Multidimensional Schemes for Hyperbolic Conservation Laws on Triangular Meshes

Qurrat-ul-Ain

Fakultät für Mathematik

Otto-von-Guericke Universität Magdeburg

Multidimensional Schemes for Hyperbolic Conservation Laws on Triangular Meshes

Dissertation
zur Erlangung des akademischen Grades

doctor rerum naturalium
(Dr. rer. nat.)

genehmigt durch die Fakultät für Mathematik
der Otto-von-Guericke-Universität Magdeburg

von **M.Phil Physik Qurrat-ul-Ain**
geb. am 02. Januar 1970 in Mardan, Pakistan

Gutachter:
Prof. Dr. Jiequan Li
Prof. Dr. Mária Lukáčová
Prof. Dr. Gerald Warnecke

Eingericht am: 1. März 2005
Verteidigung am: 25. April 2005

Abstract

In this dissertation we present two kinds of multidimensional schemes for hyperbolic systems based on triangular meshes. The first kind of schemes are evolution Galerkin schemes (EG) which are truly multidimensional schemes and the second kind is a new space-time conservative central-type method which we name a slope propagation (SP) method.

Our first scheme is an extension of the EG schemes for hyperbolic systems from rectangular to triangular meshes. We develop EG schemes for the linear wave equation system, the nonlinear wave equation system, the linearized Euler equations, the advection wave equation system and the nonlinear Euler equations for structured/unstructured triangular meshes. We have also extended these EG schemes on triangular meshes to second order by using linear reconstruction. The accuracy and experimental order of convergence (EOC) of the schemes are demonstrated by numerical experiments. The accuracy of second order scheme is several times greater than the first order however the EOC of 2 has not been achieved. Several numerical test cases are presented which show that apart from such convergence difficulties EG schemes work equally well for structured and unstructured triangular meshes.

Our second scheme i.e. the SP-method, which we have newly introduced is a space-time conservative second-order scheme. The scheme treats space and time in a unified manner. The flow variables and their slopes are the basic unknowns in the scheme. The scheme utilizes the advantages of the space-time conservation element and solution element (CE/SE) method of Chang [6] as well as central schemes of Nessyahu and Tadmor [48]. However, unlike the CE/SE method the present scheme is Jacobian-free and hence like the central schemes can also be applied to any hyperbolic system. In Chang's method a finite difference approach is being used for the slope calculation in case of nonlinear hyperbolic equations. We propose to propagate the slopes by a scheme even in the case of nonlinear systems. By introducing a suitable limiter for the slopes of flow variables, we can apply the same scheme to linear and non-linear problems with discontinuities. The scheme is simple, efficient and has a good resolution especially at contact discontinuities. We derive the scheme for one and two space dimensions. In two-space dimensions we use structured triangular mesh. The second order accuracy of the scheme has been verified by numerical experiments. Several numerical tests presented in this dissertation validate the accuracy and robustness of the present scheme.

Zusammenfassung

In dieser Dissertation stellen wir zwei verschiedene mehrdimensionale numerische Lösungsschemata auf Dreiecksgittern für hyperbolische Erhaltungsgleichungen vor. Beim ersten Schema handelt es sich um echt mehrdimensionale Evolution Galerkin (EG) Verfahren, und beim zweiten um ein neues Zentralschema bzgl. Zeit und Ort, das wir im folgenden als Slope-Propagation-Verfahren (SP-Verfahren) bezeichnen. Unser erstes Schema ist eine

Erweiterung des EG-Verfahrens für hyperbolische Systeme von Rechtecksgittern auf Dreicksgitter. Wir entwickeln dieses EG-Schema sowohl für lineare als auch für nichtlineare Systeme von Wellengleichungen mit Anwendung auf die Eulergleichungen bzw. Advektions-Wellengleichungen. Unter Benutzung der linearen Rekonstruktion haben wir diese EG-Verfahren auf Dreiecksgittern auf die zweite Ordnung übertragen. Die Genauigkeit und die numerisch ermittelte Konvergenzordung mit der EOC (experimental order of convergence) Methode werden anhand numerischer Testbeispiele überprüft. Das Schema zweiter Order erweist sich dabei als wesentlich genauer gegenüber dem Schema erster Ordnung, obwohl die EOC Methode eine Konvergenzordnung kleiner als zwei vermuten lässt. Die verschiedenen numerischen Testfälle belegen dabei eindeutig, daß trotz der noch nicht geklärten Frage der Konvergenzordung diese EG Verfahren sowohl auf regelmäßigen als auch auf unstrukturierten Dreicksgittern vergleichbar gute Ergebnisse liefern. Die konservative

Formulierung unseres jüngst eingeführten SP Schemas behandelt die Zeit- bzw. Ortsvariable vollkommen gleichwertig, und führt so bezüglich dieser Variablen in natürlicher Weise auf ein neues Verfahren zweiter Ordung. Die konservativen Variablen und ihre partiellen Ortsableitungen sind die Unbekannten des Schemas. Dieses Schema vereinigt in sich die Vorteile einer konservativen Formulierung mit gleichwertiger Behandlung von Zeit- und Ortskoordinaten wie in der CE/SE Methode von Chang [6] und die Vorteile eines Zentralschemas von Nessyahu und Tadmor [48]. Im Gegensatz zur CE/SE Methode ist unser Schema frei von der Jacobi Determinante der konservativen Variablen und kann daher wie das Zentralschema auf sehr allgemeine hyperbolische Systeme angewendet werden. Durch die Einführung geeigneter Limiter für die Steigungen der konservativen Variablen können wir dasselbe Schema sowohl auf lineare als auch auf nichtlineare hyperbolische Systeme mit unstetigen Anfangsdaten anwenden. Im Gegensatz dazu verwendet die Chang Methode einen finiten Differenzenansatz für die Berechnung der Steigungen im Falle nichtlinearer hyperbolischer Systeme mit unstetigen Anfdangsdaten. Das Schema ist einfach zu implementieren und benötigt moderate Rechenzeiten bei guter Stoß-Auflösung selbst an Kontaktunstetigkeiten. Wir leiten das Schema in ein bzw. zwei Raumdimensionen her, wobei wir in zwei Raumdimensionen ein reguläres Dreiecksgitter zugrunde legen. Sowohl im räumlich ein- und zweidimensionalen Fall ergab sich hierbei mit der EOC Methode die Konvergenzordnung zwei! Mehrere numerische Testrechnungen belegen eine hohe Genauigkeit und Stabilität des Verfahrens.

Acknowledgements

I am highly indebted to my supervisor Prof. Gerald Warnecke for giving me opportunity to carry out Ph.D studies at Magdeburg university. His invaluable guidance and kind behavior encouraged me to accomplish this work successfully. I am also grateful to Prof. Dr. Mária Lukáčová for her useful technical discussions and suggestions. Finally I am also thankful to Prof. Dr. Jiequan Li for his interest in my thesis and many useful ideas.

I express my deep gratitude to Dr. Yousaf Zahaykah and Dr. Shamsul Qamar for their cooperation at each step. I am also thankful to our colleagues Dr. Matthias Kunik and Wolfram Heineken for their cooperation.

I would also like to thank my father for his best wishes and ambitions for me to get higher education.

Many thanks for Dr. Tariq Solaija and Dr. Abdul Waheed Quraishi at my office in Islamabad for granting me permission to pursue higher studies.

This work is partially supported by the project:

Adaptive error analysis for non-stationary hyperbolic systems in reactive and multi-phase flow", contract # DFG WA 633/10-3.

Contents

Chapter 1

Introduction

The fact that certain quantities such as energy, momentum and charge are conserved in physical processes has led to an increasing number of conservation laws. With the advent of quantum physics, new conserved quantities such as baryon and lepton numbers have been found. Certain conservation laws which lead to hyperbolic differential equations are known as hyperbolic conservation laws which govern a broad spectrum of physical phenomena in various fields e.g. material science, solid state physics, astrophysics, cosmology, fluid dynamics, atmospheric physics and multiphase flows. New problems in plasma physics, lasers and nonlinear optics created interest in the developments of the theory of nonlinear hyperbolic equations. In recent years major progress has been made in developing the theoretical and the numerical aspects of this field. Theory of conservation laws can be found in [19], [28], [34], [59] and [67]. Examples of first order hyperbolic systems are the wave equation system, the Maxwell equations and the Euler equations.

Solutions to many hyperbolic equations contain localized phenomena, for example sharp transition layers and discontinuities or complicated patterns in time (turbulence). In such cases the exact solution is very difficult to obtain, hence a good numerical approximation is needed to resolve these discontinuities efficiently. Examples of nonlinear waves are solutions to the Euler equations of gas dynamics, electromagnetic waves in crystals, dynamics of atomic lattices, surface ocean waves, light propagation along optical wave guides and traffic problems. Many of the numerical schemes use the finite element method (FEM) or the finite volume method (FVM) as a discretization procedure. The finite element method is mostly used for boundary value problems in the incompressible fluid flow, mechanical deformations and electromagnetic fields. The advantage of the finite element method is that it is very natural for problems that come from a variational formulation.

On the other hand the FVM, based on the integral formulation of the conservation laws or other balance laws in divergence form, fulfills the discrete conservation property. It also resolves discontinuities, e.g. shocks, efficiently. The FVM can discretize a domain in space using triangles, quadrilaterals or other polygons in 2D and tetrahedra or other polyhedra in 3D. This proves the FVM, like the FEM, to be a better discretization technique than the fi-

nite difference method for complex geometries where an unstructured grid is advantageous. Physical conservation laws are given by integrals over finite volumes and the FVM is based on the integral formulation of the fluxes over the boundary of the discretization cells which are called the control volumes. Hence the FVM is locally conservative. This property is very important especially in problems where fluxes are important such as fluid dynamics and heat transfer. It gives the approximate value for the derivative of a field at a given point using the values of the field at a few locations neighboring the point. The method uses the divergence theorem, constructs a finite volume around the point, discretizes the surface bounding the volume and applies the conservation law at each finite volume. The FVM is also a cheap and feasible method for industrial problems and can be more flexible than finite difference methods.

Many phenomena in nature which lead to multidimensional systems of hyperbolic differential equations involve infinite directions of wave propagation, hence for any numerical scheme used to solve multidimensional systems, it is important to take into account the infinitely many directions of wave propagation, otherwise the solution suffers from large discrepancies. Some of the numerical schemes exploit dimensional splitting. The splitting takes into account the mesh orientation which leads to errors in the solution. Flux vector splitting schemes (FVS) [31] take into account wave interactions in a few directions which contribute to the numerical dissipation. In 1994 Quirk [56] reported the failures of one-dimensional Riemann solvers in multidimensional problems like odd-even decoupling, carbuncle phenomenon and attributed the failure due to the limitations of one dimensional method and deficiencies of conservative properties. Later LeVeque [33] also used one-dimensional Riemann solvers to solve multidimensional hyperbolic systems of conservation laws by taking into account the fluxes normal to the interfaces as well as in the tangential directions.

There are two main classes of finite volume schemes, Godunov-type upwind schemes and central schemes. In both types of methods the approximate solution is realized by a piecewise polynomial which is reconstructed from evolving cell-averages.

Godunov's original scheme [21] forms the basis of all *upwind schemes*. Its high order and multidimensional generalizations were constructed, analyzed, and implemented with great success during the 1970s and 1980s, consult [20, 34, 64] and references therein. Upwind schemes evaluate their cell-averages over the same spatial cells at all time steps. This in turn requires characteristic information along the discontinuous interfaces between these spatial cells. It is needed to rudimentarily construct the wave structure even when using approximate Riemann solvers and dimensional splitting for multidimensional problems. This greatly complicates the upwind approach, especially for more complex problems, see the book by Toro [64] for further details.

All the *central schemes* can be viewed as extensions of the first-order Lax-Friedrichs (LxF) scheme [18]. Like the Godunov scheme, these are based on piecewise polynomials approx-

imate solution. However, their Riemann-solver-free recipe is considerably simpler.

The common feature of all NT central schemes is the evolution of cell averages over staggered cells, that is, cells which alternate every other time step. The importance of staggering is due to the fact that fluxes are computed in neighborhoods around the smooth midcells of the previous time step. The main advantage is *simplicity* due to the replacement of costly Riemann characteristic decompositions from the upwind framework with straightforward *component wise* quadratures and in higher dimensions the dimensional splitting errors are avoided. At the same time, the use of high-order nonoscillatory piecewise polynomials, which are reconstructed from the staggered cell-averages, retain high resolution that is comparable with upwind results. For further study of these schemes the reader is referred to [2, 25, 35, 48].

Morton et al. [47] used the classical characteristic theory, see e.g. Courant and Hilbert [13], for general linear hyperbolic systems in the context of the finite element method and derived the so called Evolution Galerkin Schemes (EG). These schemes belong to the category of upwind schemes and are genuinely multidimensional as they take into account infinite directions of wave propagation. They shifted the transported quantities along the bicharacteristics which were straight lines in this case and then projected on to a finite element space. Ostkamp [52] extended the idea of EG schemes to the wave equation and to the nonlinear Euler equations in two space dimensions, however her scheme involved the calculations of three-dimensional integrals which were not practically feasible especially for shallow water equations and the nonlinear Euler equations. To overcome this problem Lukáčová, Morton and Warnecke [41] proposed the finite volume evolution Galerkin schemes (FVEG) namely EG1, EG2, EG3. In these methods the fluxes are evaluated at the cell interfaces by using the approximate evolution operators applied at the quadrature points. Since the approximate evolution operator involves integration around the sonic circle which constitutes the base of the characteristic cone, all the infinitely many directions of wave propagations are taken into consideration. These schemes are therefore regarded as truly multidimensional schemes. In [71] another approximate evolution operator has been derived which is referred to as EG4 scheme for the two-dimensional wave equation system. This has been derived from the integral equations by neglecting higher order terms. Both the EG3 and EG4 schemes are of comparable efficiency however the numerical dissipation for EG4 is slightly greater than for the EG3 scheme. The approximate evolution operator for the solution of wave equation in three space dimensions has also been derived in [71]. These methods and their finite volume versions were applied to the nonlinear Euler equations, see [43], [45], as well as to the linearized Euler equations and Maxwell equations, see [37], using a square mesh grid.

It is important to mention that these schemes are only first order schemes when a space of piecewise constant functions is used. To obtain a higher order approximation, a recovery stage must be coupled to the approximate evolution operator. Higher order finite volume EG methods have been introduced and studied in [38], [43] and [44]. In [43] a conservative

bilinear recovery is used to get second order approximation for nonlinear Euler equations. Error estimates of the schemes have also been discussed. Third order EG scheme [38], has been derived for a two-dimensional wave equation system. In [46], well balanced FVEG schemes have been developed for shallow water equations with a geometrical source term modeling the bottom topography. The stationary states and the quasi stationary states are evaluated using FVEG scheme. In [40], the stability limits of the first and the second order FVEG schemes are estimated using von-Neumann analysis and Fourier transformation.

