frac{1}{2}} = u_j + \frac{1}{2}\phi_{j+\frac{1}{2}}(u_{j+1} - u_j) + \frac{1}{12}\phi_{j+\frac{1}{2}}\Big((u_{j+1} - u_j) - (u_{j-1} - u_{j+2})\Big). \tag{95}$$

It can clearly be seen that when $\phi_{j+\frac{1}{2}}$ is equal to zero, this scheme recovers first-order, and reverts to $u_{j+\frac{1}{2}} = u_j$. However, we also want a scheme that is fourth-order throughout most of the domain. In this case, when $\phi_{j+\frac{1}{2}} = 1$, Eq. 95 is

$$u_{j+\frac{1}{2}} = \frac{(u_{j+2} + u_{j+1} - u_{j-1} + 11u_j)}{12}, \tag{96}$$

which is not the same as the fourth-order average in given by Eq. 67.

CONCLUSION:

It is not possible to construct a face interpolation scheme for $u_{j+\frac{1}{2}}$ that is fourth-order in the smooth region that reverts for first-order in the case of shocks unless an additional limiting factor is applied on top of the $\phi_{j+\frac{1}{2}}$. This idea is consistent with other fourth-order limiters [21], in that a fourth-order finite-volume limiter will not recover a first-order scheme in discontinuous regions. The work [21] states that "fixed stencil interpolation of second or higher order accuracy is necessarily oscillatory near a discontinuity." In other words, regardless of how carefully a high-order face interoplant is constructed, it must necessarily reduce to first-order accurate around shocks. However, in the current fourth-order face interpolation devised as Eq. 94 does not degenerate to the first order with the choice of the range of $\phi_{j+\frac{1}{2}}$ values, because this reduces to second-order around shocks. There must be a further reduction in the order of accuracy around a shock in order for the solution to a discontinuous problem to be stable.

The question now is, is it possible to construct a hybrid scheme that uses the divergence of velocity as a sensor for shocks? For example, if the divergence of velocity, $\nabla \cdot u_j$, is less than zero (indicative of a shock) then the face interpolation

$$u_{j+\frac{1}{2}} = u_j + \frac{1}{2}\phi_{j+\frac{1}{2}}(u_{j+1} - u_j), \tag{97}$$

is used. If the divergence of the velocity, $\nabla \cdot u_j$ is greater than or equal to zero, then the fourth-order face interpolation is used.

$$u_{j+\frac{1}{2}} = \frac{1}{2}(u_j + u_{j+1}) - \frac{1}{12}\phi_{j+\frac{1}{2}}(u_{j-1} + u_j + u_{j+1} - u_{j+2}) \tag{98}$$

Results for the fourth-order accurate scheme dictated by the face interpolation given by Eq. 94 are given in the next chapter, as well as the results for the hybrid face interpolation given by Eq. 97 and 98.

CHAPTER 4

Fourth-Order Results and Discussion

This chapter exercises the entropy stable method in the fourth-order discretization as discussed in the previous chapter with application to Burgers' equation. Again, for consistency, the initial condition is a two-period sin wave with unit amplitude. This problem contains sonic points that develop into shocks. The domain length is $d = 4\pi$ and the solution is run to a time of $t = 1.0s$. Fig. 4.1 below shows the solution run to a time $t = 1.0s$ without the application of the entropy limiter. This is clearly more oscillatory than the previous methods which is to be expected from a high-order scheme. The oscillations occur around the shock and propagate further, both upstream and downstream of the shock, than the lower-order methods because of the larger stencil.

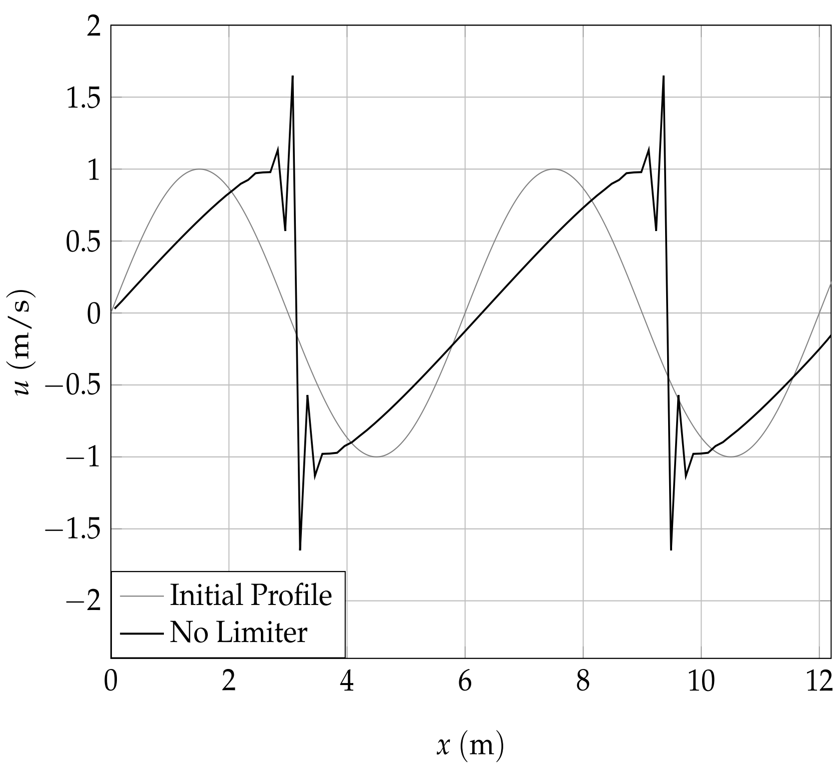

Figure 4.1. Fourth-order solution results without the use of a limiter

Introducing the limiter should mitigate the spurious oscillations that occur at the discontinuity. We expect the amplitude of the shock to decrease, even though we know that this scheme is not truly first-order around the shock. This is exactly what we observe, and this phenomenon is shown below.

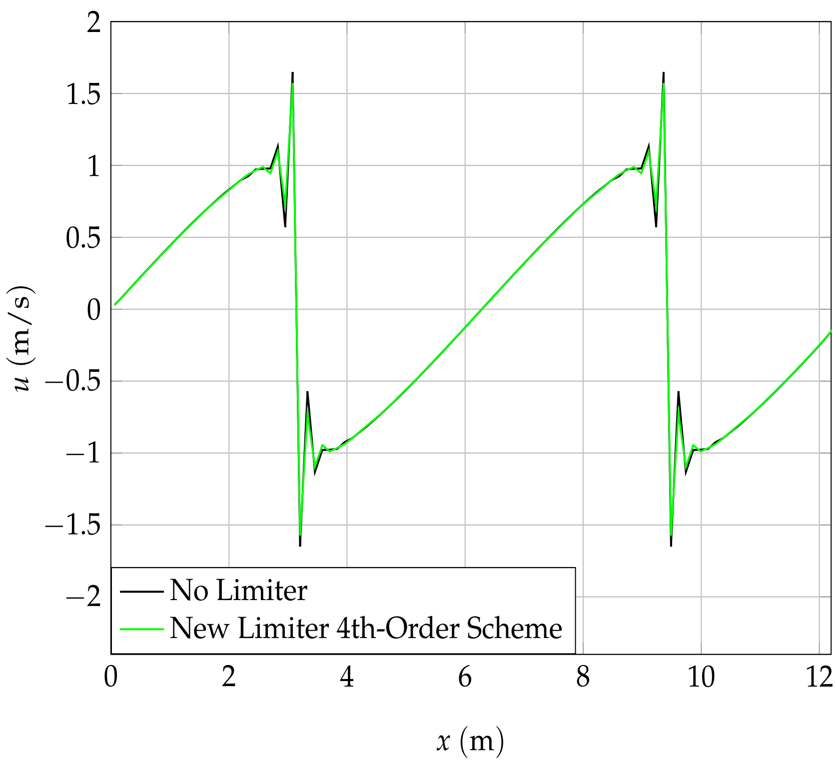

FIGURE 4.2. Fourth-order discretization limiting scheme solution results compared to results obtained without the use of a limiter

Fig. 4.3 shows an enlarged version of the region of the solution where the shock forms (from $x = 2$ to $x = 4$, and $u = -1.7$ to $u = 1.7$) as to better illustrate the decrease in amplitude of the oscillations around the shock.

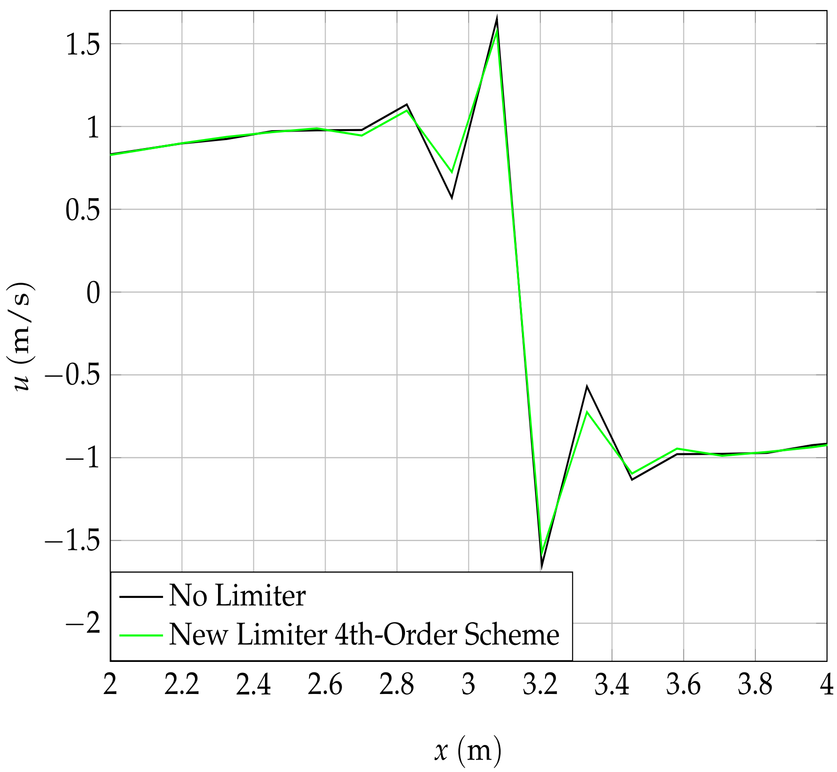

FIGURE 4.3. Enlarged results

It is also necessary to look at the entropy generation throughout the duration of the solution. The results for entropy generation over time for the limited solution and the solution without a limiter are plotted below in Fig. 4.5

The entropy generation over time has an initially positive trend, however it quickly moves into the negative region. Additionally, Fig. 4.5 shows the entropy generated in the domain. There is negative entropy production around the shock locations, and this method violates the second law of thermodynamics.

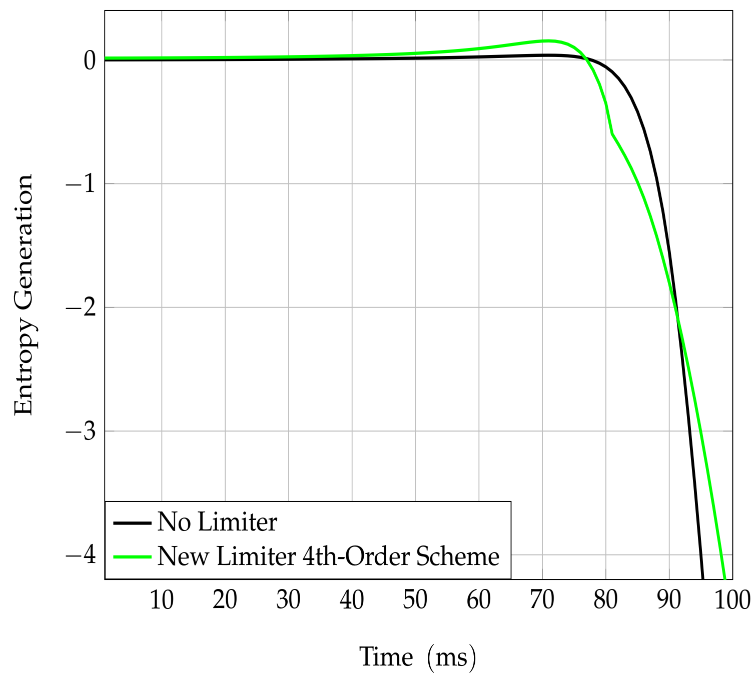

FIGURE 4.4. Entropy generation over time for the the new limiter in fourth-order discretization compared to the entropy generation without a limiter

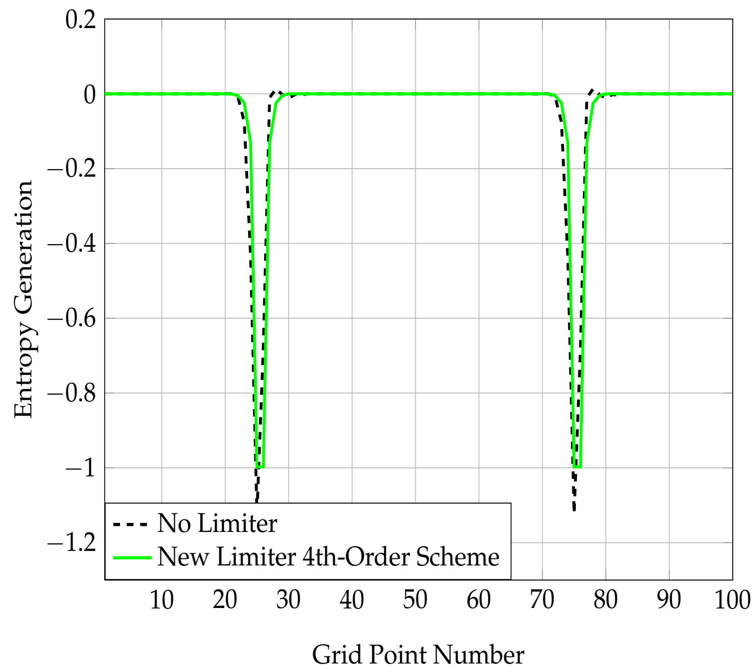

FIGURE 4.5. Entropy generation comparison throughout the domain at the final time $t = 1.0s$

To understand this, it is necessary to look at the values for $\phi_{j+\frac{1}{2}}$ that are associated with this solution. Fig. 4.6 compares the solution with the values for $\phi_{j+\frac{1}{2}}$ at the final time step. As mentioned before, when a single value e for $\phi_{j+\frac{1}{2}}$ does not exist that simultaneously satisfies any of the combinations of the cell entropy constraints given again by Eq. 39 and Eq. 41 and restated here as

$$(P_s^+)_j \geq \min\Big(-\overline{(P_s^-)}_j, 0\Big),$$
$$(P_s^-)_{j+1} \geq \min\Big(-\overline{(P_s^+)}_{j+1}, 0\Big),$$

then a choice must be made for the value of $\phi_{j+\frac{1}{2}}$. If the code is left to choose values that make the solution most stable, then these choices will tend towards the extreme values of $\phi_{j+\frac{1}{2}} = 0$ and $\phi_{j+\frac{1}{2}} = 2$. Consequently, because these values for $\phi_{j+\frac{1}{2}}$ tend towards zero, the

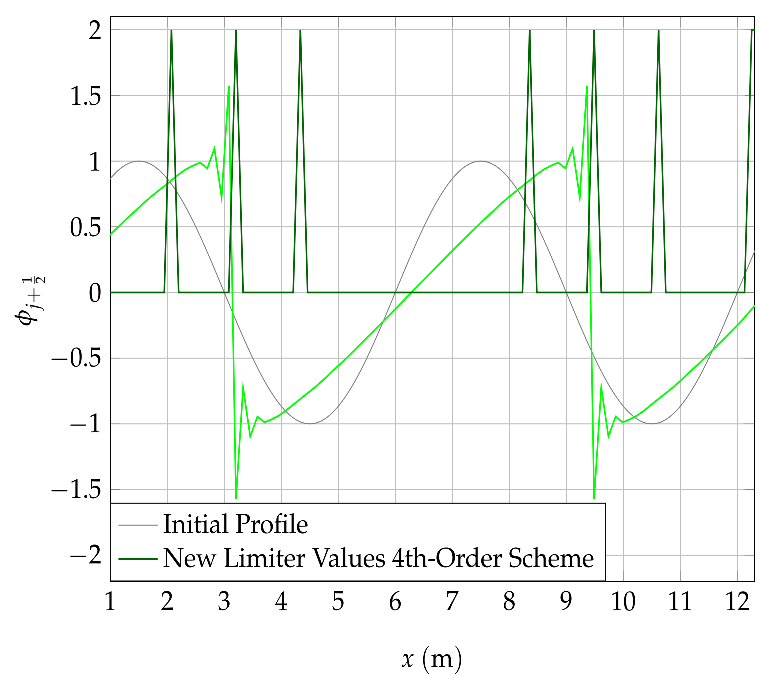

FIGURE 4.6. Values for $\phi_{j+\frac{1}{2}}$ from the new limiting scheme

scheme reverts to a second-order linear average throughout most of the domain. This fourth-order face extrapolation violates the second law of thermodynamics. Next, the results of the hybrid scheme discussed in the previous chapter will be given. This scheme uses a divergence test to dynamically change the order of accuracy around the shock. While some oscillations are still present in the solution, it still offers some insight towards the development of stable methods down the road.

The solution for the fourth-order hybrid scheme is given below in Fig. 4.7. Oscillations are still present around the shock, however the magnitude of these spurious oscillations have been greatly reduced.

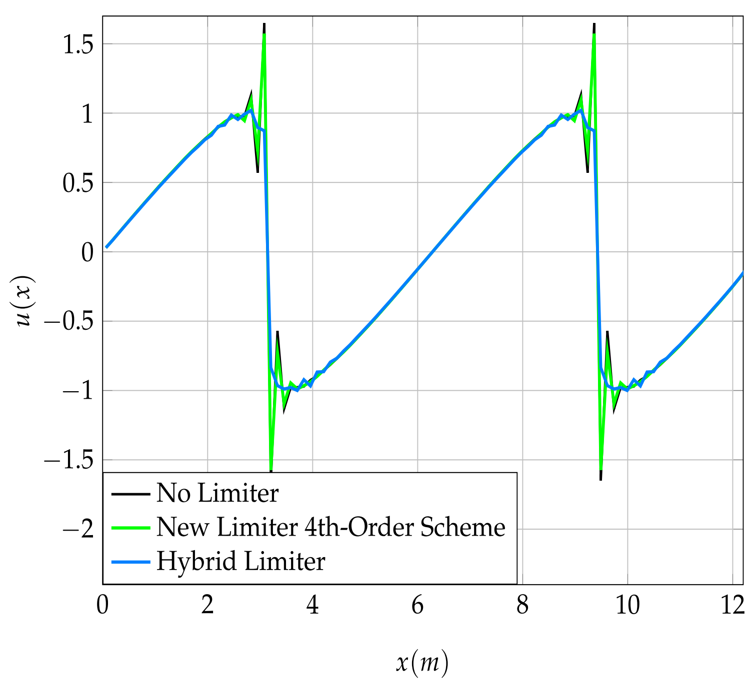

FIGURE 4.7. Solution of hybrid fourth-order limiting scheme

In Fig. 4.8, the entropy generation for each discrete cell in the domain is plotted. The hybrid scheme shown in light blue and produces positive entropy values in almost all areas around the discontinuity.

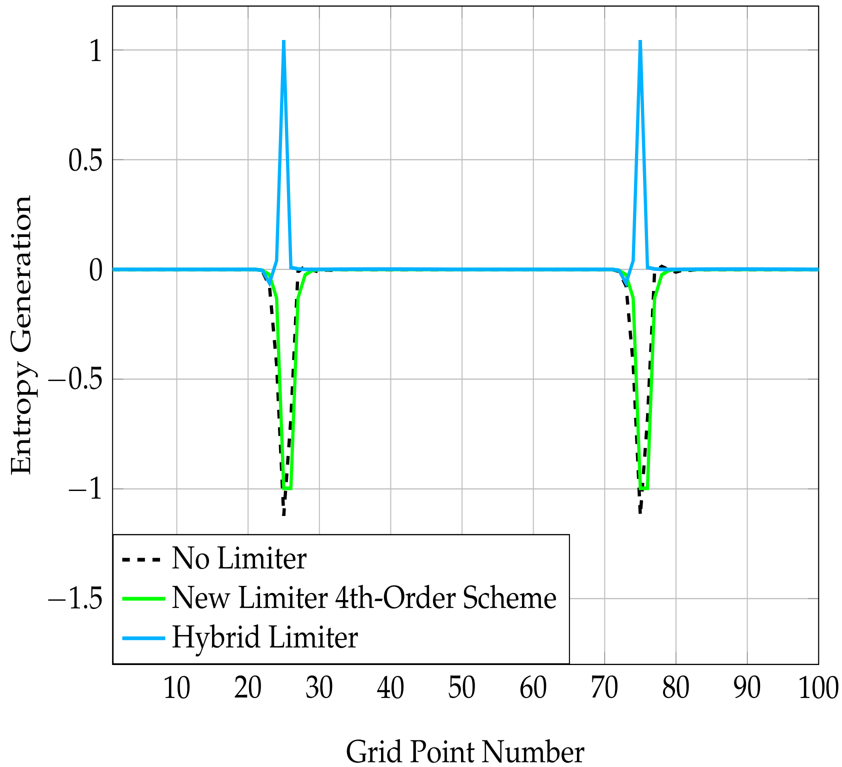

FIGURE 4.8. Comparison of entropy generation throughout the domain of unlimited, limited, and hybrid solutions

Additionally, this scheme ensures that entropy increases as time increases. Fig 4.9 shows the entropy generation over time for the duration of the solution. For a high-order scheme strictly satisfying the coupling of the entropy constraints is not enough to ensure that entropy is generated positively. It is suspected that either the constraints need to be revised, or more likely that situations occur where a solution cannot be found to simultaneously satisfy these constraints. The addition of a dynamic face interpolation scheme is necessary in order to guarantee that the scheme drops to a mostly first-order accurate scheme in the neighborhood of shocks.