In this dissertation we have developed the above mentioned upwind EG schemes for structured/unstructured triangular meshes. We have applied these schemes to the linear and nonlinear wave equation systems, the linearized Euler equations, the advection wave equation and the nonlinear Euler equations. Which turned out to be technically challenging.

Apart from the above mentioned schemes there is another family of schemes called space-time conservation element and solution element (CE/SE) methods of Chang et al. [6, 7, 8, 73]. Like central schemes, these schemes also do not need Riemann solvers. However, unlike the upwind schemes and central schemes, the flow variable distribution inside the solution element (SE) is not calculated through a reconstruction procedure using its neighbouring values at the same time level. Instead they are calculated as a part of local space-time flux conservation in the linear case. However in the nonlinear case Chang [6] reverted to reconstruction by finite differences. A similar procedure was used by Xu [70] for an upwind kinetic scheme. We propose a way to maintain the slope propagation idea even in the nonlinear case.

In the CE/SE method, the space time domain of interest is first divided into many conservative Elements (CEs). These conservation elements are non-overlapping space-time domains such that the computational domain is the union of these subdomains. The flux conservation can be enforced over each of these subdomains and can also be applied to their union. In each solution element (SE), flow variables are assumed continuous. A first order Taylor series is then used by Chang et al. to discretize the flow variables. Thus the scheme is second order accurate. Across the boundaries of neighbouring SEs, flow discontinuities are allowed. Flow variables are calculated through a local space-time flux balance, which is enforced by integrating over the surfaces of conservation elements (CE). The number of the CEs employed matches the number of unknowns designated by the scheme. In addition to the flow variables, the spatial gradients of the flow variables are also treated as unknowns. This means that the slopes are propagated separately, instead of the commonly used concept of reconstruction. Note that slope propagation has also been used by Ben-Artzi and Falcovitz [3]. In order to propagate the slope, two CE are used to solve a one-dimensional conservation equation, because the variable u and its spatial derivative u_x are unknowns. Similarly three CEs are used for two-dimensional equations, because u, u_x and u_y are the unknowns, and four CEs are required for three-dimensional conservation equations. As shown by Chang et al. [7, 8, 73], triangles, tetrahedra, quadrilateral and hexahedral are the basic mesh stencils to construct the necessary CEs for two- and three-dimensional equa-

tions. The CE/SE method is a family of schemes, i.e., the a scheme for linear problems, the $a-\epsilon$ scheme, and the $a-\alpha$ scheme. The a scheme determines the space-time geometry of the numerical mesh employed. The $a-\epsilon$ and the $a-\alpha$ schemes are extensions of a scheme for nonlinear equations and shock capturing.

In this dissertation we also present a new multidimensional second order method for hyperbolic conservation laws. We refer to this method as the slope propagation method (SP-method). The main aim of our scheme is to retain the advantages of the both the CE/SE method for linear problems, the a scheme, and central schemes. We derive the scheme in a simple and straightforward way by using the basic concepts of finite volume schemes and conservation laws. The scheme uses space-time control volumes in order to compute the conservative flow variables and their slopes. In our scheme, unlike the CE/SE method, we do not assume the space-time linear variation of fluxes in each element. We assume instead the linear variation of the conservative flow variables only. The fluxes are calculated from the flow variables at the midpoint of the faces of the space-time control volumes. We approximate the time integrals of the fluxes by using the midpoint rule. This procedure eliminates the use of Jacobian matrices in our scheme. In the one-dimensional case our staggered mesh stencil is similar to that of central schemes [48] and Chang's method [6]. Note that Breuss [4] showed that a staggered central scheme produces less oscillations at local extrema in the data. In the 2D triangular mesh case we use the same staggered stencil which is used in the CE/SE method [7]. For linear equations, our scheme reduces to the a scheme of Chang [6] and differs from the central schemes since reconstructions are not used.

The main features of the our scheme are as follows:

(i) Space and time are treated in a unified manner.

(ii) The discrete space-time control volumes are the basic conservation regions.

(iii) The derivatives of the dependent variables are also treated as independent variables.

(iv) The mesh is staggered in time.

(v) A multidimensional scheme is reconstructed on triangular meshes.

(vi) The scheme enjoys the advantages of both central schemes and the CE/SE method.

(vii) The second order accuracy of the scheme is verified with numerical experiments.

1.1 A Overview of the Dissertation

In this dissertation, we extend the FVEG schemes from rectangular to triangular meshes. We start from the general theory of hyperbolic equations, see [41], described in Chapter 2 and recall the integral equations for a two-dimensional wave equation system using the

theory of bicharacteristics. We present the solution of the wave equation system using the approximate evolution operators for EG3 and EG4 schemes applied to structured triangular meshes. Moreover we give results about L^2-error for numerical experiments when exact and periodic boundary conditions are implemented. A second order approximation is made by using a linear reconstruction stage which uses three immediate neighbors of a triangle under consideration. Numerical algorithms for the first and second order schemes are also given. Further we extend the FVEG scheme to an irregular mesh. The contour plots for first and second order schemes are given for the individual components of the solution for irregular mesh. The numerical scheme is further extended to the nonlinear wave equation system [5].

The numerical implementation of EG schemes for triangular meshes is quite involved since these schemes consider infinite directions of wave propagation to compute fluxes. The fluxes are computed by using approximate evolution operator which involves integration along the circumference of a circle known as the sonic circle which is the base of the characteristic cone. The center of the circle lies at the quadrature point. The edge fluxes are approximated by using Simpson's rule. In this case our quadrature points are vertices and midpoints of the edges. Trapezoidal rule can also be used for this purpose which requires the approximate evolution operator to be applied only at the vertices. At a vertex the sonic circle is intersected by the neighboring triangles sharing that vertex. To consider the infinite directions of wave propagation lying inside the circle, one has to compute the angular contribution of these neighboring triangles. Similarly at the midpoint of the edges the sonic circle is intersected by two triangles sharing that edge. In this case also the angular contribution of these triangles is necessary to know for the application of approximate evolution operator. This means that a complete knowledge of the geometry of the mesh is required for EG schemes. This data include total number of vertex neighbors, their serial numbers, three immediate neighbors of each triangle, the elements sharing an edge, the normals to each edge, the centroids of all elements, the coordinates of midpoints, vertices, element areas, element vertex numbering, edge lengths, the angles subtended by neighboring triangles at the vertices and at the edges also. We have separated all our codes into two parts. The first part computes all the geometry data and writes it to a file. The second part reads that geometry file and implements the EG schemes. This reduces the computation time considerably.

In Chapter 3, FVEG schemes are applied to the linearized Euler equations both for regular and irregular triangular meshes. The approximate evolution operators for the linearized Euler equations have been given and the linearization of the Jacobian matrices is also explained. The numerical algorithms for the first and second order schemes for the linearized Euler equations are also given. Results for the advection wave equation system with first and second order approximations are presented for structured and unstructured triangular grids. Further we present some numerical tests which demonstrate the efficiency of the scheme.

The implementation of EG schemes for the linearized Euler equations is more complex than

the wave equation system. The reason is that due to the advection effect the center of the sonic circle is displaced and does not coincide with the vertex or the midpoint of the cell edge, therefore the computation of the angles becomes more complicated. The pressure, the velocities and the density are freezed locally. The local speed of sound is determined using these values. This means that some constant value is assigned to these variables outside the time loop and this value is same at all quadrature points. These values remain constant with time. The effect of this advection is that now the sonic circle is displaced from the quadrature point. The extent of this displacement depends upon the local velocities. The values of the local velocities determine whether the physical condition is subsonic or supersonic. In subsonic case the displacement of the sonic circle is small and it still intersects all the neighboring triangles both for vertex and edge midpoint. The angles subtended by the neighboring elements are required to be recalculated now. These angles are calculated with respect to a reference axis with its origin at the center of the displaced circle. For this purpose the points of intersections of the circle and elements edges are computed. In the sonic case the displacement of the circle center from the quadrature point is equal to the radius of the circle, hence the circle is tangent at that point. The computation of the angles is similar to the subsonic case. However there are various possible positions of the circle in this case and the program has to search for each possibility. Each possible position involves different vertex neighbors since in this case all neighboring elements of quadrature point are not intersected. The code has to sort out the right serial numbers of the triangles sharing the circle which is not easy. The supersonic case is even more complicated than the subsonic or sonic cases. Here now the displacement of the circle center is larger than its radius therefore the circle does not cross the quadrature point. Again there are many possible positions of the circle, each involving different number of neighboring elements of the point. In order to determine the angular contribution of different elements to the circle the code has to find the elements intersecting the circle.

In Chapter 4, we apply the FVEG schemes to nonlinear Euler equations. In this case also some kind of initial linearization is required. Here we compute local variables inside the time loop at each vertex and edge midpoint. This is carried out by using a weighted average of the piecewise constant values at the neigbouring elements of that point. We compute the length of the arc cut form the circle by an element. This length is considered as a weight for the piecewise constant value of that element. In this manner the values of the local variables are different at each vertex or midpoint and in each time cycle these values are computed from the current values. Since the local velocities also vary with time, the time step which is a function of these velocities is also variable i.e. for each time cycle we have a different time step. Note that in case of the linearized Euler equations the time step was constant because the local velocities were also constant. Furthermore the displacement of the sonic circle is different at each point. This displacement depends upon the values of the local velocities at that point. Likewise the angular contributions from the neighboring elements also vary at each point and each time cycle hence the angles and the number of neighbors intersected are updated with time. The code switches from subsonic to sonic or supersonic situation depending upon the values of the current local velocities at a par-

ticular point. The computational time increases considerably because those computations which were carried out once for the linearized Euler equations, are performed in each time cycle. A graphical picture of the three physical conditions and computation of the angles subtended by the neighbors is presented. For the second order scheme we use the same recovery that has been used for wave equation system and the linearized Euler equations. Numerical tests demonstrate the accuracy of the schemes.

In Chapter 5, we derive a new second order slope propagation method (SP-method) which combines the features of CE/SE methods and central schemes. We use space-time control volumes as the basic conservation regions. The flow variables and their slopes are the basic unknowns in the scheme. We assume linear variation of the flow variables with respect to space and time inside the control volumes. The number of control volumes required is equal to the number of unknowns. Unlike the CE/SE method of Chang [6], we do not assume linear variation of fluxes with respect to space and time. Therefore the present scheme is Jacobian-free as well as Riemann solver-free and hence can also be applied to any hyperbolic system. We derive this scheme from the conservation law for one-dimensional hyperbolic system as well as for a two-dimensional hyperbolic system based on a regular triangular mesh. Further this scheme gives a correct experimental order of convergence (EOC) in 1D and in 2D with triangular mesh. The scheme is efficiently implemented both for the wave equation system and the Euler equations in one and two space dimensions. Different numerical examples are presented and the accuracy of the scheme is compared with central and EG schemes.

Chapter 2

EG Schemes for the Wave Equation System

In this chapter we will develop schemes for the two-dimensional wave equation system based on triangular meshes. Evolution Galerkin schemes (EG) are truly multidimensional schemes developed by Ostkamp [52] and Lukáčová, Morton and Warnecke [41]. They used the general theory of linear hyperbolic systems to derive the approximate evolution operators for the two-dimensional first order wave equation system and Euler equations. In the above papers rectangular grids were used. The importance of the use of triangular grids arises from complex geometries where the boundaries are irregular and regular mesh cannot be constructed. In such cases triangles, quadrilaterals or other polygons can be used in two dimensions while tetrahedra can be used in three dimensions.

In Section 2.1 and 2.2, we will present the general theory of bicharacteristics for linear hyperbolic systems and recall the integral equations for the two-dimensional wave equation system, see [41]. In Section 2.3, we will discuss the finite volume evolution Galerkin scheme (FVEG) for triangular meshes. In Section 2.4 and 2.5 we discuss the first and second order EG schemes for the wave equation system and explain its numerical algorithm. For the second order scheme we discuss the recovery procedure. In Section 2.6 we recall the nonlinear wave equation [5] and explain the linearization of the coefficients as well as the numerical algorithms for the first and second order numerical schemes. For the nonlinear wave equation system we use a different method for the computation of slopes than the linear wave equation system. Finally we present some numerical experiments and their results for structured as well as unstructured triangular meshes. The results for structured and unstructured triangular meshes are equivalent to each other. We present our results for EG3 and EG4 schemes. EG schemes for triangular and rectangular meshes are compared.

2.1 General Theory

In this section we derive the integral equations for a general linear hyperbolic system. The general form of the linear hyperbolic system is given as

$$\mathbf{U}_t + \sum_{k=1}^{d} \mathcal{A}_k \mathbf{U}_{x_k} = 0, \;\; \mathbf{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d, \tag{2.1}$$

where the coefficient matrices $\mathcal{A}_k, k = 1, ..., d$ are elements of $\mathbb{R}^{p\times p}$ and the dependent variables are $\mathbf{U} = (u_1, ..., u_p)^T = \mathbf{U}(\mathbf{x}, t) \in \mathbb{R}^p$. Let $\mathcal{A}(\mathbf{n}) = \sum_{k=1}^{d} n_k \mathcal{A}_k$ be the **pencil matrix** where $\mathbf{n} = (n_1, ..., n_d)^T$ is a directional vector in $\mathbb{R}^d$. Since system (2.1) is hyperbolic then the matrix $\mathcal{A}(\mathbf{n})$ has p real eigenvalues λ_k, $k = 1, ..., p$ and p corresponding linearly independent right eigenvectors $\mathbf{r}_k = \mathbf{r}_k(\mathbf{n})$, $k = 1, ..., p$ and $\mathcal{R} = [\mathbf{r}_1|\mathbf{r}_2|...|\mathbf{r}_p]$ is the matrix composed of the right eigenvectors. To derive the integral equations, we use the concept of bicharacteristics. This requires to convert our general hyperbolic system (2.1) into a characteristic system. We define the characteristic variable $\mathbf{W} = \mathbf{W}(\mathbf{n})$ as $\partial\mathbf{W}(\mathbf{n}) = \mathcal{R}^{-1}\partial\mathbf{U}$. Since system (2.1) has constant coefficient matrices $\mathcal{A}_k$ we have $\mathbf{W} = \mathcal{R}^{-1}\mathbf{U}$ or $\mathbf{U} = \mathcal{R}\,\mathbf{W}$. Multiplying equation (2.1) by $\mathcal{R}^{-1}$ from the left we get

$$\mathcal{R}^{-1}\mathbf{U}_t + \sum_{k=1}^{d} \mathcal{R}^{-1}\mathcal{A}_k\mathcal{R}\,\mathcal{R}^{-1}\mathbf{U}_{x_k} = 0. \tag{2.2}$$