It is again illuminating to compare the solution to the locations where entropy is produced. Overlaying these plots shows explicitly that the entropy generated in the domain is zero in

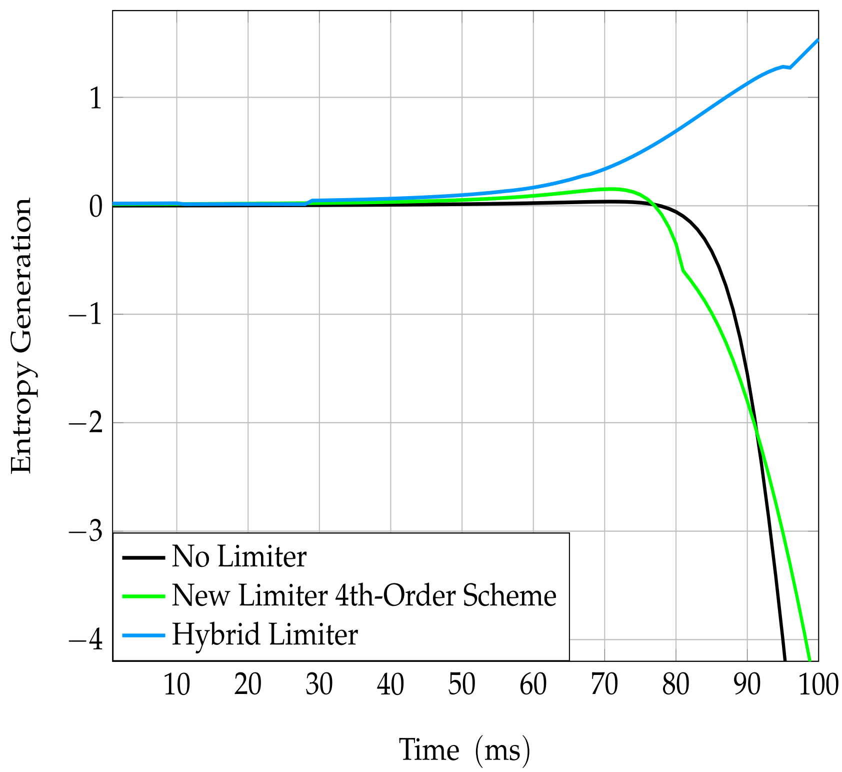

FIGURE 4.9. Entropy generation over time of new limiting methods

the smooth regions, and positive in the discontinuous regions. Fig. 4.10 demonstrates this below.

Just as was done for the previous methods, the values of $\phi_{j+\frac{1}{2}}$ that satisfy the cell entropy inequalities are plotted along with the solution in Fig. 4.11. In this case, because the values for $\phi_{j+\frac{1}{2}}$ oscillate between zero and two, the scheme reverts to mostly second-order accurate when $\phi_{j+\frac{1}{2}} = 0$, weighted fourth-order accurate when $\phi_{j+\frac{1}{2}} = 2$, and first-order accurate around the shock when $\nabla \cdot u < 0$.

This hybrid methodology for a fourth-order face extrapolation provides some improvements in terms of suppressing oscillations and generating entropy. However, more research is required to arrive at a viable solution. Conclusions and future research directions follow.

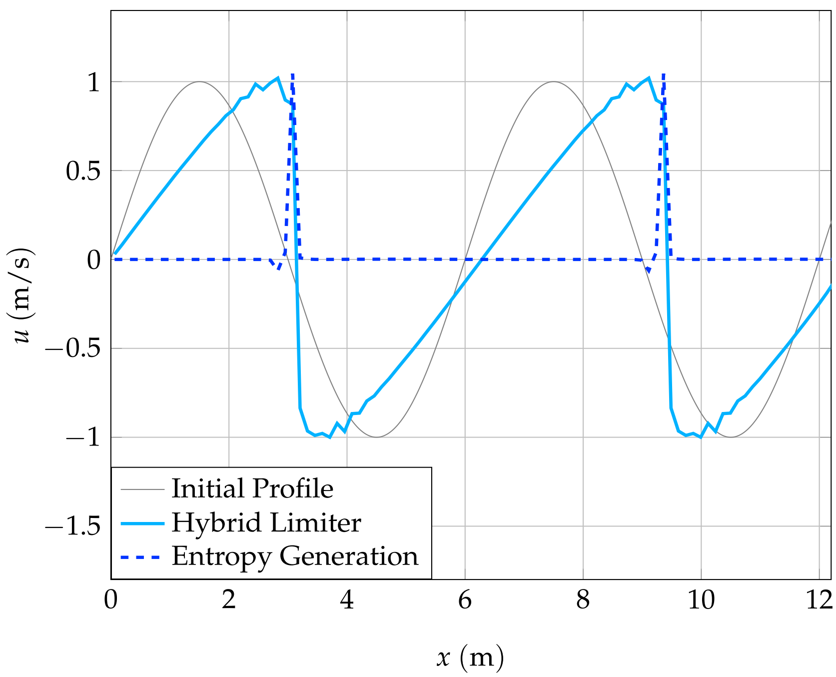

FIGURE 4.10. Solution of hybrid limiting scheme plotted with entropy production

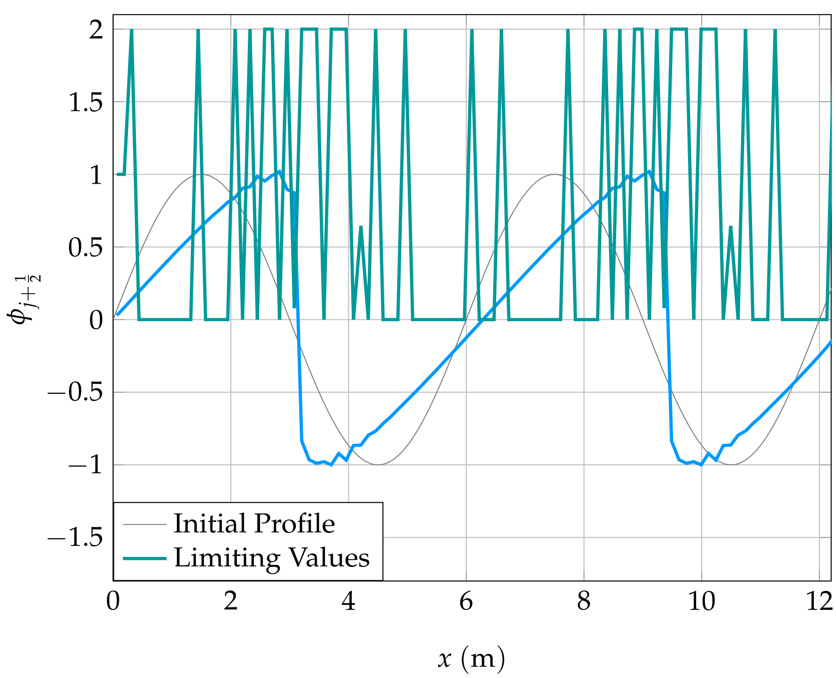

FIGURE 4.11. $\phi_{j+\frac{1}{2}}$ values for the hybrid limiting scheme with a fourth-order discretization

CHAPTER 5

Conclusions and Future Work

The present study examined the entropy concept outlined by Merriam [1] in detail and the methods were tested using the Burgers' equation. The results from these tests were analyzed thoroughly. Conclusions and directions for future work are summarized hereafter.

5.1. Conclusion

Satisfying a cell entropy inequality that ensures that the second law of thermodynamics in the solution of flows with shocks appeared to be an appealing limiting methodology. However, this research demonstrates that in satisfying the cell entropy inequality for high-order methods, the scheme reverts to a majority low-order scheme to stay entropy stable. This indicates that it is not straightforward to obtain the entropy stability criteria for a high-order scheme. Additionally, even for Burgers' equation, the parameters surrounding the implementation of the methodology proposed in this research can quickly make satisfaction of the entropy inequality for high-order methods impractical. The logical statements associated with each entropy inequality grow uncontrollably as the order of accuracy increases. In the present work, this amounts to computing the intersection between two curves which is significantly more expensive than the methodology proposed by Merriam, which makes linear approximations. Because of this, the implementation of this limiting method has been deemed impractical. Consequently however, this research has shown that simply satisfying each cell entropy constraint proposed by Merriam is not robust enough to guarantee positive entropy production in the extension to high-order spatial discretization schemes.

Application of this methodology to Burgers' equation has provided insight into the principles that dictate entropy stability for high-order entropy stable FVM. Both methods were thoroughly analyzed, but unfortunately deemed as nonviable for truly suppressing oscillations near shocks while preserving the underlying numerical scheme accuracy except for a first-order accurate method. It is believed through this study that the entropy concept is unlikely applicable to a system of nonlinear PDEs.

5.2. Future Work

A potential solution to the issue of the methodology being overly dissipative is the computation of an averaged entropy value in each cell that could potentially generate small amounts of entropy in smooth regions, but less entropy globally. If the entropy generated in each half of a cell (for example, $(P_s^+)_j$) is positive, but a value for $\phi_{j+\frac{1}{2}}$ does not exist that satisfies the intersection between $(P_s^+)_j$ and $(P_s^-)_{j+1}$, then the two values of $\phi_{j+\frac{1}{2}}$ that provide maximum entropy generation for each half of the cell are averaged, and one face value is selected. This would be guaranteed to generate more entropy than the minimum amount present in the domain, but less than the maximum amount.

It is worth mentioning that in the work done by Merriam, the Euler equations were used to demonstrate the entropy stability method. Effort was made to reproduce the reference work in order to develop an entropy stable method for the Euler equations, however, the results were unable to be reproduced.

BIBLIOGRAPHY

[1] M. L. Merriam, "An entropy-based approach to nonlinear stability," tech. rep., NASA, March 1989.

[2] E. M. Murman and J. D. Cole, "Calculation of plane steady transonic flows," *AIAA Journal*, vol. 9, no. 19, p. 5, 1971.

[3] J. Glimm and P. D. Lax, *Decay of Solutions of Systems of Nonlinear Hyperbolic Conservation Laws*, ch. The Construction of Exact Solutions as a Limit of Approximate Solutions. Mem. Amer. Math. Soc. V101, 1970.

[4] O. A. Oleinik, "Discontinuous solutions of non-linear differential equations," tech. rep., American Math. Soc. Transl. V26, 1957.

[5] T. C. Fisher and M. H. Carpenter, "High-order entropy stable finite-difference schemes for nonlinear conservation laws: Finite domains," tech. rep., NASA, February 2013.

[6] P. Lax, "Shock waves and entropy," E.H Zarantonello, ed 603-634, University of Wisconsin, 1971.

[7] F. Bouchut, "Nonlinear stability of finite-volume methods for hyperbolic conservation laws," in *And Well-Balanced Schemes for Sources*, no. 76M12, Department of Mathematics and Applications CNRS, 2000.

[8] F. Ismail and P. L. Roe, "A parallel adaptive numerical method with generalized curvilinear coordinate transformation for compressible Navier-Stokes equations," *Int. J. Numer. Meth. Fluids*, vol. 82, pp. 664–688, 2016.

[9] S. Pirozzoli, "Conservative hybrid compact-weno schemes for shock-turbulence interaction," *Journal of Compuational Physics*, vol. 178, no. 1, pp. 81–117, 2002.

[10] D. P. D.J. Hill, "Hybrid tuned center difference-weno method for large eddy simulations in the presence of strong shocks," *Journal of Computational Physics*, vol. 194, no. 1, pp. 435–450, 2004.

[11] S. L. S. Kawai, "Localized artificial diffusivity schemes for discontinuity capturing on a curviliear mesh," *Journal of Computational Physics*, vol. 227, no. 1, pp. 9498–9526, 2008.

[12] C.-W. Shu, "High order weighted essentially nonoscillatory schemes for convection dominated problems," *SIAM Review*, vol. 51, no. 1, pp. 82–126, 2009.

[13] E. Tadmor, "The numerical viscosity of entropy stable schemes for systems of conservation laws," *Mathematics of Computation*, vol. 49, no. 1, pp. 91–103, 1987.

[14] E. Tadmor, "Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems," *Acta Numerica*, vol. 12, no. 1, pp. 451–512, 2003.

[15] A.Jameson, "The construction of discretely conservative finite-volume schemes that also globally conserve energy or entropy," *Journal of Scientific Computing*, vol. 34, no. 2, pp. 152–187, 2008.

[16] C. R. P.G. LeFloch, J.M. Mercier, "Fully discrete, entropy conservative schemes of arbitrary order," *SIAM Journal of Numerical Analysis*, vol. 40, no. 1, pp. 1968–1992, 2002.

[17] E. T. U.S. Fjordholm, S. Mishra, "Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws," *SIAM Journal of Numerical Analysis*, vol. 50, no. 1, pp. 544–573, 2012.

[18] Maxima, "Maxima, a computer algebra system. version 5.34.1," 2014.

[19] A. Harten, "On the symmetric form of systems of conservation laws with entropy," *J. Comp. Phy.*, vol. 49(1), pp. 151–164, 1983.

[20] P. Wesseling, *Principles of Computational Fluid Dynamics*, ch. Finite-Volume and Finite-Difference Discretization on Nonnuniform grids. Springer, 2001.

[21] C.-W. Shu, B. Cockburn, C. Johnson, and E. Tadmor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, ch. Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws. Springer, 1997.

APPENDIX A

Second-Order Maxima Code

–> kill(all)$ ujval:ucell; ujpval:ujpcell;

$$ucell \tag{ujval}$$

$$ujpcell \tag{ujpval}$$

–> uhalf(uj,ujp,phi):=uj + phi/2*(ujp-uj);

$$\mathrm{uhalf}\left(uj, ujp, \phi\right) := uj + \frac{\phi}{2}\left(ujp - uj\right) \tag{\%o3}$$

–> S:-uj^2;

$$-uj^2 \tag{S}$$

–> Sjp:-ujp^2;

$$-ujp^2 \tag{Sjp}$$

–> Stemp1:diff(-ujv^2,ujv,1);

$$-2ujv \tag{Stemp1}$$

-> Sprime(uj)::=subst(uj,ujv,Stemp1);

$$\text{Sprime}\,(uj)::=\text{subst}\,(uj,ujv,\mathit{Stemp1}) \tag{\%o7}$$

-> Stemp2:diff(-ujpv^2,ujpv,1);

$$-2ujpv \tag{Stemp2}$$

-> Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

-> fj(uj):=(uj^2)/2;

$$\text{fj}\,(uj):=\frac{uj^2}{2} \tag{\%o10}$$

-> fjhalf(uj,ujp,phi):=(uhalf(uj,ujp,phi)^2/2);

$$\text{fjhalf}\,(uj,ujp,\phi):=\frac{\text{uhalf}\,(uj,ujp,\phi)^2}{2} \tag{\%o11}$$

-> fjp(ujp):=ujp^2/2;

$$\text{fjp}\,(ujp):=\frac{ujp^2}{2} \tag{\%o12}$$

-> Fj(uj):=(-2/3)*uj^3;

$$\text{Fj}\,(uj):=\frac{-2}{3}uj^3 \tag{\%o13}$$

-> Fjhalf(uj,ujp,phi):=-(2*uhalf(uj,ujp,phi)^3)/3;

$$\mathrm{Fjhalf}\left(uj, ujp, \phi\right) := \frac{-2\mathrm{uhalf}\left(uj, ujp, \phi\right)^{3}}{3} \tag{\%o14}$$

-> Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3}ujp^{3} \tag{\%o15}$$

-> Pjplus(uj,ujp,phi):=-Sprime(uj)* (fjhalf(uj,ujp,phi) - fj(uj))

+ (Fjhalf(uj,ujp,phi) - Fj(uj))$

-> Pjplus(ujval,ujpval,phival);

$$-\frac{2\left(\frac{phival\,(ujpcell-ucell)}{2}+ucell\right)^{3}}{3}+2ucell\left(\frac{\left(\frac{phival\,(ujpcell-ucell)}{2}+ucell\right)^{2}}{2}-\frac{ucell^{2}}{2}\right)+\frac{2ucell^{3}}{3} \tag{\%o17}$$

-> ratsimp(Pjplus(ujval,ujpval,phival));

$$-\Big(phival^{3}\,ujpcell^{3}+\left(3phival^{2}-3phival^{3}\right)\,ucell\,ujpcell^{2} + \left(3phival^{3}-6phival^{2}\right)\,ucell^{2}\,ujpcell+\left(3phival^{2}-phival^{3}\right)\,ucell^{3}\Big)/12 \tag{\%o18}$$

-> factor(%);

$$-\frac{phival^{2}\left(ujpcell-ucell\right)^{2}\left(phival\,ujpcell-phival\,ucell+3ucell\right)}{12} \tag{\%o19}$$

--> solve([Pjplus(ujval,ujpval,phival)],[phival]);

$$[phival = -\frac{3ucell}{ujpcell - ucell}, phival = 0] \quad (\%o20)$$

--> kill(all)$ ujval:ucell; ujmval:ujmcell;

$$ucell \quad (ujval)$$

$$ujmcell \quad (ujmval)$$

--> uhalf(uj,ujm,phi):=(ujm + phi/2*(uj - ujm));

$$\mathrm{uhalf}\left(uj, ujm, \phi\right) := ujm + \frac{\phi}{2}\left(uj - ujm\right) \quad (\%o3)$$

--> S:-uj^2;

$$-uj^2 \quad (S)$$

--> Sjm:-ujm^2;

$$-ujm^2 \quad (Sjm)$$

--> Stemp1:diff(-ujv^2,ujv,1)$

--> Sprime(uj)::=subst(uj,ujv,Stemp1)$

```
-->  Stemp2:diff(-ujmv^2,ujmv,1)$
-->  Sprimejm(ujm)::=subst(ujm,ujmv,Stemp2)$
-->  fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \quad (\%o10)$$

```
-->  fjhalf(uj,ujm,phi):=(uhalf(uj,ujm,phi)^2/2);
```

$$\mathrm{fjhalf}\left(uj, ujm, \phi\right) := \frac{\mathrm{uhalf}\left(uj, ujm, \phi\right)^2}{2} \quad (\%o11)$$

```
-->  fjm(ujm):=ujm^2/2;
```

$$\mathrm{fjm}\left(ujm\right) := \frac{ujm^2}{2} \quad (\%o12)$$

```
-->  Fj(uj):=(-2/3)*uj^3;
```

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \quad (\%o13)$$

```
-->  Fjhalf(uj,ujm,phi):=-(2*uhalf(uj,ujm,phi)^3)/3;
```

$$\mathrm{Fjhalf}\left(uj, ujm, \phi\right) := \frac{-2\mathrm{uhalf}\left(uj, ujm, \phi\right)^3}{3} \quad (\%o14)$$