Let $\mathcal{B}_k = \mathcal{R}^{-1}\mathcal{A}_k\mathcal{R} = (b_{ij}^k)_{i,j=1}^{p}$, where $k = 1, 2, ..., d$ then equation (2.2) can be rewritten in the following form

$$\mathbf{W}_t + \sum_{k=1}^{d} \mathcal{B}_k \mathbf{W}_{x_k} = 0.$$

We decompose the matrix $\mathcal{B}_k$ into $\mathcal{B}_k = \mathcal{D}_k + \mathcal{B}'_k$, where $\mathcal{D}_k$ contains the diagonal part of the matrix $\mathcal{B}_k$, then we get

$$\mathbf{W}_t + \sum_{k=1}^{d} \mathcal{D}_k \mathbf{W}_{x_k} = -\sum_{k=1}^{d} \mathcal{B}'_k \mathbf{W}_{x_k} =: \mathbf{S}. \tag{2.3}$$

The i-th bicharacteristic corresponding to the i-th equation of (2.3) is defined by

$$\frac{d\mathbf{x}_i}{d\tilde{t}} = \mathbf{b}_{ii}(\mathbf{n}) = (b_{ii}^1, b_{ii}^2, ..., b_{ii}^d)^T,$$

where $i = 1, ..., p$. Here b_{ii}^k are the diagonal entries of the matrix $\mathcal{B}_k$, $k = 1, ..., d$, $i = 1, ..., p$. We consider the bicharacteristics backward in time. Therefore the initial conditions are $\mathbf{x}_i(t+\Delta t, \mathbf{n}) = \mathbf{x}$ for all $\mathbf{n} \in \mathbb{R}^d$ and $i = 1, ..., p$, i.e. $\mathbf{x}_i(\tilde{t}, \mathbf{n}) = \mathbf{x} - \mathbf{b}_{ii}(\mathbf{n})(t+\Delta t-\tilde{t})$. We will integrate the i-th equation of the system (2.3) from the point P down to the point $Q_i(\mathbf{n})$,

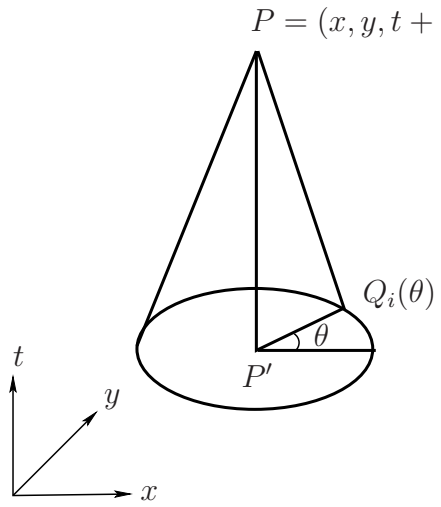

Figure 2.1: Bicharacteristics along the Mach cone through P and $Q_i(\mathbf{n})$, $d = 2$.

where the bicharacteristics hit the basic plane. Since the system is linear with constant coefficients, the bicharacteristics are straight lines in this case.
Now the i-th equation reads

$$\frac{\partial w_i}{\partial t} + \sum_{k=1}^{d} b_{ii}^k \frac{\partial w_i}{\partial x_k} = -\left(\sum_{j=1, i\neq j}^{d} \left(b_{ij}^1 \frac{\partial w_j}{\partial x_1} + b_{ij}^2 \frac{\partial w_j}{\partial x_2} + ... + b_{ij}^d \frac{\partial w_j}{\partial x_d} \right) \right) = S_i, \tag{2.4}$$

where $P \equiv (\mathbf{x}, t + \Delta t) \in \mathbb{R}^p \times \mathbb{R}_+$ is taken to be a fixed point, while $Q_i(\mathbf{n}) = (\mathbf{x}_i(\mathbf{n}, t), t) = (\mathbf{x} - \Delta t \mathbf{b}_{ii}, t)$. Taking a vector $\sigma_i = (b_{ii}^1, b_{ii}^2, ..., b_{ii}^d, 1)$, we can define the directional derivative

$$\frac{dw_i}{d\sigma_i} = \left(\frac{\partial w_i}{\partial x_1}, \frac{\partial w_i}{\partial x_2}, ..., \frac{\partial w_i}{\partial x_d}, \frac{\partial w_i}{\partial t} \right) \cdot \sigma_i = \frac{\partial w_i}{\partial t} + b_{ii}^1 \frac{\partial w_i}{\partial x_1} + b_{ii}^2 \frac{\partial w_i}{\partial x_2} + ... + b_{ii}^d \frac{\partial w_i}{\partial x_d}.$$

Hence the i-th equation (2.4) can be rewritten as follows

$$\frac{dw_i}{d\sigma_i} = S_i = -\sum_{j=1, i\neq j}^{d} \left(b_{ij}^1 \frac{\partial w_j}{\partial x_1} + b_{ij}^2 \frac{\partial w_j}{\partial x_2} + ... + b_{ij}^d \frac{\partial w_j}{\partial x_d} \right).$$

Now the integration from P to $Q_i(\mathbf{n})$ gives

$$w_i(P) - w_i(Q_i(\mathbf{n})) = S_i', \tag{2.5}$$

where

$$S_i' = \int_t^{t+\Delta t} S_i(\mathbf{x}_i(\tilde{t}, \mathbf{n}), \tilde{t}, \mathbf{n}) d\tilde{t} = \int_0^{\Delta t} S_i(\mathbf{x}_i(\tau, \mathbf{n}), t + \Delta t - \tau, \mathbf{n}) \mathrm{d}\tau \,.$$

Multiplication of equation (2.5) by $\mathcal{R}$ from the left and $(d-1)$-dimensional integration of the variable $\mathbf{n}$ over the unit sphere O in $\mathbb{R}^d$ leads to the integral representation

$$\mathbf{U}(P) = \mathbf{U}(\mathbf{x}, t+\Delta t) = \frac{1}{|O|}\int_O \mathcal{R}(\mathbf{n}) \begin{pmatrix} w_1(Q_1(\mathbf{n}),\mathbf{n}) \\ w_2(Q_2(\mathbf{n}),\mathbf{n}) \\ w_3(Q_3(\mathbf{n}),\mathbf{n}) \\ \vdots \\ w_p(Q_p(\mathbf{n}),\mathbf{n}) \end{pmatrix} \mathrm{d}O + \tilde{\mathbf{S}}, \tag{2.6}$$

where

$$\tilde{\mathbf{S}} = (\tilde{S}_1, \tilde{S}_2, ..., \tilde{S}_p)^T = \frac{1}{|O|}\int_O \mathcal{R}(\mathbf{n})\mathbf{S}'\mathrm{d}O = \frac{1}{|O|}\int_O \int_0^{\Delta t} \mathcal{R}(\mathbf{n})\mathbf{S}(t+\Delta t-\tau,\mathbf{n})\mathrm{d}\tau\mathrm{d}O\,,$$

and $|O|$ is the domain of integration.

2.2 Exact Integral Equations and Approximate Evolution Operators for the Wave Equation System

We will consider the two-dimensional wave equation system given as

$$\begin{aligned} \phi_t + c(u_x + v_y) &= 0\,, \\ u_t + c\,\phi_x &= 0\,, \\ v_t + c\,\phi_y &= 0\,, \end{aligned} \tag{2.7}$$

where c is a given constant. We recall here the exact integral equations derived in [41]. These are used to derive the so-called finite difference schemes EG1, EG2, EG3 and EG4, see [41] and [71] for more details on these schemes. Let $P = (x, y, t+\Delta t)$, $P' = (x, y, t)$, $Q = (x + c\Delta t\cos\theta, y + c\Delta t\sin\theta, t) = (\mathbf{x} + c\Delta t\mathbf{n}(\theta), t)$ and the so-called source term be given as

$$S = c\left[u_x \sin^2\theta - (u_y + v_x)\sin\theta\cos\theta + v_y\cos^2\theta\right]\,, \tag{2.8}$$

then the exact integral equations are given by

Exact Integral Equations:

$$\phi_P = \frac{1}{2\pi}\int_0^{2\pi} (\phi_Q - u_Q\,\cos\theta - v_Q\,\sin\theta)\,d\theta + \tilde{S}_1\,, \tag{2.9}$$

$$u_P = \frac{1}{2}u_{P'} + \frac{1}{2\pi}\int_0^{2\pi} (-\phi_Q\,\cos\theta + u_Q\,\cos^2\theta + v_Q\,\sin\theta\,\cos\theta)\,d\theta + \tilde{S}_2\,, \tag{2.10}$$

$$v_P = \frac{1}{2}v_{P'} + \frac{1}{2\pi}\int_0^{2\pi} (-\phi_Q\,\sin\theta + u_Q\,\cos\theta\,\sin\theta + v_Q\,\sin^2\theta\,d\theta + \tilde{S}_3, \tag{2.11}$$

where

$$\tilde{S}_1 = \frac{-1}{2\pi}\int_0^{2\pi}\int_0^{\triangle t} S(\mathbf{x}+c\tau\mathbf{n}(\theta), t+\Delta t-\tau, \theta)\,\mathrm{d}\tau\,\mathrm{d}\theta,$$

$$\tilde{S}_2 = \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\triangle t} \cos\theta S(\mathbf{x}+c\tau\mathbf{n}(\theta), t+\Delta t-\tau, \theta)\,\mathrm{d}\tau\,\mathrm{d}\theta$$

$$-\frac{1}{2\pi}\int_0^{2\pi}\int_0^{\triangle t} \left[c\phi_x(\mathbf{x}, t+\Delta t-\tau)\sin^2\theta - c\phi_y(\mathbf{x}, t+\Delta t-\tau)\sin\theta\cos\theta\right]\,\mathrm{d}\tau\,\mathrm{d}\theta,$$

$$\tilde{S}_3 = \frac{1}{2\pi}\int_0^{2\pi}\int_0^{\triangle t} \sin\theta S(\mathbf{x}+c\tau\mathbf{n}(\theta), t+\Delta t-\tau, \theta)\,\mathrm{d}\tau\,\mathrm{d}\theta$$

$$-\frac{1}{2\pi}\int_0^{2\pi}\int_0^{\Delta t} \left[c\phi_y(\mathbf{x}, t+\Delta t-\tau)\cos^2\theta - c\phi_x(\mathbf{x}, t+\Delta t-\tau)\sin\theta\cos\theta\right]\,\mathrm{d}\tau\,\mathrm{d}\theta\,.$$

To get an explicit formulation of the solution $\mathbf{U}^{n+1}$ at the point $P=(x,y,t^{n+1})$, we need to approximate the time integral from 0 to Δt in the above equations. This can be done by using suitable numerical quadrature. We use backward rectangle rule which gives us an $\mathcal{O}(\Delta t^2)$ approximation of the time integrals appearing in $\tilde{S}_1$, $\tilde{S}_2$ and $\tilde{S}_3$. Further we use the following result [41, Lemma 2.1]

$$\Delta t\int_0^{2\pi} S(t,\theta)\mathrm{d}\theta = \int_0^{2\pi}(u\cos\theta + v\sin\theta)\mathrm{d}\theta\,.$$

This gives the approximate evolution operator for the first variable ϕ. Other expressions for the time integrals of $S\cos\theta$ and $S\sin\theta$ can be used to get approximations for u and v, see [41]. In the above formula as well as similar formulae for $S\cos\theta$ and $S\sin\theta$, the derivatives of dependent variable in S have been replaced by the variables themselves. Hence the approximate evolution operators for EG3 and EG4 are given by

Approximate Evolution Operator for EG3:

$$\phi_P = \frac{1}{2\pi}\int_0^{2\pi}(\phi_Q - 2u_Q\cos\theta - 2v_Q\sin\theta)\mathrm{d}\theta + O(\Delta t^2)\,, \tag{2.12}$$

$$u_P = \frac{1}{2}u_{P'} + \frac{1}{2\pi}\int_0^{2\pi}(-2\phi_Q\cos\theta + u_Q(3\cos^2\theta - 1) + 3v_Q\sin\theta\cos\theta)\mathrm{d}\theta + O(\Delta t^2)\,, \tag{2.13}$$

$$v_P = \frac{1}{2}v_{P'} + \frac{1}{2\pi}\int_0^{2\pi}(-2\phi_Q\sin\theta + 3u_Q\sin\theta\cos\theta + v_Q(3\sin^2\theta - 1))\mathrm{d}\theta + O(\Delta t^2)\,. \tag{2.14}$$

Approximate Evolution Operator for EG4:

$$\phi_P = \frac{1}{2\pi}\int_0^{2\pi}(\phi_Q - 2u_Q\cos\theta - 2v_Q\sin\theta)\mathrm{d}\theta + O(\Delta t^2)\,, \tag{2.15}$$

$$u_P = \frac{1}{2\pi}\int_0^{2\pi}(-2\phi_Q\cos\theta + 2u_Q\cos^2\theta + 2v_Q\sin\theta\cos\theta)\mathrm{d}\theta + O(\Delta t^2)\,, \tag{2.16}$$

$$v_P = \frac{1}{2\pi}\int_0^{2\pi}(-2\phi_Q\sin\theta + 2u_Q\sin\theta\cos\theta + 2v_Q\sin^2\theta)\mathrm{d}\theta + O(\Delta t^2)\,. \tag{2.17}$$

2.3 Evolution Galerkin Schemes

For simplicity we consider two-dimensional case, i.e. $d = 2$. Let us consider $(T_i)_i$ to be a general unstructured triangular mesh in $\mathbb{R}^2$. We denote by $H^\kappa(\mathbb{R}^2)$ the Sobolov space of distributions with derivatives up to order κ in L^2 space, where $\kappa \in \mathbb{N}$. Consider the general hyperbolic system given by

$$\mathbf{U}_t + \sum_{k=1}^{d}\mathcal{A}_k\mathbf{U}_{x_k} = 0,\ \ \mathbf{x} = (x_1,\ldots,x_d)^T \in \mathbb{R}^d\,, \tag{2.18}$$

where the coefficient matrices $\mathcal{A}_k, k = 1,...,d$ are elements of $\mathbb{R}^{p\times p}$and the dependent variables are $\mathbf{U} = (u_1,...,u_p)^T = \mathbf{U}(\mathbf{x},t) \in \mathbb{R}^p$. Let us denote by $E(s) : (H^\kappa(\mathbb{R}^2))^p \to (H^\kappa(\mathbb{R}^2))^p$ the exact evolution operator for the system (2.1), i.e.

$$\mathbf{U}(.,t+s) = E(s)\mathbf{U}(.,t)\,. \tag{2.19}$$

We suppose that S_h^m is a finite element space consisting of piecewise polynomials of order $m \geq 0$ with respect to the triangular mesh. Assume constant time step, i.e. $t_n = n\Delta t$. Let $\mathbf{U}^n$ be an approximation in the space S_h^m to the exact solution $\mathbf{u}(.,t_n)$ at time $t_n \geq 0$. We consider $E_\tau : L^1_{loc}(\mathbb{R}^2) \to (H^\kappa(\mathbb{R}^2))^p$ to be a suitable approximate evolution operator for $E(\tau)$. In practice we will use restrictions of E_τ to the subspace S_h^m for $m \geq 0$. We denote by $R_h : S_h^m \to S_h^r$ a recovery stage, $r > m \geq 0$ and consider our approximate evolution operator E_τ on S_h^r. Further we will limit our considerations to the case where $m = 0$. Then we define finite volume evolution Galerkin scheme as:

Definition 2.1 *Starting from some initial data $\mathbf{U}^0 \in S_h^m$, the finite volume evolution Galerkin method (FVEG) is recursively defined by means of*

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{1}{|T_i|}\int_0^{\Delta t}\sum_{j=1}^{3}\left(n_{x_{s_{i_j}}}\mathbf{f}\left(\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}}\right) + n_{y_{s_{i_j}}}\mathbf{g}\left(\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}}\right)\right)d\tau\,, \tag{2.20}$$

where $n_{x_{s_{i_j}}}\mathbf{f}(\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}})$ and $n_{y_{s_{i_j}}}\mathbf{g}(\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}})$ represent approximations to the edge fluxes and $\mathbf{n}_{s_{i_j}} = (n_{x_{s_{i_j}}}, n_{y_{s_{i_j}}})^T$ is the outward normal to the side s_{i_j} and $|T_i|$ is the area of the

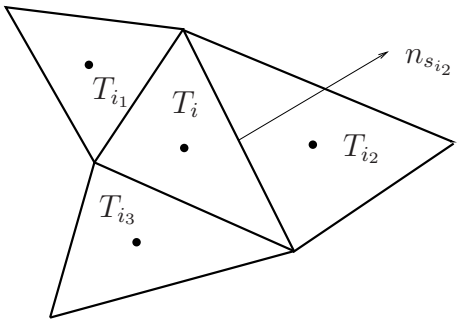

Figure 2.2: Stencil for the second order reconstruction.

element T_i, see Figure 2.2. The cell boundary value $\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}}$ is evolved using the approximate evolution operator E_τ to $t_n + \tau$ and integrated along the cell boundary, i.e.

$$\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}} = \int_{s_{i_j}} E_\tau R_h \mathbf{U}^n \, dL \,. \tag{2.21}$$

The integral in equation (2.21) is evaluated by means of some numerical quadratures such as Simpson's rule

$$\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}} = \frac{|\partial T_{i_j}|}{6} \left[E_\tau \, R_h \, \mathbf{U}^n(A) + 4 E_\tau \, R_h \mathbf{U}^n(M) + E_\tau \, R_h \, \mathbf{U}^n(B) \right] , \tag{2.22}$$

where A and B are the vertices and M is the middle point of the boundary ∂T_{ij} or the trapezoidal rule

$$\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{\tau}{\Delta t}} = \frac{|\partial T_{ij}|}{2} \left[E_\tau \, R_h \mathbf{U}^n(A) + E_\tau \, R_h \, \mathbf{U}^n(B) \right] . \tag{2.23}$$

Moreover when R_h is the identity mapping, i.e when we carry out no recovery then the approximate values $\mathbf{U}^n$ are piecewise constants. Considering these piecewise constants, the resulting schemes will only be of first order, even when E_τ is approximated to a higher order. Hence by inserting a recovery stage R_h before the evolution step in equation (2.21) we obtain higher order schemes. For more details on higher order finite volume evolution Galerkin schemes see [38], [43] and [45]. For other schemes on triangular meshes see [1, 17, 23, 24, 32, 62]. We approximate the time integral in Definition 2.1 by using the midpoint rule.

$$\int_{t^n}^{t^{n+1}} f(t) dt = \Delta t f(t^{n+\frac{1}{2}}) + O(\Delta t^3) \,.$$

Hence we obtain

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{|T_i|} \sum_{j=1}^{3} \left(\mathbf{f}(\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{1}{2}}) n_{x_{s_{i_j}}} + \mathbf{g}(\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{1}{2}}) n_{y_{s_{i_j}}} \right) .$$

Let $h_{\min} = \min_i h_i$, where $h_i = \min_{j=1,2,3} |h_{s_{i_j}}|$ and $|h_{s_{i_j}}|$ is the distance between the center of gravity of T_i and the midpoint of side s_{i_j}. The time step is computed using

$$\Delta t = \frac{h_{\min}\mathrm{CFL}}{c}\,.$$

Where CFL stands for the Courant, Friedrichs and Levy number [12]. The reason for making this choice of $h_{\min}$ is that the radius of the sonic circle must be smaller enough so that the circle is confined only to the immediate neighbors at the quadrature point.

2.4 First Order Schemes

In this section we give the first order numerical algorithms for finite volume evolution Galerkin schemes for the wave equation system. As we mentioned, using piecewise constant values $\mathbf{U}^n$, i.e. when no recovery is applied, the numerical schemes are always first order schemes. The algorithm for the first order finite evolution Galerkin scheme is given as:

- Input the initial data: $\phi_i^0,\ u_i^0,\ v_i^0$.
- Compute the time step $\Delta t = \mathrm{CFL}(\frac{h_{\min}}{c})$.
- Do the time loop
  1. Compute the intermediate values: $\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}},\ \tilde{u}_{s_{i_j}}^{n+\frac{1}{2}},\ \tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}$.
  2. Update:

$$\phi_i^{n+1} = \phi_i^n - \frac{c\Delta t}{|T_i|}\sum_{j=1}^{3}\left(n_{x_{s_{i_j}}}\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}} + n_{y_{s_{i_j}}}\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}\right), \tag{2.24}$$

$$u_i^{n+1} = u_i^n - \frac{c\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}}, \tag{2.25}$$

$$v_i^{n+1} = v_i^n - \frac{c\Delta t}{|T_i|}\sum_{j=1}^{3} n_{y_{s_{i_j}}}\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}}\,. \tag{2.26}$$

  3. Apply the boundary conditions.
- End the time loop.

The crucial step in this algorithm is to find the intermediate values $\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{1}{2}}$. Here we apply the approximate evolution operator. Let us put $\mathbf{U}^{n+\frac{1}{2}} = E_{\frac{\Delta t}{2}}\mathbf{U}^n$ then

$$\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{1}{2}} = \int_{s_{i_j}} \mathbf{U}^{n+\frac{1}{2}}\mathrm{d}L\,. \tag{2.27}$$

As we stated before, we approximate the integral in equation (2.27) either by using Simpson's rule or by means of the trapezoidal rule. This requires the evaluation of $\mathbf{U}^{n+\frac{1}{2}}$ at the two vertices as well as the midpoint of each side s_{i_j}, see Figure 2.3. Details on boundary conditions for structured triangular meshes are given in Appendix A. Consider the first

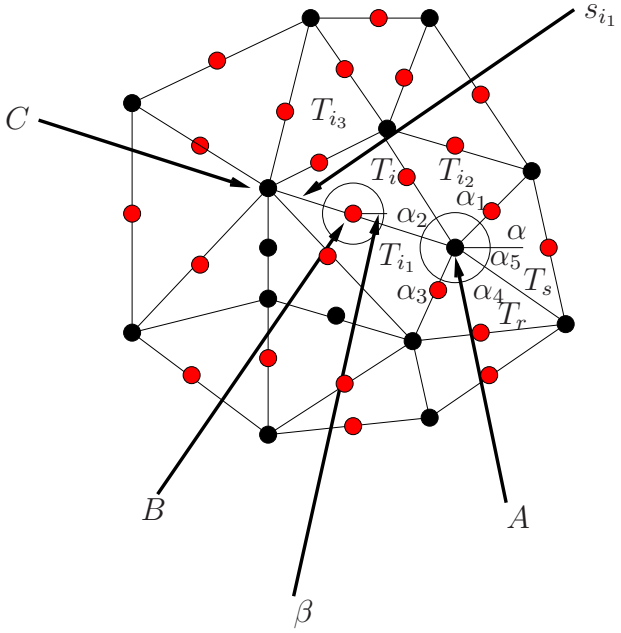

Figure 2.3: Stencil for computation of angles at quadrature points.

side of the element T_i, i.e the side s_{i_1}, then to evaluate the cone base integral required to determine $\mathbf{U}^{n+\frac{1}{2}}$ at the vertex A, we determine the angles α, α_1, α_2, α_3, α_4, α_5 where $\alpha+\alpha_5$ is the contribution from the element T_s, α_4 is the contribution from the element T_r and α_1, α_2, α_3 are the contributions from the elements T_{i_2}, T_i, T_{i_1} respectively. Then the integral from 0 to 2π is evaluated according to the sequence

$$0 \longrightarrow \alpha \longrightarrow \alpha_1+\alpha \longrightarrow \sum_{l=1}^{2}\alpha_l+\alpha \longrightarrow \sum_{l=1}^{3}\alpha_l+\alpha \longrightarrow \sum_{l=1}^{4}\alpha_l+\alpha \longrightarrow 2\pi\,.$$

The piecewise constant $\mathbf{U}^n$ is taken to be

$$\mathbf{U}^n = \begin{cases} \mathbf{U}_s^n & , 0 & < \theta < & \alpha, \\ \mathbf{U}_{i_2}^n & , \alpha & < \theta < & \alpha_1+\alpha, \\ \mathbf{U}_i^n & , \alpha_1+\alpha & < \theta < & \sum_{l=1}^{2}\alpha_l+\alpha, \\ \mathbf{U}_{i_1}^n & , \sum_{l=1}^{2}\alpha_l+\alpha & < \theta < & \sum_{l=1}^{3}\alpha_l+\alpha, \\ \mathbf{U}_r^n & , \sum_{l=1}^{3}\alpha_l+\alpha & < \theta < & \sum_{l=1}^{4}\alpha_l+\alpha, \\ \mathbf{U}_s^n & , \sum_{l=1}^{4}\alpha_l+\alpha & < \theta < & 2\pi\,. \end{cases}$$

To determine the intermediate value $\mathbf{U}^{n+\frac{1}{2}}$ at the midpoint we determine the angle β, see Figure 2.3. The cone base integral is evaluated according to the sequence

$$-\beta \longrightarrow \pi - \beta \longrightarrow 2\pi - \beta.$$

In this case the piecewise constant $\mathbf{U}^n$ is taken to be

$$\mathbf{U}^n = \begin{cases} \mathbf{U}_i^n & , -\beta < \theta < \pi - \beta, \\ \mathbf{U}_{i_1}^n & , \pi - \beta < \theta < 2\pi - \beta. \end{cases}$$

2.5 Second Order Schemes

To have second order schemes, a recovery stage is required to be carried out before the evolution step. A recovery method for triangular meshes is given in the book by Sonar [61] where two edge neighbors of a triangle are used to compute the slopes. We use the same linear recovery given as follows

$$R_h \mathbf{U}^n \Big|_{T_i} = \mathbf{U}_c + \mathbf{U}_x (x - x_{c_i}) + \mathbf{U}_y (y - y_{c_i}), \tag{2.28}$$

where (x_{c_i}, y_{c_i}) is the centroid of the element T_i. The coefficients $\mathbf{U}_c$, $\mathbf{U}_x$ and $\mathbf{U}_y$ are determined such that

$$\frac{1}{|T_{ij}|} \int_{T_{i_j}} R_h \mathbf{U}^n \mathrm{d}x.dy = \mathbf{U}_{i_j}^n \,, \quad j = 1, 2, 3 \,.$$

Where T_{i_j} are the three close neighbors to the cell T_i, i.e. the neighbors having one edge in common with it, see Figure 2.2. To maintain conservativity, we adjust the recovery such that it preserves cell averages.

2.5.1 Recovery with Three Neighbors

Consider the recovery given by equation (2.28). We use three neighbors to compute the slopes. In [61] only two neighbors have been used, however we have modified this method by using three edge neighbors to compute flux slopes. This method gives slightly better results than the one with two neighbors. Taking averages over the *ith* cell neighbors gives

$$\mathbf{U}_{i_1} = \mathbf{U}_c + \mathbf{U}_x \frac{1}{|T_{i_1}|} \int_{T_{i_1}} (x - x_{c_i}) dx\, dy + \mathbf{U}_y \frac{1}{|T_{i_1}|} \int_{T_{i_1}} (y - y_{c_i}) dx\, dy \,, \tag{2.29}$$

$$\mathbf{U}_{i_2} = \mathbf{U}_c + \mathbf{U}_x \frac{1}{|T_{i_2}|} \int_{T_{i_2}} (x - x_{c_i}) dx\, dy + \mathbf{U}_y \frac{1}{|T_{i_2}|} \int_{T_{i_2}} (y - y_{c_i}) dx\, dy \,, \tag{2.30}$$

$$\mathbf{U}_{i_3} = \mathbf{U}_c + \mathbf{U}_x \frac{1}{|T_{i_3}|} \int_{T_{i_3}} (x - x_{c_i}) dx\, dy + \mathbf{U}_y \frac{1}{|T_{i_3}|} \int_{T_{i_3}} (y - y_{c_i}) dx\, dy\,. \tag{2.31}$$

This implies that

$$\mathbf{U}_{i_1} = \mathbf{U}_c + \mathbf{U}_x (x_{c_{i_1}} - x_{c_i}) + \mathbf{U}_y (y_{c_{i_1}} - y_{c_i}) \tag{2.32}$$

$$\mathbf{U}_{i_2} = \mathbf{U}_c + \mathbf{U}_x (x_{c_{i_2}} - x_{c_i}) + \mathbf{U}_y (y_{c_{i_2}} - y_{c_i})\,, \tag{2.33}$$

$$\mathbf{U}_{i_3} = \mathbf{U}_c + \mathbf{U}_x (x_{c_{i_3}} - x_{c_i}) + \mathbf{U}_y (y_{c_{i_3}} - y_{c_i})\,. \tag{2.34}$$

Solving this system of algebraic equations, we get the coefficients $\mathbf{U}_c$, $\mathbf{U}_x$ and $\mathbf{U}_y$. Finally to preserve the conservativity property we take $\mathbf{U}_c$ equal to $\mathbf{U}_i = \frac{1}{|T_i|} \int_{T_i} R_h \mathbf{U}^n \, dxdy$. Another procedure for the second order reconstruction is the least squares method, given in [50] and [14]. This method gives equivalent results to one we discussed. For other reconstruction methods see [62].

2.5.2 Numerical Algorithm

Now the Numerical algorithm for the second order finite evolution Galerkin scheme is given as:

- Input the initial data: ϕ_i^0, u_i^0, v_i^0.
- **Carry out the recovery stage**.
- Compute the time step Δt.
- Do the time loop
  1. Compute the intermediate values: $\tilde{\phi}_{s_{ij}}^{n+\frac{1}{2}}$, $\tilde{u}_{s_{ij}}^{n+\frac{1}{2}}$, $\tilde{v}_{s_{ij}}^{n+\frac{1}{2}}$.
  2. Update: Equations (2.24)-(2.26).
  3. **Carry out the recovery stage**.
  4. Apply the boundary conditions.
- End the time loop.

Note that in this algorithm as well as in the first order one, we have used the boundary conditions described in Appendix A.