```
-->  Fjm(ujm):=(-2/3)*ujm^3;
```

$$\mathrm{Fjm}\left(ujm\right) := \frac{-2}{3} ujm^3 \quad (\%o15)$$

--> Pjminus(uj,ujm,phi):=-Sprime(uj)* (fj(uj) - fjhalf(uj,ujm,phi)

+ (Fj(uj) - Fjhalf(uj,ujm,phi));

--> Pjminus(ujval,ujmval,phival);

$$\frac{2\left(ujmcell+\frac{phival\,(ucell-ujmcell)}{2}\right)^3}{3}+2ucell\left(\frac{ucell^2}{2}-\frac{\left(ujmcell+\frac{phival\,(ucell-ujmcell)}{2}\right)^2}{2}\right) \quad (\%o17)$$

$$-\frac{2ucell^3}{3}$$

--> ratsimp(Pjminus(ujval,ujmval,phival));

$$-\Big(\left(phival^3-6phival^2+12phival-8\right)\,ujmcell^3 \quad (\%o18)$$

$$+\left(-3phival^3+15phival^2-24phival+12\right)\,ucell\;ujmcell^2$$

$$+\left(3phival^3-12phival^2+12phival\right)\,ucell^2\;ujmcell$$

$$+\left(-phival^3+3phival^2-4\right)\,ucell^3\Big)/12$$

--> ex1:factor(%);

$$-\frac{(phival-2)^2\,(ujmcell-ucell)^2\,(phival\;ujmcell-2ujmcell-phival\;ucell-ucell)}{12} \quad (ex1)$$

--> kill(all)$ ujval:ucell; ujpval:ujpcell;

$$ucell \quad (ujval)$$

$$ujpcell \tag{ujpval}$$

-> uhalf(uj,ujp,phi):=uj + phi/2*(ujp-uj);

$$\mathrm{uhalf}\left(uj, ujp, \phi\right) := uj + \frac{\phi}{2}\left(ujp - uj\right) \tag{\%o3}$$

-> S:-uj^2;

$$-uj^2 \tag{S}$$

-> Sjp:-ujp^2;

$$-ujp^2 \tag{Sjp}$$

-> Stemp1:diff(-ujv^2,ujv,1)$

-> Sprime(uj)::=subst(uj,ujv,Stemp1)$

-> Stemp2:diff(-ujpv^2,ujpv,1)$

-> Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

-> fj(uj):=(uj^2)/2;

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o10}$$

-> fjhalf(uj,ujp,phi):=(uhalf(uj,ujp,phi)^2/2);

$$\mathrm{fjhalf}\left(uj, ujp, \phi\right) := \frac{\mathrm{uhalf}\left(uj, ujp, \phi\right)^2}{2} \tag{\%o11}$$

−> fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\left(ujp\right):=\frac{ujp^2}{2} \tag{\%o12}$$

−> Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\left(uj\right):=\frac{-2}{3}uj^3 \tag{\%o13}$$

−> Fjhalf(uj,ujp,phi):=-(2*uhalf(uj,ujp,phi)^3)/3;

$$\mathrm{Fjhalf}\left(uj,ujp,\phi\right):=\frac{-2\mathrm{uhalf}\left(uj,ujp,\phi\right)^3}{3} \tag{\%o14}$$

−> Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right):=\frac{-2}{3}ujp^3 \tag{\%o15}$$

−> Pjpminus(uj,ujp,phi):=-Sprimejp(ujp)* (fjp(ujp) - fjhalf(uj,ujp,phi))
+ (Fjp(ujp)-Fjhalf(uj,ujp,phi));
−> Pjpminus(ujval,ujpval,phival)$

−> ratsimp(Pjpminus(ujval,ujpval,phival));

$$\Big(\left(phival^3-3phival^2+4\right)\,ujpcell^3+\left(-3phival^3+12phival^2-12phival\right)\,ucell\;ujpcell^2 + \left(3phival^3-15phival^2+24phival-12\right)\,ucell^2\;ujpcell + \left(-phival^3+6phival^2-12phival+8\right)\,ucell^3\Big)/12 \tag{\%o18}$$

-->   factor(%);

$$\frac{(phival-2)^2\,(ujpcell-ucell)^2\,(phival\;ujpcell+ujpcell-phival\;ucell+2ucell)}{12} \quad (\%o19)$$

-->   solve([Pjpminus(ujval,ujpval,phival)],[phival]);

$$\left[phival=-\frac{ujpcell+2ucell}{ujpcell-ucell},phival=2\right] \quad (\%o20)$$

-->   kill(all)$ ujval:ucell; ujpval:ujpcell;

$$ucell \quad \text{(ujval)}$$

$$ujpcell \quad \text{(ujpval)}$$

-->   uhalf(uj,ujp):=(uj + ujp) / 2;

$$\mathrm{uhalf}\,(uj,ujp):=\frac{uj+ujp}{2} \quad (\%o3)$$

-->   S:-uj^2;

$$-uj^2 \quad \text{(S)}$$

-->   Sjp:-ujp^2;

$$-ujp^2 \quad \text{(Sjp)}$$

```
-->    Stemp1:diff(-ujv^2,ujv,1)$
-->    Sprime(uj)::=subst(uj,ujv,Stemp1)$
-->    Stemp2:diff(-ujpv^2,ujpv,1)$
-->    Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$
-->    fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o10}$$

```
-->    fjhalf(uj,ujp):=(uhalf(uj,ujp)^2/2);
```

$$\mathrm{fjhalf}\left(uj, ujp\right) := \frac{\mathrm{uhalf}\left(uj, ujp\right)^2}{2} \tag{\%o11}$$

```
-->    fjp(ujp):=ujp^2/2;
```

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o12}$$

```
-->    Fj(uj):=(-2/3)*uj^3;
```

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o13}$$

```
-->    Fjhalf(uj,ujp):=-(2*uhalf(uj,ujp)^3)/3;
```

$$\mathrm{Fjhalf}\left(uj, ujp\right) := \frac{-2\mathrm{uhalf}\left(uj, ujp\right)^3}{3} \tag{\%o14}$$

−> Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3}\, ujp^3 \quad (\%o15)$$

−> OverlinePjplus(uj,ujp):=-Sprime(uj)* (fjhalf(uj,ujp) - fj(uj))

+ (Fjhalf(uj,ujp) - Fj(uj))$
−> OverlinePjplus(ujval,ujpval);

$$-\frac{\left(ujpcell + ucell\right)^3}{12} + 2ucell\left(\frac{\left(ujpcell + ucell\right)^2}{8} - \frac{ucell^2}{2}\right) + \frac{2ucell^3}{3} \quad (\%o17)$$

−> ratsimp(OverlinePjplus(ujval,ujpval));

$$-\frac{ujpcell^3 - 3ucell^2\, ujpcell + 2ucell^3}{12} \quad (\%o18)$$

−> factor(%);

$$-\frac{\left(ujpcell - ucell\right)^2\,\left(ujpcell + 2ucell\right)}{12} \quad (\%o19)$$

(%i2) kill(all)$ ujval:ucell; ujmval:ujmcell;

$$ucell \quad \text{(ujval)}$$

$$ujmcell \quad \text{(ujmval)}$$

(%i3) uhalf(uj,ujm):=(uj + ujm) / 2;

$$\text{uhalf}\left(uj, ujm\right) := \frac{uj + ujm}{2} \tag{\%o3}$$

(%i4) S:-uj^2;

$$-uj^2 \tag{S}$$

(%i5) Sjm:-ujm^2;

$$-ujm^2 \tag{Sjm}$$

(%i6) Stemp1:diff(-ujv^2,ujv,1)$

(%i7) Sprime(uj)::=subst(uj,ujv,Stemp1)$

(%i8) Stemp2:diff(-ujmv^2,ujmv,1)$

(%i9) Sprimejm(ujm)::=subst(ujm,ujmv,Stemp2)$

(%i10) fj(uj):=(uj^2)/2;

$$\text{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o10}$$

(%i11) fjhalf(uj,ujm):=(uhalf(uj,ujm)^2/2);

$$\text{fjhalf}\left(uj, ujm\right) := \frac{\text{uhalf}\left(uj, ujm\right)^2}{2} \tag{\%o11}$$

(%i12) fjm(ujm):=ujm^2/2;

$$\text{fjm}\left(ujm\right) := \frac{ujm^2}{2} \tag{\%o12}$$

(%i13) Fj(uj):=(-2/3)*uj^3;

$$\text{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o13}$$

(%i14) Fjhalf(uj,ujm):=-(2*uhalf(uj,ujm)^3)/3;

$$\text{Fjhalf}\left(uj, ujm\right) := \frac{-2\text{uhalf}\left(uj, ujm\right)^3}{3} \tag{\%o14}$$

(%i15) Fjm(ujm):=(-2/3)*ujm^3;

$$\text{Fjm}\left(ujm\right) := \frac{-2}{3} ujm^3 \tag{\%o15}$$

(%i16) OverlinePjminus(uj,ujm):=-Sprime(uj)* (fj(uj) - fjhalf(uj,ujm))

+ (Fj(uj) - Fjhalf(uj,ujm))$

(%i17) OverlinePjminus(ujval,ujmval);

$$\frac{(ujmcell + ucell)^3}{12} + 2ucell\left(\frac{ucell^2}{2} - \frac{(ujmcell + ucell)^2}{8}\right) - \frac{2ucell^3}{3} \tag{\%o17}$$

(%i18) ratsimp(OverlinePjminus(ujval,ujmval));

$$\frac{ujmcell^3 - 3ucell^2\, ujmcell + 2ucell^3}{12} \tag{\%o18}$$

(%i19) factor(%);

$$\frac{(ujmcell - ucell)^2 \, (ujmcell + 2\,ucell)}{12} \tag{\%o19}$$

−> kill(all)$ ujpval:upcell; ujp2val:ujp2cell;

$$upcell \tag{ujpval}$$

$$ujp2cell \tag{ujp2val}$$

−> uhalf(ujp,ujp2):=(ujp + ujp2) / 2;

$$\mathrm{uhalf}\,(ujp, ujp2) := \frac{ujp + ujp2}{2} \tag{\%o3}$$

−> S:-uj^2;

$$-uj^2 \tag{S}$$

−> Sjp:-ujp^2;

$$-ujp^2 \tag{Sjp}$$

−> Stemp1:diff(-ujv^2,ujv,1)$

−> Sprime(uj)::=subst(uj,ujv,Stemp1)$

-> Stemp2:diff(-ujpv^2,ujpv,1)$

-> Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

-> fj(uj):=(uj^2)/2;

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o10}$$

-> fjhalf(ujp,ujp2):=(uhalf(ujp,ujp2)^2/2);

$$\mathrm{fjhalf}\left(ujp, ujp2\right) := \frac{\mathrm{uhalf}\left(ujp, ujp2\right)^2}{2} \tag{\%o11}$$

-> fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o12}$$

-> Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o13}$$

-> Fjhalf(ujp,ujp2):=-(2*uhalf(ujp,ujp2)^3)/3;

$$\mathrm{Fjhalf}\left(ujp, ujp2\right) := \frac{-2\mathrm{uhalf}\left(ujp, ujp2\right)^3}{3} \tag{\%o14}$$

-> Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^3 \tag{\%o15}$$

-->   OverlinePjpplus(ujp,ujp2):=-Sprimejp(ujp)* (fjhalf(ujp,ujp2) - fjp(ujp))

+ (Fjhalf(ujp,ujp2) - Fjp(ujp))$

-->   OverlinePjpplus(ujpval,ujp2val);

$$-\frac{(upcell+ujp2cell)^3}{12}+2upcell\left(\frac{(upcell+ujp2cell)^2}{8}-\frac{upcell^2}{2}\right)+\frac{2upcell^3}{3} \qquad (\%o17)$$

-->   ratsimp(OverlinePjpplus(ujpval,ujp2val));

$$-\frac{2upcell^3-3ujp2cell\,upcell^2+ujp2cell^3}{12} \qquad (\%o18)$$

-->   factor(%);

$$-\frac{(upcell-ujp2cell)^2\,(2upcell+ujp2cell)}{12} \qquad (\%o19)$$

(%i2) kill(all)$ ujpval:upcell; ujp2val:ujp2cell;

$$upcell \qquad (ujpval)$$

$$ujp2cell \qquad (ujp2val)$$

(%i3) uhalf(ujp,ujp2):=(ujp + ujp2) / 2;

$$\mathrm{uhalf}(ujp, ujp2) := \frac{ujp+ujp2}{2} \qquad (\%o3)$$

(%i4) S:-uj^2;

$$-uj^2 \tag{S}$$

(%i5) Sjp:-ujp^2;

$$-ujp^2 \tag{Sjp}$$

(%i6) Stemp1:diff(-ujv^2,ujv,1)$

(%i7) Sprime(uj)::=subst(uj,ujv,Stemp1)$

(%i8) Stemp2:diff(-ujpv^2,ujpv,1)$

(%i9) Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

(%i10) fj(uj):=(uj^2)/2;

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o10}$$

(%i11) fjhalf(ujp,ujp2):=(uhalf(ujp,ujp2)^2/2);

$$\mathrm{fjhalf}\left(ujp, ujp2\right) := \frac{\mathrm{uhalf}\left(ujp, ujp2\right)^2}{2} \tag{\%o11}$$

(%i12) fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o12}$$

(%i13) Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o13}$$

(%i14) Fjhalf(ujp,ujp2):=-(2*uhalf(ujp,ujp2)^3)/3;

$$\mathrm{Fjhalf}\left(ujp, ujp2\right) := \frac{-2 \mathrm{uhalf}\left(ujp, ujp2\right)^3}{3} \tag{\%o14}$$

(%i15) Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^3 \tag{\%o15}$$

(%i16) OverlinePjpminus(ujp,ujp2):=-Sprimejp(ujp)*
(fjhalf(ujp,ujp2) - fjp(ujp)) + (Fjhalf(ujp,ujp2) - Fjp(ujp))$
(%i17) OverlinePjpminus(ujpval,ujp2val);

$$-\frac{\left(upcell + ujp2cell\right)^3}{12} + 2 upcell \left(\frac{\left(upcell + ujp2cell\right)^2}{8} - \frac{upcell^2}{2}\right) + \frac{2 upcell^3}{3} \tag{\%o17}$$

(%i18) ratsimp(OverlinePjpminus(ujpval,ujp2val));

$$-\frac{2 upcell^3 - 3 ujp2cell\, upcell^2 + ujp2cell^3}{12} \tag{\%o18}$$

(%i19) factor(%);

$$-\frac{\left(upcell - ujp2cell\right)^2 \left(2 upcell + ujp2cell\right)}{12} \tag{\%o19}$$

APPENDIX B

Fourth-Order Maxima Code

```
--> kill(all)$ ujval:ucell; ujpval:ujpcell; ujmval:ujmcell; ujp2val:ujp2cell;
```

$$ucell \quad \text{(ujval)}$$

$$ujpcell \quad \text{(ujpval)}$$

$$ujmcell \quad \text{(ujmval)}$$

$$ujp2cell \quad \text{(ujp2val)}$$

```
--> uhalfP(ujm,uj,ujp,ujp2,phiP):=((uj+ujp)/2 - phiP/12*(ujm - uj - ujp + ujp2));
```

$$\text{uhalfP}\,(ujm, uj, ujp, ujp2, phiP) := \frac{uj + ujp}{2} - \frac{phiP}{12}\,(ujm - uj - ujp + ujp2) \quad \text{(\%o5)}$$

```
--> ratsimp(%);
--> S:-uj^2;
```

$$-uj^2 \quad \text{(S)}$$

```
-->  Sjp:-ujp^2;
```

$$-ujp^2 \tag{Sjp}$$

```
-->  Stemp1:diff(-ujv^2,ujv,1)$
-->  Sprime(uj)::=subst(uj,ujv,Stemp1)$
-->  Stemp2:diff(-ujpv^2,ujpv,1)$
-->  Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$
-->  fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o13}$$

```
-->  fjhalfP(ujm,uj,ujp,ujp2,phiP):=(uhalfP(ujm,uj,ujp,ujp2,phiP)^2/2);
```

$$\mathrm{fjhalfP}\left(ujm, uj, ujp, ujp2, phiP\right) := \frac{\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2, phiP\right)^2}{2} \tag{\%o14}$$

```
-->  fjp(ujp):=ujp^2/2;
```

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o15}$$

```
-->  Fj(uj):=(-2/3)*uj^3;
```

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o16}$$

-> FjhalfP(ujm,uj,ujp,ujp2,phiP):=-(2*uhalfP(ujm,uj,ujp,ujp2,phiP)^3)/3;

$$\mathrm{FjhalfP}\left(ujm, uj, ujp, ujp2, phiP\right) := \frac{-2\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2, phiP\right)^{3}}{3} \qquad (\%o17)$$

-> Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^{3} \qquad (\%o18)$$

-> Pjplus(ujm,uj,ujp,ujp2,phiP):=-Sprime(uj)* (fjhalfP(ujm,uj,ujp,ujp2,phiP) - fj(uj))
+ (FjhalfP(ujm,uj,ujp,ujp2,phiP) - Fj(uj))$

-> Pjplus(ujmval,ujval,ujpval,ujp2val,phivalP);

$$-\frac{2\left(\frac{ujpcell+ucell}{2}-\frac{phivalP\left(-ujpcell+ujp2cell+ujmcell-ucell\right)}{12}\right)^{3}}{3} + 2ucell\left(\frac{\left(\frac{ujpcell+ucell}{2}-\frac{phivalP\left(-ujpcell+ujp2cell+ujmcell-ucell\right)}{12}\right)^{2}}{2}-\frac{ucell^{2}}{2}\right)+\frac{2ucell^{3}}{3} \qquad (\%o20)$$

-> factor(%);

-> solve([Pjplus(ujmval,ujval,ujpval,ujp2val,phivalP)],[phivalP]);

$$[phivalP = -\frac{6ujpcell+12ucell}{ujpcell-ujp2cell-ujmcell+ucell}, \quad phivalP = \frac{6ucell-6ujpcell}{ujpcell-ujp2cell-ujmcell+ucell}] \qquad (\%o22)$$

(%i4) kill(all)$ ujval:ucell; ujpval:ujpcell; ujmval:ujmcell; ujm2val:ujm2cell;

$$ucell \qquad (\mathrm{ujval})$$

$$ujpcell \tag{ujpval}$$

$$ujmcell \tag{ujmval}$$

$$ujm2cell \tag{ujm2val}$$

(%i5) uhalfM(ujm2,ujm,uj,ujp,phiM):=((uj+ujm)/2 - phiM/12
*(5*ujm + 13*ujm2 - 7*uj + ujp));

$$\mathrm{uhalfM}\left(ujm2, ujm, uj, ujp, phiM\right) := \frac{uj + ujm}{2} - \frac{phiM}{12}\left(5ujm + 13ujm2 + (-7)\ uj + ujp\right) \tag{\%o5}$$

(%i6) S:-uj^2;

$$-uj^2 \tag{S}$$

(%i7) Sjp:-ujp^2;

$$-ujp^2 \tag{Sjp}$$

(%i8) Stemp1:diff(-ujv^2,ujv,1)$

(%i9) Sprime(uj)::=subst(uj,ujv,Stemp1)$

(%i10) Stemp2:diff(-ujpv^2,ujpv,1)$

(%i11) Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

(%i12) fj(uj):=(uj^2)/2;

$$\mathrm{fj}\,(uj) := \frac{uj^2}{2} \tag{\%o12}$$

(%i13) fjhalfM(ujm2,ujm,uj,ujp,phiM):=(uhalfM(ujm2,ujm,uj,ujp,phiM)^2/2);

$$\mathrm{fjhalfM}\,(ujm2, ujm, uj, ujp, phiM) := \frac{\mathrm{uhalfM}\,(ujm2, ujm, uj, ujp, phiM)^2}{2} \tag{\%o13}$$

(%i14) fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\,(ujp) := \frac{ujp^2}{2} \tag{\%o14}$$

(%i15) Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\,(uj) := \frac{-2}{3} uj^3 \tag{\%o15}$$

(%i16) FjhalfM(ujm2,ujm,uj,ujp,phiM):=-(2*uhalfM(ujm2,ujm,uj,ujp,phiM)^3)/3;

$$\mathrm{FjhalfM}\,(ujm2, ujm, uj, ujp, phiM) := \frac{-2\mathrm{uhalfM}\,(ujm2, ujm, uj, ujp, phiM)^3}{3} \tag{\%o16}$$

(%i17) Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\,(ujp) := \frac{-2}{3} ujp^3 \tag{\%o17}$$

(%i18) Pjminus(ujm2,ujm,uj,ujp,phiM):=-Sprime(uj)*

(fj(uj) - fjhalfM(ujm2,ujm,uj,ujp,phiM))

+ (Fj(uj) - FjhalfM(ujm2,ujm,uj,ujp,phiM))$

(%i19) Pjminus(ujm2val,ujmval,ujval,ujpval,phivalM);

$$\frac{2\left(\frac{ujmcell+ucell}{2}-\frac{phivalM\ (ujpcell+5ujmcell+13ujm2cell-7ucell)}{12}\right)^3}{3}+2ucell\left(\frac{ucell^2}{2}-\frac{\left(\frac{ujmcell+ucell}{2}-\frac{phivalM\ (ujpcell+5ujmcell+13ujm2cell-7ucell)}{12}\right)^2}{2}\right)-\frac{2ucell^3}{3} \quad (\%o19)$$

(%i20) ratsimp(Pjminus(ujm2val,ujmval,ujval,ujpval,phivalM))$

(%i21) solve([Pjminus(ujm2val,ujmval,ujval,ujpval,phivalM)],[phivalM]);

$$phivalM=-\frac{6ujmcell+12ucell}{-ujpcell-5ujmcell-13ujm2cell+7ucell},$$
$$phivalM=\frac{6ucell-6ujmcell}{-ujpcell-5ujmcell-13ujm2cell+7ucell} \quad (\%o21)$$

--> kill(all)$ ujval:ucell; ujpval:ujpcell; ujmval:ujmcell; ujp2val:ujp2cell;

$$ucell \quad (ujval)$$

$$ujpcell \quad (ujpval)$$

$$ujmcell \tag{ujmval}$$

$$ujp2cell \tag{ujp2val}$$