2.6 Nonlinear Wave Equation System

In the previous algorithms we have numerically solved the linear wave equation system. There are various phenomena which can be modeled by the nonlinear wave equation system [5]. In this case nonlinear effects cannot be ignored. If these effects are taken into consideration then jumps in solutions (shocks) can occur. In fluid dynamics these nonlinear effects create two main problems, instabilities and generation of new wave modes through nonlinear wave interactions. Many physical processes such as phase changes are nonlinear. In the Navier-Stokes equations the most significant nonlinear term is the advection term. However different nonlinear phenomena are modeled by different equations such as Euler equations, shallow water equations and Burgers equation. In most of these equations the nonlinearity is the quadratic nonlinearity. In the linear problem, all the points of the wave move at a constant speed and the shape of the wave does not change, however for a nonlinear wave, different parts of the wave advect themselves according to their local speeds such that the local speed depends upon the wave amplitude and hence the shape of the wave changes with time. This process soon leads to a shock. In this section we discuss the nonlinear wave equation system which describes the compressible flow. This system has been derived in [5] from the inviscid system of compressible Euler equations of gas dynamics by neglecting the inertial terms. The equation is well posed in space and time and have a nonlinear acoustic-wave dependence similar to gas dynamic equations. In addition the equation can be changed to second order quasi linear self similar equation at the sonic line where it becomes a parabolic equation. It offers a more realistic model for gas dynamics. Note that the pressure-gradient equations derived by Zhang et al., see [36] and [72] for details, are similar to nonlinear wave equations. The compressible Euler equations for isentropic flow in two space dimensions are as follows.

$$\begin{aligned} \rho_t + (\rho u)_x + (\rho v)_y &= 0\,, \\ (\rho u)_t + (\rho u^2 + p)_x + (\rho uv)_y &= 0\,, \\ (\rho v)_t + (\rho uv)_x + (\rho v^2 + p)_y &= 0\,, \end{aligned} \tag{2.35}$$

where ρ u, v and $p(\rho)$ stand for the density, the components of velocity and the pressure respectively. The pressure and density are related through the relation $p(\rho) = A\rho^\gamma$, where $\gamma > 1$ is the ratio of specific heats. The local speed of sound is $c^2 = \frac{dp}{d\rho}$. If we neglect the quadratic terms in u and v in system (2.35) and substitute $f = \rho u$ and $g = \rho v$, $\frac{dp}{dx} = c^2\rho_x$ and $\frac{dp}{dy} = c^2\rho_y$ then we obtain the following nonlinear wave equation system.

$$\begin{aligned} \rho_t + f_x + g_y &= 0\,, \\ f_t + c^2\rho_x &= 0\,, \\ g_t + c^2\rho_y &= 0\,, \end{aligned} \tag{2.36}$$

where ρ, f and g are conservative variables. We linearize this equation by freezing the pressure, density and speed of sound locally. The linearized coefficients are denoted by p',

ρ' and c'. Substituting $\phi = c'\,\rho$, we obtain the linearized wave equation as follows

$$\begin{aligned} \phi_t + c'(f_x + g_y) &= 0\,, \\ f_t + c'\phi_x &= 0\,, \\ g_t + c'\phi_y &= 0\,. \end{aligned} \tag{2.37}$$

The above equation resembles the wave equation system (2.7) and we can use the evolution operators (2.12)-(2.17) to calculate the fluxes. Linearization can be done in two ways; linearization of the coefficients at each element and linearization of the coefficients at each quadrature point which includes the vertices and midpoints of the edges of the triangle. In the former method the local variables have constant value at every element in a particular time cycle which is computed by taking simple average of the values from the edge neighbors of the element. In the later method coefficients are freezed at vertices and midpoints of the edges. In this case the values of the local variable for a vertex are computed by taking simple average of the values at the neighbouring elements sharing a vertex while for the midpoint of the edges, the elements sharing the edge are used. We have used the former method in the first order scheme and later method in the second order scheme. In the first order scheme both methods work well. For the second order method however linearization at each quadrature point is required to compute the intermediate values $\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{f}_{s_{i_j}}^{n+\frac{1}{2}}$ and $\tilde{g}_{s_{i_j}}^{n+\frac{1}{2}}$.

2.6.1 First Order Schemes

The numerical algorithms for the first order as well as the second order EG schemes for the nonlinear wave equation system are similar to the algorithms for the linear wave equation system given in Section 2.4 and 2.5. The solution at each quadrature point has been approximated by taking the piecewise constant value $\mathbf{U}^n$ from the close neighbors. The contribution from each neighbor is taken according to the angular sector cut by the base of the Mach cone. In the nonlinear wave equation system, our primitive variables are ρ, u, v and p while the conservative variables are ρ, f and g. The time step is computed using

$$\Delta t = \frac{\mathrm{CFL} h_{\min}}{c'}\,.$$

Since the local speed of sound is computed in each time cycle using the current data, therefore time step is variable. The numerical algorithm for the first order finite volume evolution Galerkin scheme for the nonlinear wave equation system reads:

- Input the initial data:
  - The primitive variables ρ_i^0, u_i^0, v_i^0, p_i^0.
  - Compute ϕ_i^0 and the conservative variables f_i^0, g_i^0.
- Do the time loop

1. Compute the local variable c'.
2. Compute the time step Δt.
3. Compute the intermediate values: $\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{f}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{g}_{s_{i_j}}^{n+\frac{1}{2}}$.
4. Update:

$$\phi_i^{n+1} = \phi_i^n - \frac{c'\Delta t}{|T_i|}\sum_{j=1}^{3}\left(n_{x_{s_{i_j}}}\tilde{f}_{s_{i_j}}^{n+\frac{1}{2}} + n_{y_{s_{i_j}}}\tilde{g}_{s_{i_j}}^{n+\frac{1}{2}}\right), \tag{2.38}$$

$$f_i^{n+1} = f_i^n - \frac{c'\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}}, \tag{2.39}$$

$$g_i^{n+1} = g_i^n - \frac{c'\Delta t}{|T_i|}\sum_{j=1}^{3} n_{y_{s_{i_j}}}\tilde{\phi}_{s_{i_j}}^{n+\frac{1}{2}}. \tag{2.40}$$

5. Compute the primitive variables ρ_i, u_i, v_i, p_i.
6. Apply the boundary conditions.

- End the time loop.

Note that the updated values of the primitive variables ρ and p have been used to determine the local speed of sound $c' = \sqrt{\frac{\gamma p'}{\rho'}}$, ϕ, f and g. Moreover the updated conservative variables have been used in the approximate evolution operator to determine the intermediate values $\tilde{\phi}_{s_{ij}}^{n+\frac{1}{2}}$, $\tilde{f}_{s_{ij}}^{n+\frac{1}{2}}$ and $\tilde{g}_{s_{ij}}^{n+\frac{1}{2}}$.

2.6.2 Second Order Schemes

In this subsection we present the second order scheme for the nonlinear wave equation system. We will use linear recovery of the form (2.28). The method stated in Section 2.5 for the determination of the slopes gives oscillation in in case of the nonlinear wave equation system. Therefore we have used a central difference type reconstruction approach used by Zhang et al. in [73]. This procedure automatically limits the slopes. However this is not a min-mod limiter. Let $(x_{c_{i_j}}, y_{c_{i_j}})$, $j = 1, 2, 3$ denote the coordinates of the centroids of the edge neighbors of the element T_i and let (x_{c_i}, y_{c_i}) be the coordinates of the centroid of T_i. Set

$$\Delta x_j = x_{c_{i_j}} - x_{c_i}\,, \quad \Delta y_j = y_{c_{i_j}} - y_{c_i}\,, \qquad \Delta \mathbf{U}^{(j)} = \mathbf{U}_{i_j} - \mathbf{U}_i\,, \;\; j = 1, 2, 3.$$

Then the slopes are given by

$$\mathbf{U}_x{}^{(i)} = \frac{1}{3}\sum_{j=1}^{3}\mathbf{U}_x{}^{(j)}, \qquad \mathbf{U}_y{}^{(i)} = \frac{1}{3}\sum_{j=1}^{3}\mathbf{U}_y{}^{(j)}\,,$$

with

$$\mathbf{U}_x{}^{(j)} = \frac{D_x{}^{(j)}}{D}, \qquad \mathbf{U}_y{}^{(j)} = \frac{D_y{}^{(j)}}{D},$$

where

$$D := \begin{vmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \end{vmatrix}, \qquad D_x^{(1)} := \begin{vmatrix} \Delta \mathbf{U}^{(1)} & \Delta y_1 \\ \Delta \mathbf{U}^{(2)} & \Delta y_2 \end{vmatrix}, \quad D_y^{(1)} := \begin{vmatrix} \Delta x_1 & \Delta \mathbf{U}^{(1)} \\ \Delta x_2 & \Delta \mathbf{U}^{(2)} \end{vmatrix}, \tag{2.41}$$

$$D_x^{(2)} := \begin{vmatrix} \Delta \mathbf{U}^{(2)} & \Delta y_2 \\ \Delta \mathbf{U}^{(3)} & \Delta y_3 \end{vmatrix}, \quad D_y^{(2)} := \begin{vmatrix} \Delta x_2 & \Delta \mathbf{U}^{(2)} \\ \Delta x_3 & \Delta \mathbf{U}^{(3)} \end{vmatrix}, \tag{2.42}$$

$$D_x^{(3)} := \begin{vmatrix} \Delta \mathbf{U}^{(3)} & \Delta y_3 \\ \Delta \mathbf{U}^{(1)} & \Delta y_1 \end{vmatrix}, \quad D_y^{(3)} := \begin{vmatrix} \Delta x_3 & \Delta \mathbf{U}^{(3)} \\ \Delta x_1 & \Delta \mathbf{U}^{(1)} \end{vmatrix}. \tag{2.43}$$

Moreover to eliminate spurious oscillations further, one can modify by a re-weighting procedure as

$$\mathbf{U}_x{}^{(i)} = \frac{\sum_{j=1}^{3} W^{(j)} \mathbf{U}_x{}^{(j)}}{\sum_{j=1}^{3} W^{(j)}}, \quad \mathbf{U}_y{}^{(i)} = \frac{\sum_{j=1}^{3} W^{(j)} \mathbf{U}_y{}^{(j)}}{\sum_{j=1}^{3} W^{(j)}},$$

where

$$W^{(j)} = \prod_{l=1, l\neq j}^{3} \theta_l, \text{ and } \quad \theta_l = \sqrt{(\mathbf{U}_x{}^{(l)})^2 + (\mathbf{U}_y{}^{(l)})^2}.$$

Now the numerical algorithm for the second order finite volume EG scheme for the nonlinear wave equation resembles the first order algorithm except for the recovery stage which is carried out before the evolution step.

2.7 Numerical Experiments

Example 2.1

We consider the two-dimensional wave equation system (2.7) together with the initial data

$$\phi(\mathbf{x}, 0) = -\frac{1}{c}(\sin 2\pi x + \sin 2\pi y), \;\; u(\mathbf{x}, 0) = 0 = v(\mathbf{x}, 0).$$

In this case the exact solution is

$$\begin{aligned} \phi(\mathbf{x}, t) &= -\frac{1}{c}\cos(2\pi ct)(\sin(2\pi x) + \sin(2\pi y)), \\ u(\mathbf{x}, t) &= \frac{1}{c}\sin(2\pi ct)\cos(2\pi x), \\ v(\mathbf{x}, t) &= \frac{1}{c}\sin(2\pi ct)\cos(2\pi y). \end{aligned}$$

Where c is a given constant. We take $c = 1$ and consider a computational domain $\Omega = [-1, 1] \times [-1, 1]$. If N_x and N_y are the number of rectangular cells along x and y-axis respectively in the regular triangular mesh then $2N_xN_y$ will be the number of triangular cells in the mesh. We denote the total number of triangular cells by N. In this experiment we used end time $T = 0.2$ and CFL $= 0.4$. Table 2.3 shows the L^2-error between the exact and the approximated solutions using the first order EG4 scheme. The first column indicates the total number of triangular cells in the mesh. The next four columns show the L^2-error with individual components ϕ, u, v and the combined error respectively. The last column represents the so-called experimental order of convergence (EOC), which is defined with respect to the L^p-norm

$$\text{EOC} = \ln\left[\frac{||\mathbf{u} - \mathbf{U}_h^n||_{L^p}}{||\mathbf{u} - \mathbf{U}_{\frac{h}{2}}^n||_{L^p}}\right] / \ln(2), \quad \text{for } (1 < p < \infty).$$

where $\frac{h}{2}$ and h are the two consecutive mesh sizes. Moreover L^1 and L^2-error are given as

$$||\mathbf{u} - \mathbf{U}_h^n||_{L^1} = \sum_{i=1}^{N} |T_i||\mathbf{u} - \mathbf{U}_h^n| \text{ and } \quad ||\mathbf{u} - \mathbf{U}_h^n||_{L^2} = \sqrt{\sum_{i=1}^{N} |T_i||\mathbf{u} - \mathbf{U}_h^n|^2}\,.$$

In Appendix B we show results of first order EG schemes with periodic boundary conditions and with two different types of structured triangular meshes. Further Tables 2.1 and B.2 present the L^2-error for EG3 scheme with exact and periodic boundary conditions respectively.

The plots for second order numerical schemes are shown in Figures 2.5 and 2.7. We also present some results for an unstructured triangular mesh and compare it with the results obtained from a structured mesh. Tables 2.7 and 2.8 show the L^2-error for an unstructured mesh and a structured mesh with comparable number of total elements because it is difficult to get exactly equal number of elements for a structured and an unstructured mesh. Note that the first column stands for the total number of elements in the mesh. For unstructured meshes we implement exact boundary conditions.

2.7.1 Discussion of First Order Results

First order results for EG schemes are presented in Tables 2.1 and 2.3. We see from these tables that the error in the component ϕ is half of the error in u and v when exact boundary conditions were used while for periodic boundary conditions, see Tables B.2 and B.1, the error in ϕ slightly exceeds that of u and v. The reason for this behavior of ϕ may be due to the fact that ϕ is a function of both x and y while u and v are functions of one coordinate, that is either x or y. Therefore with periodic boundary conditions the accuracy for ϕ is less than the other two components. The last column indicates that an EOC of 1 is achieved successfully for both EG3 and EG4 schemes which implies that both schemes

are of first order when no recovery is carried out. In Figure 2.4 and 2.6 top we plot the numerical solution of ϕ along the line $y = 0$ using the first order EG3 and EG4 schemes. We considered meshes consisting of $2 \times 40 \times 40$ and $2 \times 160 \times 160$ number of elements. At the bottom we plot the numerical solution of u. These plots clearly indicate that the developed schemes predict the smooth waves in a good manner. The accuracy of the EG3 is slightly better than the EG4 scheme. Te results of first order EG schemes with periodic periodic boundary conditions are given in Tables B.1, B.2, B.5 and B.6. These results indicate that such boundary conditions work well for triangular meshes.

We have applied the EG schemes to the two types of regular meshes shown in Figure A.1. It is clear from these results that we obtain more accurate results with mesh 2 as compared to mesh 1. A possible explanation to these results is that, mesh 2 is more uniform as compared to mesh 1 since the number of neighboring elements at each vertex remain same i.e. 6 elements for mesh 2. For mesh 1 there are two types of vertices, namely vertices with 4 neighboring triangles and vertices with 8 neighbors. For symmetric initial data such as those in Example 2.7, the uniformity of the mesh also contributes to the accuracy of the solution to some extent. This behavior is evident in both first and second order EG3 and EG4 schemes. In Tables 2.5 and 2.6, we compare the results for first order EG4 scheme on an unstructured mesh as well as on structured mesh 2. We employ, as boundary conditions, the values of the known exact solution. It is clear that for an irregular mesh the error in the individual components of the solution and the total error are comparable with that of a regular mesh. Moreover the EOC in the last column shows that the first order EG schemes have equally good results for an irregular mesh. However one important thing is the symmetry of the function u and v which is reflected in the error also, has not been maintained for irregular meshs. For a regular mesh the error in u and v is symmetric upto the last digit. This effect is obvious due to the non-uniformity of the mesh. The isolines of the solution components ϕ, u and v for the first order EG4 scheme are shown in Figure 2.8 with an unstructured mesh. The isolines of ϕ consist of small circular waves distributed regularly in the region. The plot for the component u consists of vertical lines while that of v contains horizontal lines showing the dependence of u on x coordinate and v on y coordinate.

We have also compared first order EG4 scheme on rectangular and structured triangular meshes. The L^2-error is shown in Table 2.9. We take N_x number of rectangles along x-axis and N_y number of rectangles along y-axis. Then the number of triangles in the triangular mesh are taken as $2N_xN_y$. Here N_R represents the total number of rectangles and N represents the total number of triangles in the mesh. In Figure 2.9, we plot the components ϕ and u on rectangular and triangular meshes for 50×50 rectangles or $2 \times 50 \times 50$ triangles after $T = 0.2$. Both the L^2-errors and plots agree with each other for both types of meshes.

2.7.2 Second Order Results and Convergence Problems

Tables 2.2 and 2.4 show the L^2-error and the EOC results for the second order EG schemes. If we compare both these tables, then the error for the second order EG4 scheme is 9 times smaller than that of the first order scheme for the same number of mesh points. Similar behavior can be seen with the EG3 scheme, see Tables 2.1 and 2.2. Further note that EOC equal to 2 is not achieved although we couple a recovery step with the evolution step in the second order scheme. However the plots for the second order EG schemes demonstrate improved accuracy and resolution over the first order schemes, see Figures 2.5 and 2.7. We have also tested other kinds of recoveries for the second order schemes including least squares recovery [14] and recovery using two neighbors. However the results are not included in this thesis. The accuracy of least squares recovery is comparable with that described in Section 2.5. While the the reconstruction using two edge neighbors of triangle gives slightly more disspation. However none of then gives the correct EOC. In Tables B.13-B.15, we present the L^1-error for the second order EG4 scheme with CFL = 0.1, 0.2 and 0.4 respectively. We notice that with CFL = 0.1, the EOC is better however as we increase the CFL number, the EOC values drop. Another aspect of this convergence behavior is that when the mesh size increases, the EOC decreases gradually both for EG3 and EG4 schemes. This can be seen in Tables 2.2 and 2.4. The reasons for this kind of convergence behavior of EG schemes for triangular meshes are yet to be explored.

In the above results we have used Simpson's rule for computing edge flux integrals. We have also tested the trapezoidal rule. In the later case we need to apply the evolution operator at the vertices of the triangle to obtain $\tilde{\mathbf{U}}_{s_{i_j}}^{n+\frac{1}{2}}$, while in the case of Simpson's rule we need these computations at the midpoints of the edges as well as at the vertices. In Tables B.9-B.12, we present results for EG schemes with trapezoidal rule. These results show that for both EG3 and EG4 schemes, the L^2-error is considerably greater than the one we have obtained when Simpson's rule was used. This means that for EG schemes when used together with triangular meshes, Simpson's rule provides a better approximation of the edge fluxes as compared to the trapezoidal rule. In fact in a single triangle Simpson's rule uses information at 6 points, three vertices and 3 midpoints, while the trapezoidal rule uses information at 3 points only. Figures 2.5 and 2.7 show the one-dimensional profiles for the components ϕ and u along the line $y = 0$ for the second order EG4 and EG3 schemes respectively. At the top ϕ is plotted for two meshes consisting of $2\times40\times40$ and $2\times160\times160$ elements. It is evident that with the increase of mesh size the numerical solution approaches much closer to the exact solution and almost completely overlaps it. At the bottom we plot u for similar mesh sizes. Here also the exact solution is in good agreement with the numerical solution. If we compare these plots with the first order schemes, see Figures 2.4 and 2.6, we notice a clear improvement of accuracy in the second order schemes. In Tables 2.7 and 2.8 we compare the second order results for the EG4 scheme for an unstructured and a structured mesh with approximately the same number of mesh points. The L^2-error for individual solution components and the total error is comparable with that of structured meshes. The non-symmetry of u and v in L^2-error can also be seen for the second order

scheme. The convergence behavior of the EG schemes for an unstructured mesh therefore is similar to that of a structured mesh. From Tables 2.5 and 2.7, we can see that the error for second order EG schemes is 11 times smaller than that of first order for an unstructured mesh. This suggests that apart from convergence difficulties, the accuracy of the second order EG schemes increases considerably than the first order and the schemes are suitable for irregular meshes. Isolines of the solution components for the second order EG4 scheme with irregular mesh are shown in Figure 2.8 for 80899 number of mesh points. The second order scheme gives better resolution in the plots and spurious oscillations and noise are diminished considerably.

Example 2.2

We consider the nonlinear wave equation system (2.37) together with the following initial data

$$(\rho, u, v, p) = \begin{cases} (1, 0, 0, 1) & \text{if } (x^2 + y^2) < 0.16 \\ (0.125, 0, 0, 0.1) & \text{otherwise}\,. \end{cases}$$

The above problem has been given in [64]. We take the computational domain to be $\Omega = [-1, 1] \times [-1, 1]$, the CFL $= 0.4$ and the absolute time $T = 0.1$. Furthermore we keep $A = 2$ and $\gamma = 1.4$. We consider a structured triangular mesh labelled as mesh 2 in Figure A.1 for this test. Note that in the above initial data, the pressure inside a circular region of radius 0.4 is greater than the outside pressure. Similarly there is a difference of density inside and outside the circular region initially. The density and pressure gradients set a circular shock wave which moves outward towards the periphery of the region and a rarefaction wave starts moving inside into the high density region. Figures 2.10 and 2.11 show the profiles of the density, velocities and pressure for $2 \times 200 \times 200$ mesh elements for the first and the second order EG4 schemes. We have used extrapolated boundary conditions, see Appendix A for boundary conditions on structured triangular meshes. The second order scheme has some oscillations in the solution. The plots show the interaction of nonlinear waves, the propagation of a shock and its impact on the profiles of flow variables.

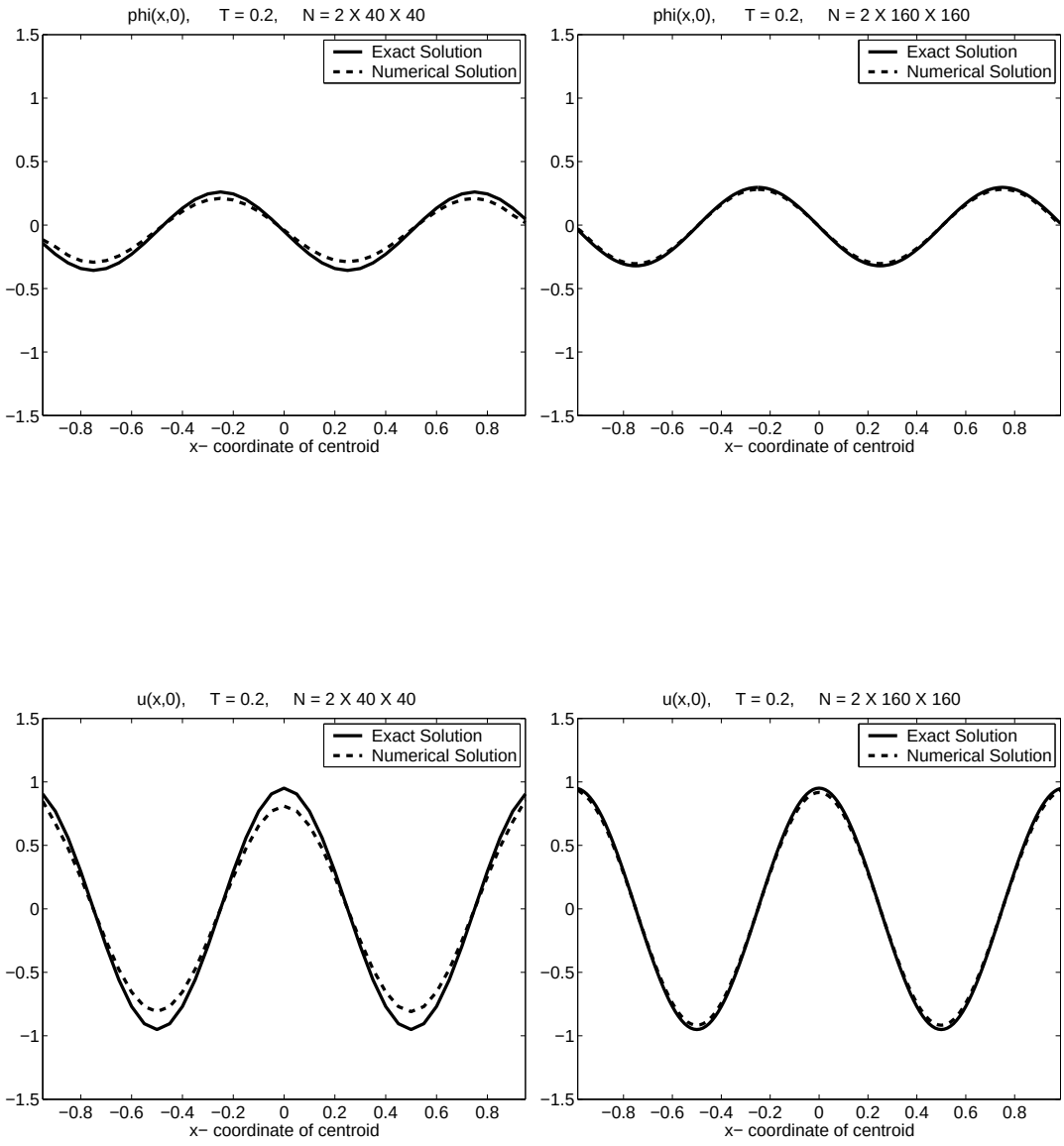

Figure 2.4: First order EG3 scheme with exact boundary conditions.

Table 2.1: First order EG3 scheme (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.1513202875	0.3098260720	0.3098260720	0.4635539011	
$2\times 40\times 40$	0.1016266255	0.1792780749	0.1792780749	0.2731468970	0.76
$2\times 80\times 80$	0.0584809029	0.0961872890	0.0961872890	0.1480675696	0.88
$2\times 160\times 160$	0.0312883404	0.0497925600	0.0497925600	0.0770555534	0.94
$2\times 320\times 320$	0.0162095376	0.0253378738	0.0253378738	0.0393289309	0.97

Table 2.2: Second order EG3 scheme (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.0388933622	0.0290612690	0.0290612690	0.0565845239	
$2\times 40\times 40$	0.0188297133	0.0125462456	0.0125462456	0.0258722760	1.13
$2\times 80\times 80$	0.0094591803	0.0060458639	0.0060458639	0.0127507268	1.02
$2\times 160\times 160$	0.0043670276	0.0029125732	0.0029125732	0.0060030905	1.10
$2\times 320\times 320$	0.0020982338	0.0014358420	0.0014358420	0.0029199092	1.04

Table 2.3: First order EG4 scheme (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.1556852847	0.3106859459	0.3106859459	0.4661431345	
$2\times 40\times 40$	0.1031277917	0.1795746993	0.1795746993	0.2740975861	0.77
$2\times 80\times 80$	0.0588980225	0.0962762052	0.0962762052	0.1483482135	0.89
$2\times 160\times 160$	0.0313897503	0.0498186003	0.0498186003	0.0771304240	0.95
$2\times 320\times 320$	0.0162095376	0.0253378738	0.0253378738	0.0393289309	0.98

Table 2.4: Second order EG4 scheme (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.0298919468	0.0315855121	0.0315855121	0.0537477222	
$2\times 40\times 40$	0.0165818142	0.0130849813	0.0130849813	0.0248473346	1.11
$2\times 80\times 80$	0.0089848325	0.0062160476	0.0062160476	0.0125700323	0.98
$2\times 160\times 160$	0.0042790672	0.0029796329	0.0029796329	0.0060055675	1.07
$2\times 320\times 320$	0.0020943985	0.0014642937	0.0014642937	0.0029453043	1.03

Table 2.5: First order EG4 scheme with unstructured mesh.

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
1264	0.1374568482	0.2574950398	0.2541604479	0.3870343834
5056	0.0859348871	0.1449409851	0.1425261079	0.2206952320
20224	0.0477021186	0.0767743899	0.0753626372	0.1176831600
80896	0.0243922168	0.0388246755	0.0380538406	0.0595854886
323584	0.0125515859	0.0197116202	0.0193092005	0.0303139490

Table 2.6: First order EG4 scheme with structured mesh.

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
1352	0.1373479698	0.2557249796	0.2557249796	0.3868526789
5408	0.0842542833	0.1426600046	0.1426600046	0.2186379155
21632	0.0466051448	0.0752423735	0.0752423735	0.1161674183
81608	0.0251891793	0.0397398351	0.0397398351	0.0615873668
323208	0.0129899128	0.0202397963	0.0202397963	0.0314330487

Table 2.7: Second order EG4 scheme with unstructured mesh.

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
1264	0.0197490039	0.0202446344	0.0206034017	0.0349909780
5056	0.0081457957	0.0071086259	0.0071999686	0.0129894611
20224	0.0042207817	0.0031803658	0.0032108200	0.0061837764
80896	0.0027861919	0.0020034522	0.0020278136	0.0039860651
323584	0.0014438052	0.0010112703	0.0010188165	0.0020359834

Table 2.8: Second order EG4 scheme with structured mesh.