```
-->   uhalfP(ujm,uj,ujp,ujp2,phiP):=(ujp+uj)/2 - phiP/12*((ujm - uj - ujp + ujp2));
```

$$\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2, phiP\right) := \frac{ujp + uj}{2} - \frac{phiP}{12}\left(ujm - uj - ujp + ujp2\right) \tag{\%o5}$$

```
-->   ratsimp(%);
-->   S:-uj^2;
```

$$-uj^2 \tag{S}$$

```
-->   Sjp:-ujp^2;
```

$$-ujp^2 \tag{Sjp}$$

```
-->   Stemp1:diff(-ujv^2,ujv,1)$
-->   Sprime(uj)::=subst(uj,ujv,Stemp1)$
-->   Stemp2:diff(-ujpv^2,ujpv,1)$
-->   Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$
-->   fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o13}$$

```
-->   fjhalfP(ujm,uj,ujp,ujp2,phiP):=(uhalfP(ujm,uj,ujp,ujp2,phiP)^2/2);
```

$$\mathrm{fjhalfP}\left(ujm, uj, ujp, ujp2, phiP\right) := \frac{\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2, phiP\right)^2}{2} \qquad (\%o14)$$

```
-->   fjp(ujp):=ujp^2/2;
```

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \qquad (\%o15)$$

```
-->   Fj(uj):=(-2/3)*uj^3;
```

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \qquad (\%o16)$$

```
-->   FjhalfP(ujm,uj,ujp,ujp2,phiP):=-(2*uhalfP(ujm,uj,ujp,ujp2,phiP)^3)/3;
```

$$\mathrm{FjhalfP}\left(ujm, uj, ujp, ujp2, phiP\right) := \frac{-2\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2, phiP\right)^3}{3} \qquad (\%o17)$$

```
-->   Fjp(ujp):=(-2/3)*ujp^3;
```

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^3 \qquad (\%o18)$$

```
-->   Pjpminus(ujm,uj,ujp,ujp2,phiP):=-Sprimejp(ujp)*
      (fjp(ujp) - fjhalfP(ujm,uj,ujp,ujp2,phiP))
      + (Fjp(ujp) - FjhalfP(ujm,uj,ujp,ujp2,phiP))$
```

-> Pjpminus(ujmval,ujval,ujpval,ujp2val,phivalP);

$$\frac{2\left(\frac{ujpcell+ucell}{2}-\frac{phivalP\,(-ujpcell+ujp2cell+ujmcell-ucell)}{12}\right)^3}{3}$$
$$+2ujpcell\left(\frac{ujpcell^2}{2}-\frac{\left(\frac{ujpcell+ucell}{2}-\frac{phivalP\,(-ujpcell+ujp2cell+ujmcell-ucell)}{12}\right)^2}{2}\right)-\frac{2ujpcell^3}{3} \tag{\%o20}$$

-> ratsimp(Pjpminus(ujmval,ujval,ujpval,ujp2val,phivalP))$

-> factor(%);

-> solve([Pjpminus(ujmval,ujval,ujpval,ujp2val,phivalP)],[phivalP]);

$$[phivalP=-\frac{12ujpcell+6ucell}{ujpcell-ujp2cell-ujmcell+ucell}, \tag{\%o23}$$
$$phivalP=-\frac{6ucell-6ujpcell}{ujpcell-ujp2cell-ujmcell+ucell}]$$

-> kill(all)$ ujval:ucell; ujpval:ujpcell; ujp2val:ujp2cell; ujp3val:ujp3cell;

$$ucell \tag{ujval}$$

$$ujpcell \tag{ujpval}$$

$$ujp2cell \tag{ujp2val}$$

$$ujp3cell \tag{ujp3val}$$

```
-->   uhalfP(uj,ujp,ujp2,ujp3,phiP):=((ujp+ujp2)/2 - phiP/12*
      (5*ujp + 13*uj - 7*ujp2 + ujp3));
```

$$\mathrm{uhalfP}\left(uj, ujp, ujp2, ujp3, phiP\right) := \frac{ujp + ujp2}{2} - \frac{phiP}{12}\left(5 ujp + 13 uj + (-7)\ ujp2 + ujp3\right) \tag{\%o5}$$

```
-->   S:-uj^2;
```

$$-uj^2 \tag{S}$$

```
-->   Sjp:-ujp^2;
```

$$-ujp^2 \tag{Sjp}$$

```
-->   Stemp1:diff(-ujv^2,ujv,1)$
-->   Sprime(uj)::=subst(uj,ujv,Stemp1)$
-->   Stemp2:diff(-ujpv^2,ujpv,1)$
-->   Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$
-->   fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o12}$$

```
-->   fjhalfP(uj,ujp,ujp2,ujp3,phiP):=(uhalfP(uj,ujp,ujp2,ujp3,phiP)^2/2);
```

$$\mathrm{fjhalfP}\left(uj, ujp, ujp2, ujp3, phiP\right) := \frac{\mathrm{uhalfP}\left(uj, ujp, ujp2, ujp3, phiP\right)^2}{2} \tag{\%o13}$$

-> fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\left(ujp\right):=\frac{ujp^2}{2} \tag{\%o14}$$

-> Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\left(uj\right):=\frac{-2}{3}uj^3 \tag{\%o15}$$

-> FjhalfP(uj,ujp,ujp2,ujp3,phiP):=-(2*uhalfP(uj,ujp,ujp2,ujp3,phiP)^3)/3;

$$\mathrm{FjhalfP}\left(uj, ujp, ujp2, ujp3, phiP\right):=\frac{-2\mathrm{uhalfP}\left(uj, ujp, ujp2, ujp3, phiP\right)^3}{3} \tag{\%o16}$$

-> Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right):=\frac{-2}{3}ujp^3 \tag{\%o17}$$

-> Pjplus(uj,ujp,uj2,ujp3,phiP):=-Sprimejp(ujp)*

(fjhalfP(uj,ujp,ujp2,ujp3,phiP) - fjp(ujp))

\+ (FjhalfP(uj,ujp,ujp2,ujp3,phiP) - Fjp(ujp))$
-> Pjplus(ujval,ujpval,ujp2val,ujp3val,phivalP);

$$-\frac{2\left(\frac{ujpcell+ujp2}{2}-\frac{phivalP\left(5ujpcell+ujp3cell-7ujp2+13ucell\right)}{12}\right)^3}{3}$$
$$+2ujpcell\left(\frac{\left(\frac{ujpcell+ujp2}{2}-\frac{phivalP\left(5ujpcell+ujp3cell-7ujp2+13ucell\right)}{12}\right)^2}{2}-\frac{ujpcell^2}{2}\right)+\frac{2ujpcell^3}{3} \tag{\%o19}$$

−> ratsimp(Pjplus(ujval,ujpval,ujp2val,ujp3val,phivalP))$

−> solve([Pjplus(ujval,ujpval,ujp2val,ujp3val,phivalP)],[phivalP]);

$$[phivalP = \frac{12 ujpcell + 6 ujp2}{5 ujpcell + ujp3cell - 7 ujp2 + 13 ucell}, \quad (\%o21)$$

$$phivalP = \frac{6 ujp2 - 6 ujpcell}{5 ujpcell + ujp3cell - 7 ujp2 + 13 ucell}]$$

−> kill(all)$ ujval:ucell; ujpval:ujpcell; ujmval:ujmcell; ujm2val:ujm2cell;

$$ucell \quad \text{(ujval)}$$

$$ujpcell \quad \text{(ujpval)}$$

$$ujmcell \quad \text{(ujmval)}$$

$$ujm2cell \quad \text{(ujm2val)}$$

−> uhalfM(ujm2,ujm,uj,ujp):=7/12*(uj + ujm) - 1/12*(ujm2 + ujp);;

$$\text{uhalfM}\left(ujm2, ujm, uj, ujp\right) := \frac{7}{12}\left(uj + ujm\right) - \frac{1}{12}\left(ujm2 + ujp\right) \quad (\%o5)$$

−> S:-uj^2;

$$-uj^2 \quad \text{(S)}$$

```
-->  Sjp:-ujp^2;
```

$$-ujp^2 \tag{Sjp}$$

```
-->  Stemp1:diff(-ujv^2,ujv,1)$
-->  Sprime(uj)::=subst(uj,ujv,Stemp1)$
-->  Stemp2:diff(-ujpv^2,ujpv,1)$
-->  Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$
-->  fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o12}$$

```
-->  fjhalfM(ujm2,ujm,uj,ujp):=(uhalfM(ujm2,ujm,uj,ujp)^2/2);
```

$$\mathrm{fjhalfM}\left(ujm2, ujm, uj, ujp\right) := \frac{\mathrm{uhalfM}\left(ujm2, ujm, uj, ujp\right)^2}{2} \tag{\%o13}$$

```
-->  fjp(ujp):=ujp^2/2;
```

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o14}$$

```
-->  Fj(uj):=(-2/3)*uj^3;
```

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o15}$$

−> FjhalfM(ujm2,ujm,uj,ujp):=-(2*uhalfM(ujm2,ujm,uj,ujp)ˆ3)/3;

$$\mathrm{FjhalfM}\left(ujm2, ujm, uj, ujp\right) := \frac{-2\mathrm{uhalfM}\left(ujm2, ujm, uj, ujp\right)^{3}}{3} \qquad (\%o16)$$

−> Fjp(ujp):=(-2/3)*ujpˆ3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3}ujp^{3} \qquad (\%o17)$$

−> OverlinePjminus(ujm2,ujm,uj,ujp):=-Sprime(uj)*

(fj(uj) - fjhalfM(ujm2,ujm,uj,ujp))

+ (Fj(uj) - FjhalfM(ujm2,ujm,uj,ujp))$

−> OverlinePjminus(ujm2val,ujmval,ujval,ujpval);

$$\frac{2\left(\frac{7(ujmcell+ucell)}{12}-\frac{ujpcell+ujm2cell}{12}\right)^{3}}{3}+2ucell\left(\frac{ucell^{2}}{2}-\frac{\left(\frac{7(ujmcell+ucell)}{12}-\frac{ujpcell+ujm2cell}{12}\right)^{2}}{2}\right) \qquad (\%o19)$$

$$-\frac{2ucell^{3}}{3}$$

−> ratsimp(OverlinePjminus(ujm2val,ujmval,ujval,ujpval));

−> factor(%);

$$-\frac{\left(ujpcell-7ujmcell+ujm2cell-13ucell\right)\left(ujpcell-7ujmcell+ujm2cell+5ucell\right)^{2}}{2592} \qquad (\%o21)$$

```
--> kill(all)$ ujval:ucell; ujpval:ujpcell; ujmval:ujmcell; ujp2val:ujp2cell;
```

$$ucell \quad \text{(ujval)}$$

$$ujpcell \quad \text{(ujpval)}$$

$$ujmcell \quad \text{(ujmval)}$$

$$ujp2cell \quad \text{(ujp2val)}$$

```
--> uhalfP(ujm,uj,ujp,ujp2):=7/12*(uj + ujp) - 1/12*(ujm + ujp2);
```

$$\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2\right) := \frac{7}{12}\left(uj + ujp\right) - \frac{1}{12}\left(ujm + ujp2\right) \quad \text{(\%o5)}$$

```
--> S:-uj^2;
```

$$-uj^2 \quad \text{(S)}$$

```
--> Sjp:-ujp^2;
```

$$-ujp^2 \quad \text{(Sjp)}$$

```
--> Stemp1:diff(-ujv^2,ujv,1)$
```

```
-->    Sprime(uj)::=subst(uj,ujv,Stemp1)$
-->    Stemp2:diff(-ujpv^2,ujpv,1)$
-->    Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$
-->    fj(uj):=(uj^2)/2;
```

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \qquad (\%o12)$$

```
-->    fjhalfP(ujm,uj,ujp,ujp2):=(uhalfP(ujm,uj,ujp,ujp2)^2/2);
```

$$\mathrm{fjhalfP}\left(ujm, uj, ujp, ujp2\right) := \frac{\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2\right)^2}{2} \qquad (\%o13)$$

```
-->    fjp(ujp):=ujp^2/2;
```

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \qquad (\%o14)$$

```
-->    Fj(uj):=(-2/3)*uj^3;
```

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \qquad (\%o15)$$

```
-->    FjhalfP(ujm,uj,ujp,ujp2):=-(2*uhalfP(ujm,uj,ujp,ujp2)^3)/3;
```

$$\mathrm{FjhalfP}\left(ujm, uj, ujp, ujp2\right) := \frac{-2\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2\right)^3}{3} \qquad (\%o16)$$

```
-->    Fjp(ujp):=(-2/3)*ujp^3;
```

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^3 \qquad (\%o17)$$

-> OverlinePjplus(ujm,uj,ujp,ujp2):=-Sprime(uj)*

(fjhalfP(ujm,uj,ujp,ujp2) - fj(uj))

+ (FjhalfP(ujm,uj,ujp,ujp2) - Fj(uj))$

-> OverlinePjplus(ujmval,ujval,ujpval,ujp2val);

$$-\frac{2\left(\frac{7(ujpcell+ucell)}{12}-\frac{ujp2cell+ujmcell}{12}\right)^3}{3}+2ucell\left(\frac{\left(\frac{7(ujpcell+ucell)}{12}-\frac{ujp2cell+ujmcell}{12}\right)^2}{2}-\frac{ucell^2}{2}\right)+\frac{2ucell^3}{3} \quad (\%o19)$$

-> ratsimp(OverlinePjplus(ujmval,ujval,ujpval,ujp2val));

-> factor(%);

$$-\frac{(7ujpcell-ujp2cell-ujmcell-5ucell)^2\,(7ujpcell-ujp2cell-ujmcell+13ucell)}{2592} \quad (\%o21)$$

(%i4) kill(all)$ ujval:ucell; ujpval:ujpcell; ujmval:ujmcell; ujp2val:ujp2cell;

$$ucell \quad \text{(ujval)}$$

$$ujpcell \quad \text{(ujpval)}$$

$$ujmcell \quad \text{(ujmval)}$$

$$ujp2cell \tag{ujp2val}$$

(%i5) uhalfP(ujm,uj,ujp,ujp2):=7/12*(ujp + uj) - 1/12*(ujp2 + ujm);

$$\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2\right) := \frac{7}{12}\left(ujp + uj\right) - \frac{1}{12}\left(ujp2 + ujm\right) \tag{\%o5}$$

(%i6) S:-uj^2;

$$-uj^{2} \tag{S}$$

(%i7) Sjp:-ujp^2;

$$-ujp^{2} \tag{Sjp}$$

(%i8) Stemp1:diff(-ujv^2,ujv,1)$

(%i9) Sprime(uj)::=subst(uj,ujv,Stemp1)$

(%i10) Stemp2:diff(-ujpv^2,ujpv,1)$

(%i11) Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

(%i12) fj(uj):=(uj^2)/2;

$$\mathrm{fj}\left(uj\right) := \frac{uj^{2}}{2} \tag{\%o12}$$

(%i13) fjhalfP(ujm,uj,ujp,ujp2):=(uhalfP(ujm,uj,ujp,ujp2)^2/2);

$$\mathrm{fjhalfP}\left(ujm, uj, ujp, ujp2\right) := \frac{\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2\right)^{2}}{2} \tag{\%o13}$$

(%i14) fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o14}$$

(%i15) Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o15}$$

(%i16) FjhalfP(ujm,uj,ujp,ujp2):=-(2*uhalfP(ujm,uj,ujp,ujp2)^3)/3;

$$\mathrm{FjhalfP}\left(ujm, uj, ujp, ujp2\right) := \frac{-2\mathrm{uhalfP}\left(ujm, uj, ujp, ujp2\right)^3}{3} \tag{\%o16}$$

(%i17) Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^3 \tag{\%o17}$$

(%i18) OverlinePjpminus(ujm,uj,ujp,ujp2):=-Sprimejp(ujp)*

(fjp(ujp) - fjhalfP(ujm,uj,ujp,ujp2))

+ (Fjp(ujp) - FjhalfP(ujm,uj,ujp,ujp2))$

(%i19) OverlinePjpminus(ujmval,ujval,ujpval,ujp2val);

$$\frac{2\left(\frac{7(ujpcell+ucell)}{12} - \frac{ujp2cell+ujmcell}{12}\right)^3}{3} + 2ujpcell\left(\frac{ujpcell^2}{2} - \frac{\left(\frac{7(ujpcell+ucell)}{12} - \frac{ujp2cell+ujmcell}{12}\right)^2}{2}\right) - \frac{2ujpcell^3}{3} \tag{\%o19}$$

(%i20) factor(%);

$$\frac{(5\,ujpcell+ujp2cell+ujmcell-7\,ucell)^{2}\,(13\,ujpcell-ujp2cell-ujmcell+7\,ucell)}{2592} \quad (\%o20)$$

(%i4) kill(all)$ ujval:ucell; ujpval:ujpcell; ujp2val:ujp2cell; ujp3val:ujp3cell;

$$ucell \quad (ujval)$$

$$ujpcell \quad (ujpval)$$

$$ujp2cell \quad (ujp2val)$$

$$ujp3cell \quad (ujp3val)$$

(%i5) uhalfP(uj,ujp,ujp2,ujp3):=7/12*(ujp + ujp2) - 1/12 * (ujp3 + uj);

$$\mathrm{uhalfP}\left(uj, ujp, ujp2, ujp3\right) := \frac{7}{12}\left(ujp+ujp2\right)-\frac{1}{12}\left(ujp3+uj\right) \quad (\%o5)$$

(%i6) S:-uj^2;

$$-uj^{2} \quad (S)$$

(%i7) Sjp:-ujp^2;

$$-ujp^2 \tag{Sjp}$$

(%i8) Stemp1:diff(-ujv^2,ujv,1)$

(%i9) Sprime(uj)::=subst(uj,ujv,Stemp1)$

(%i10) Stemp2:diff(-ujpv^2,ujpv,1)$

(%i11) Sprimejp(ujp)::=subst(ujp,ujpv,Stemp2)$

(%i12) fj(uj):=(uj^2)/2;

$$\mathrm{fj}\left(uj\right) := \frac{uj^2}{2} \tag{\%o12}$$

(%i13) fjhalfP(uj,ujp,ujp2,ujp3):=(uhalfP(uj,ujp,ujp2,ujp3)^2/2);

$$\mathrm{fjhalfP}\left(uj, ujp, ujp2, ujp3\right) := \frac{\mathrm{uhalfP}\left(uj, ujp, ujp2, ujp3\right)^2}{2} \tag{\%o13}$$

(%i14) fjp(ujp):=ujp^2/2;

$$\mathrm{fjp}\left(ujp\right) := \frac{ujp^2}{2} \tag{\%o14}$$

(%i15) Fj(uj):=(-2/3)*uj^3;

$$\mathrm{Fj}\left(uj\right) := \frac{-2}{3} uj^3 \tag{\%o15}$$

(%i16) FjhalfP(uj,ujp,ujp2,ujp3):=-(2*uhalfP(uj,ujp,ujp2,ujp3)^3)/3;

$$\mathrm{FjhalfP}\left(uj, ujp, ujp2, ujp3\right) := \frac{-2\mathrm{uhalfP}\left(uj, ujp, ujp2, ujp3\right)^3}{3} \tag{\%o16}$$

(%i17) Fjp(ujp):=(-2/3)*ujp^3;

$$\mathrm{Fjp}\left(ujp\right) := \frac{-2}{3} ujp^3 \tag{\%o17}$$

(%i18) OverlinePjplus(uj,ujp,ujp2,ujp3):=-Sprimejp(ujp)*

(fjhalfP(uj,ujp,ujp2,ujp3) - fjp(ujp))

+ (FjhalfP(uj,ujp,ujp2,ujp3) - Fjp(ujp))$

(%i19) OverlinePjplus(ujval,ujpval,ujp2val,ujp3val);

(%i20) ratsimp(OverlinePjplus(ujval,ujpval,ujp2val,ujp3val));

(%i21) factor(%);

$$-\frac{(5ujpcell + ujp3cell - 7ujp2cell + ucell)^2\ (13ujpcell - ujp3cell + 7ujp2cell - ucell)}{2592} \tag{\%o21}$$