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
1352	0.0271745183	0.0226698987	0.0226698987	0.0420274085
5408	0.0123436358	0.0095945948	0.0095945948	0.0183433324
21632	0.0063715796	0.0045995503	0.0045995503	0.0091054243
81608	0.0034771196	0.0024646569	0.0025095746	0.0049459942
323208	0.0017069590	0.0011914380	0.0011899982	0.0023977759

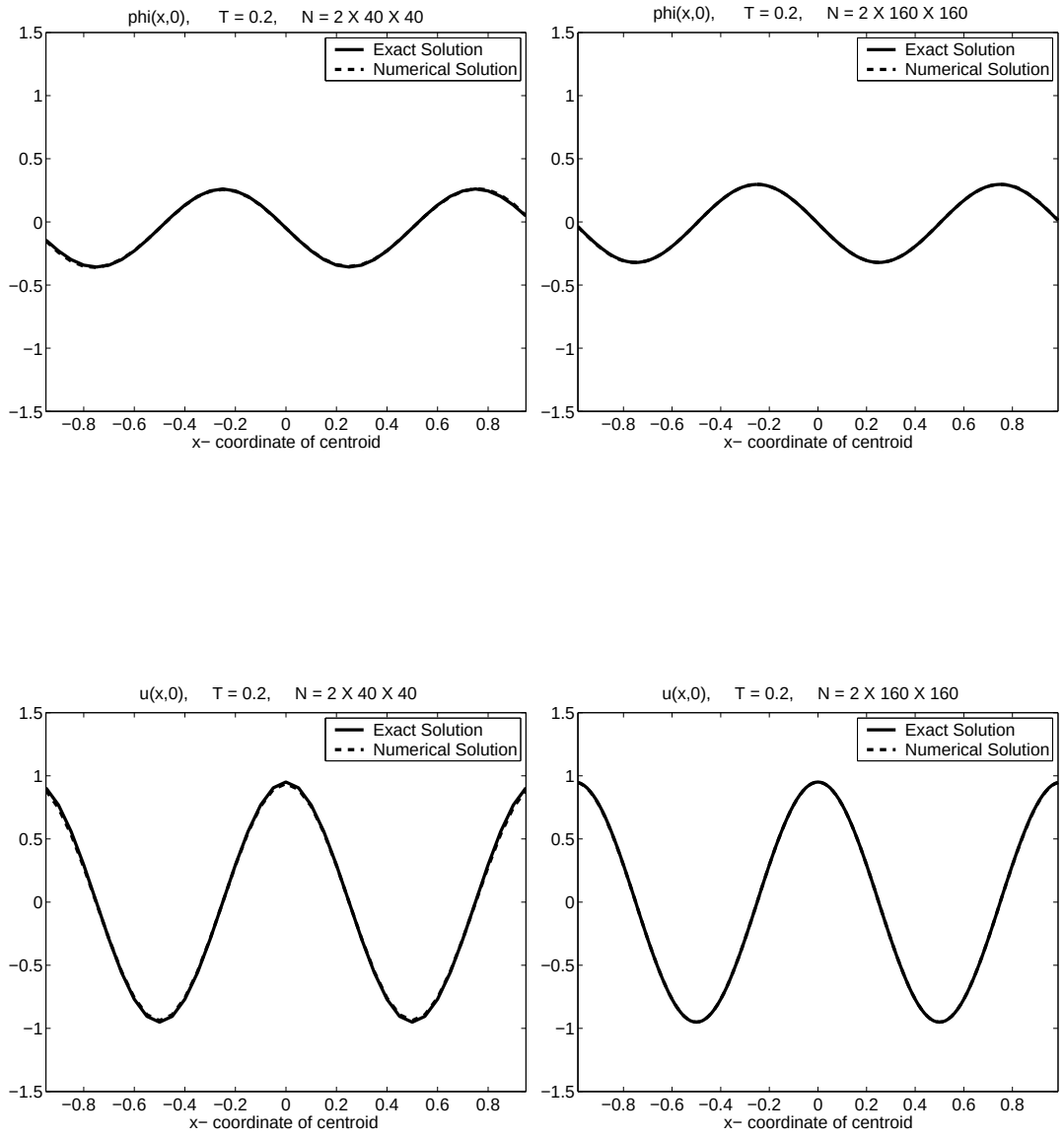

Figure 2.5: Second order EG3 scheme with exact boundary conditions.

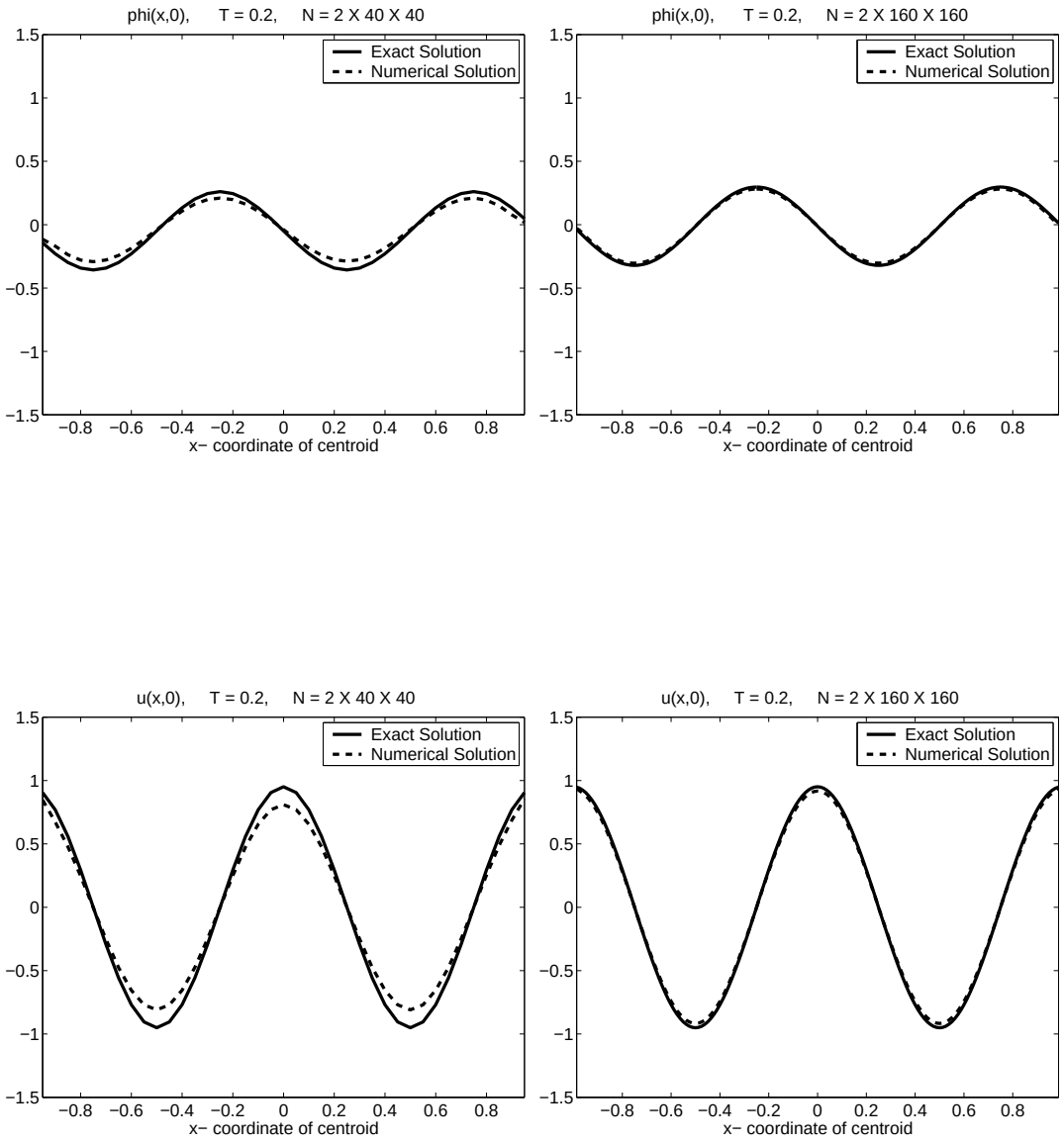

Figure 2.6: First order EG4 scheme with exact boundary conditions.

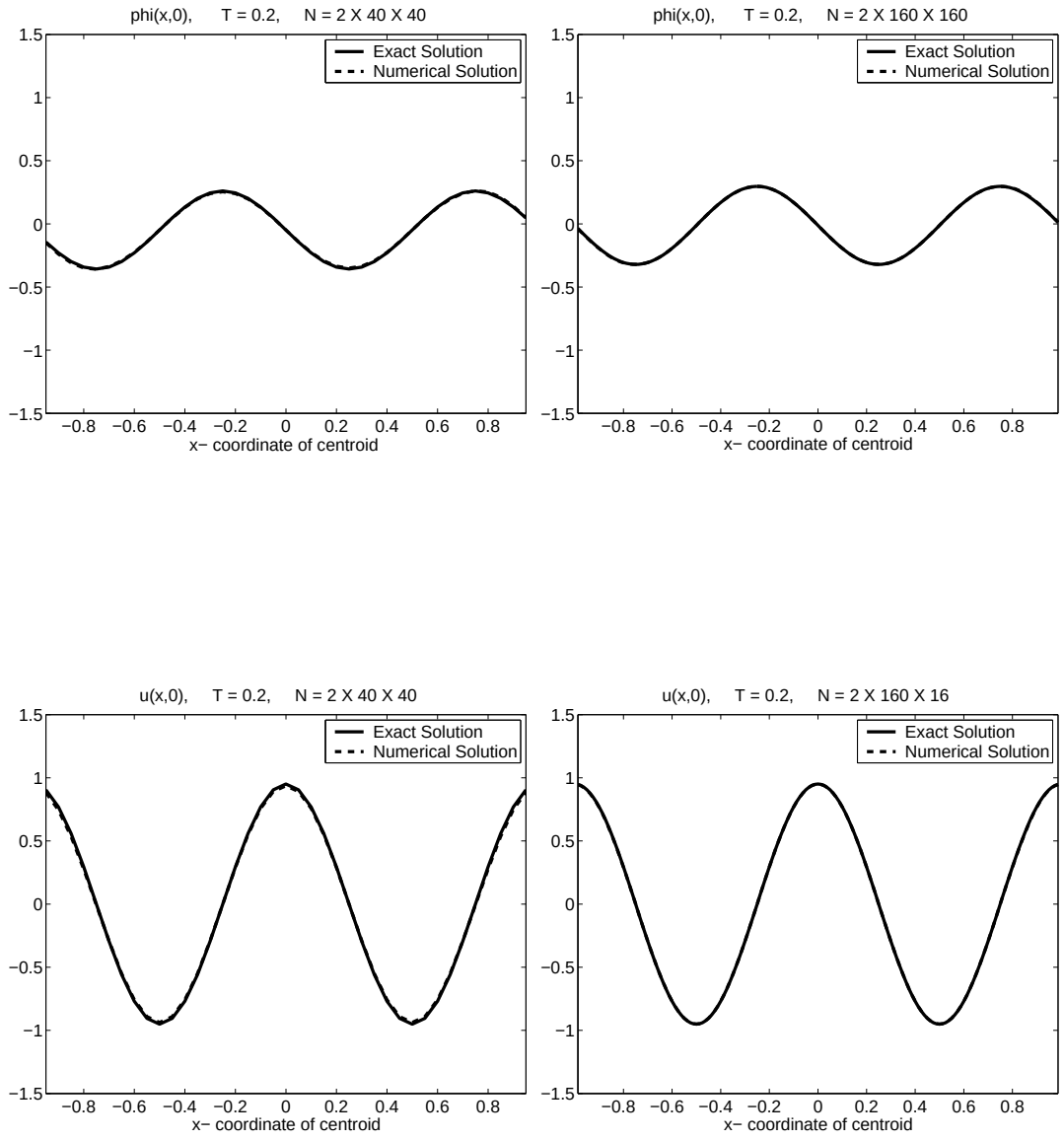

Figure 2.7: Second order EG4 scheme with exact boundary conditions.

First Order Scheme Second Order Scheme

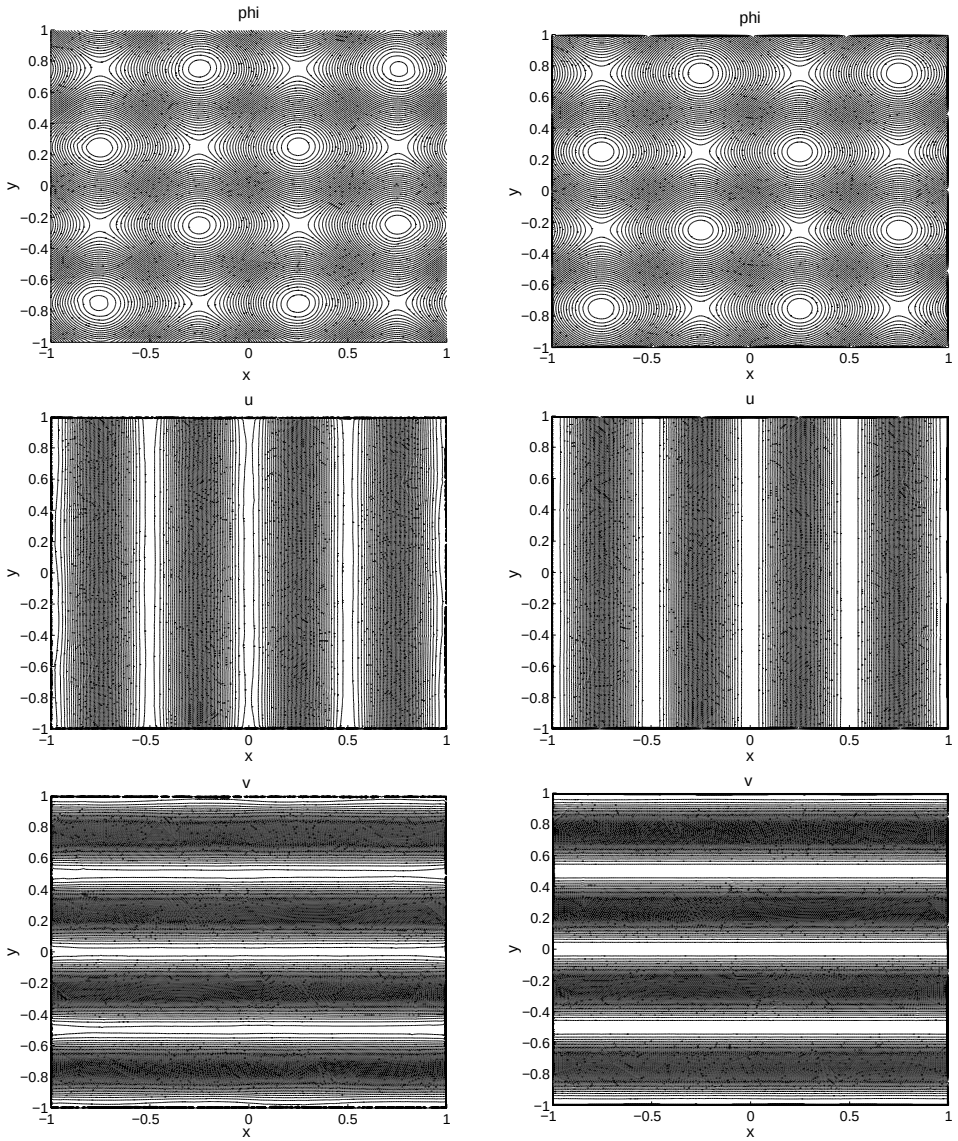

Figure 2.8: EG4 scheme with unstructured mesh.

Table 2.9: Comparison of L^2-error for rectangular and triangular meshes.

N_R	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	N	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
10×10	0.78670057163	$2\times 10\times 10$	0.69557797306
20×20	0.52951479563	$2\times 20\times 20$	0.46614313451
40×40	0.29694241375	$2\times 40\times 40$	0.27409758620
80×80	0.15897466957	$2\times 80\times 80$	0.14834821358
160×160	0.08259065119	$2\times 160\times 160$	0.07713042402
320×320	0.04216591393	$2\times 320\times 320$	0.03932893099

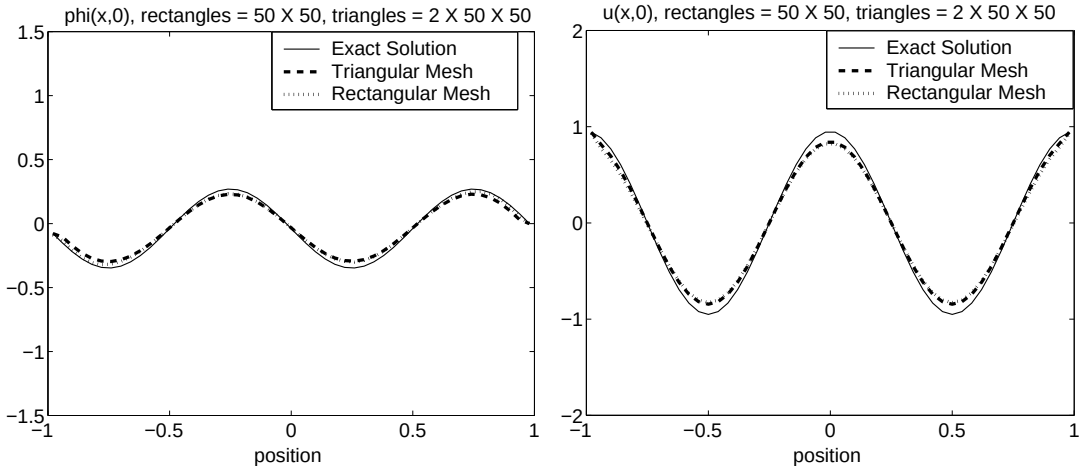

Figure 2.9: Comparison of EG4 scheme using rectangular and triangular meshes.

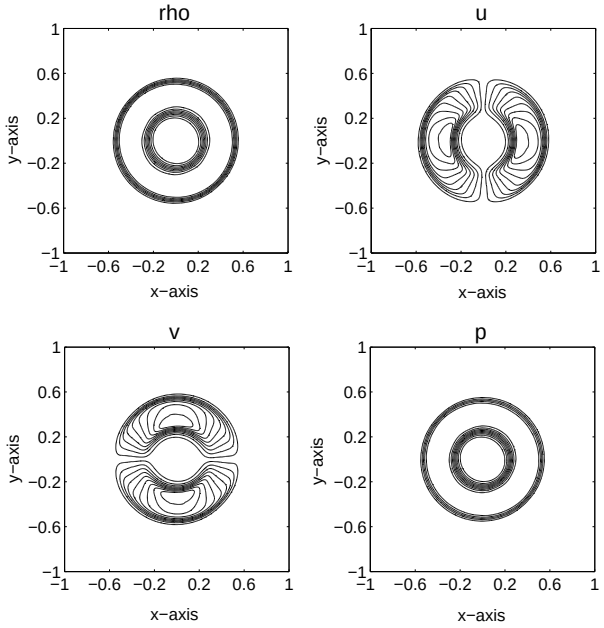

Figure 2.10: First order EG4 scheme for nonlinear wave equation system.

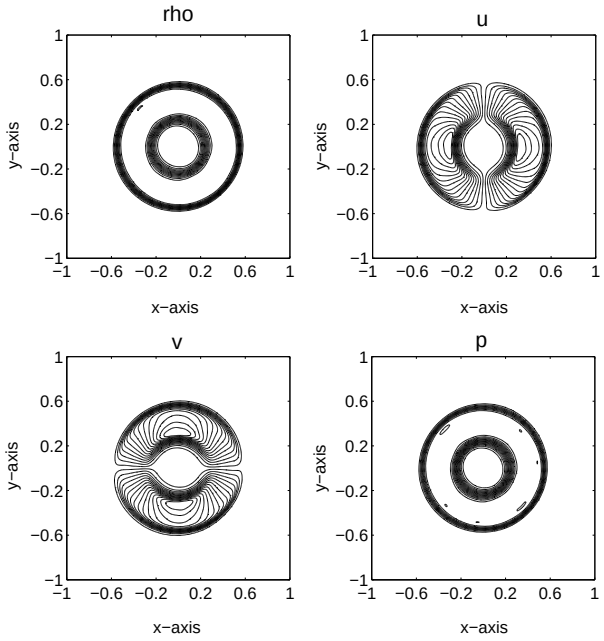

Figure 2.11: Second order EG4 scheme for nonlinear wave equation system.

Chapter 3

EG Schemes for the Linearized Euler Equations

The Euler equations are very important in the field of numerical hydrodynamics and describe the conservation of mass, momentum and energy of a compressible, inviscid fluid. These are reduced versions of Navier Stokes equations when viscosity and thermal conductivity terms are neglected. These terms are not considered to be significant in some physical phenomena e.g. in astrophysics. Mathematically the Euler equations are a set of nonlinear, coupled, hyperbolic partial differential equations. Theory of linear and nonlinear waves can be found in [59] and [68]. Hyperbolic differential equations have two important properties. The first property is that they allow discontinuous solutions. Physically this implies that the flow can contain shocks or contact discontinuities. The other property is that one can define the so called characteristics or characteristic speeds. These are the eigenvalues of the problem. The solution can be written in terms of a sum of eigenvectors, three in the case of a one-dimensional problem. The three eigenvectors are also called waves and are physically associated with the characteristic speeds v, $v - c$, $v + c$ i.e. the velocity of the flow and the velocity of sound added to and subtracted from the velocity of flow. The physical relevance of this is that in a gas no signal can travel faster than the local sound speed; v, $v - c$ and $v + c$ are the possible signal speeds within a flow with velocity v. This also means that the characteristics describe a domain of influence in space-time. Therefore a close relation exists between the characteristics and the shocks. If for example an explosion occurs at some point, its effect will spread with the shock speed.

In this chapter we discuss the system of hyperbolic conservation laws in several space variables. We start with the linear case and describe the linearization of the nonlinear system. First and second order EG schemes are presented for the linearized Euler equations and the advection wave equation system. Different physical situations such as subsonic and supersonic cases are discussed and numerical examples are presented. We also compare the EG schemes for linearized Euler equations with other numerical schemes such as the Lax-Friedrichs scheme [18] and the Lax-Wendroff [64] schemes.

3.1 General Hyperbolic System

The general form of a hyperbolic system in d space dimensions is given by

$$\frac{\partial \mathbf{u}}{\partial t} + \sum_{j=1}^{d} \frac{\partial}{\partial x_j} \mathbf{f}_j(\mathbf{u}) = 0, \quad \mathbf{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d, \quad t > 0, \tag{3.1}$$

Here Ω, a set of states, is an open subset of $\mathbb{R}^p$, and $\mathbf{f}_j$, $1 \leq j \leq d$, are d smooth fluxes from Ω into $\mathbb{R}^p$. Further

$$\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix}, \quad \mathbf{f}_j = \begin{pmatrix} f_{1j} \\ \vdots \\ f_{pj} \end{pmatrix},$$

where $\mathbf{u}$ is the vector of unknown states which is a mapping from $\mathbb{R}^d \times [0, \infty[$ into Ω. Let A_j be the Jacobian matrices of $\mathbf{f}_j(\mathbf{u})$ for $j = 1, \ldots, d$ i.e. $\mathcal{A}_j(\mathbf{u}) = \left(\frac{\partial f_{ij}}{\partial u_k}(\mathbf{u})\right)_{1 \leq i,k \leq p}$. Then the system (3.1) is hyperbolic iff for any $\mathbf{u} \in \Omega$ and any $\omega = (\omega_1, \ldots, \omega_d) \in \mathbb{R}^d$, $\omega \neq 0$, the **pencil matrix**

$$\mathcal{A}(\mathbf{u}, \omega) = \sum_{j=1}^{d} \omega_j \mathcal{A}_j(\mathbf{u})$$

has p real eigenvalues $\lambda_1(\mathbf{u}, \omega) \leq \lambda_2(\mathbf{u}, \omega) \leq \cdots \leq \lambda_p(\mathbf{u}, \omega)$ and p linearly independent corresponding eigenvectors $\mathbf{r}_1(\mathbf{u}, \omega), \ldots, \mathbf{r}_p(\mathbf{u}, \omega)$.
Cauchy problem : We define it as: Find a function $\mathbf{u} : (\mathbf{x}, t) \in \mathbb{R}^d \times [0, \infty[\rightarrow \mathbf{u}(\mathbf{x}, t) \in \Omega$ that solves system (3.1) and that satisfies the initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d,$$

where $\mathbf{u}_0 : \mathbb{R}^d \rightarrow \Omega$ is a given function.
Riemann problem: For the two-space variables, we define it as: Find $\mathbf{u}$ that solves

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y} = 0$$

along with the initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0,$$

where $\mathbf{u}_0$ are two states. The solution consists of finite number of constant states, being discontinuous along rays through the origin, see [41].

3.1.1 The Euler Equations

If we apply the conservation of mass, momentum and total energy to a compressible inviscid fluid where the thermal conductivity is neglected, we get Euler equations of gas dynamics:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \sum_{j=1}^{3} \frac{\partial}{\partial x_j}(\rho u_j) &= 0, \\
\frac{\partial}{\partial t}(\rho u_i) + \sum_{j=1}^{3} \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij}) &= 0, \quad 1 \le i \le 3, \\
\frac{\partial}{\partial t}(\rho e) + \sum_{j=1}^{3} \frac{\partial}{\partial x_j}\left((\rho e + p) u_j\right) &= 0\,.
\end{aligned} \tag{3.2}$$

Here ρ is the density of the fluid, $\mathbf{u} = (u_1, u_2, u_3)$ is the velocity, p is the pressure, $e = \mathcal{E} + \frac{|\mathbf{u}|^2}{2}$ is the specific total energy and $\mathcal{E}$ is the specific internal energy. We need an equation of state to close the above system.

$$p = p(\rho, \mathcal{E})$$

which shows the dependence of pressure on density and specific internal energy. One can see that the pressure is independent of the velocity vector $\mathbf{u}$. For a polytropic ideal gas, the equation of state has the form

$$p = (\gamma - 1)\rho\mathcal{E}, \qquad \gamma > 1.$$

where $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heat at constant pressure to the specific heat at constant volume. We can write the system (3.2) in the general form as follows. We set $q_i = \rho u_i, \quad 1 \le i \le 3$, $E = \rho e$, the set of states $\Omega = \left\{(\rho, \mathbf{q} = (q_1, q_2, q_3), E),\ \rho > 0,\ \mathbf{q} \in \mathbb{R}^3,\ E - \frac{|\mathbf{q}|^2}{2\rho} > 0\right\}$. Moreover the equation of state has the form

$$p = p\left(\rho, \frac{E}{\rho} - \frac{|\mathbf{q}|^2}{2\rho^2}\right),$$

Then the general form of Euler equations in two-space dimensions reads

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y} = 0\,, \tag{3.3}$$

where

$$\mathbf{u} = \begin{pmatrix} \rho \\ q_1 \\ q_2 \\ E \end{pmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} q_1 \\ p + \frac{q_1^2}{\rho} \\ \frac{q_1 q_2}{\rho} \\ \frac{(E+p)q_1}{\rho} \end{pmatrix}, \quad \mathbf{g}(\mathbf{u}) = \begin{pmatrix} q_2 \\ \frac{q_1 q_2}{\rho} \\ p + \frac{q_2^2}{\rho} \\ \frac{(E+p)q_2}{\rho} \end{pmatrix}.$$

Note that for isentropic flow, the equation of state is independent of energy i.e. $p = p(\rho)$, then the first three equations which describe the conservation of mass, momentum and

energy are sufficient to solve the system (3.3). Similarly if there is some kind of symmetry in the problem then the number of space variables are reduced accordingly. For example, if we consider slab symmetry, the Euler equations reduce to the following form.

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) &= 0, \\ \frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) &= 0, \\ \frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x}((\rho e + p)u) &= 0. \end{aligned}$$

3.1.2 Linear Systems

An important step to understand the nonlinear hyperbolic systems is the treatment of linear systems with constant coefficients. In this case the Riemann problem is easily solved. We consider one-dimensional linear system given by

$$\mathbf{u}_t + \mathcal{A}\,\mathbf{u}_x = 0, \tag{3.4}$$

together with the initial data

$$\mathbf{u}(x,0) = \mathbf{u}_0 = \begin{cases} \mathbf{u}_l\,, & x < 0\,, \\ \mathbf{u}_r, & x < 0\,. \end{cases}$$

where $\mathbf{u} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^p$ and $\mathcal{A}$ is a constant matrix. Similar to the nonlinear case, system (3.4) is called hyperbolic if the matrix $\mathcal{A}$ has p real eigenvalues $\lambda_1, \cdots, \lambda_p$ and p linearly independent right eigenvectors $\mathbf{r}_1, \cdots, \mathbf{r}_p$. If these eigenvalues are distinct then this system is called strictly hyperbolic system. The solution of this Riemann problem reads

$$\mathbf{u}(x,t) = \mathbf{v}\left(\frac{x}{t}\right) = \begin{cases} \mathbf{v}_0 = \mathbf{u}_l & , \quad \frac{x}{t} < \lambda_1 \\ \mathbf{v}_1 & , \quad \lambda_1 < \frac{x}{t} < \lambda_2 \\ \vdots & \\ \mathbf{v}_{p-1} & , \quad \lambda_{p-1} < \frac{x}{t} < \lambda_p \\ \mathbf{v}_p = \mathbf{u}_r & , \quad \frac{x}{t} > \lambda_p. \end{cases}$$

See [34] for further details. This shows that the initial discontinuity breaks up into p discontinuous waves, which propagate with the characteristic speed λ_k, $1 \le k \le p$, see Figure 3.1. Note that across the mth characteristic we have

$$\mathcal{A}\,[\mathbf{u}] = \lambda_m\,[\mathbf{u}]. \tag{3.5}$$

Condition (3.5) is called Rankine-Hugoniot condition. Therefore, across the line of discontinuity $x = \lambda_m t$, the Rankine-Hugoniot condition is satisfied.

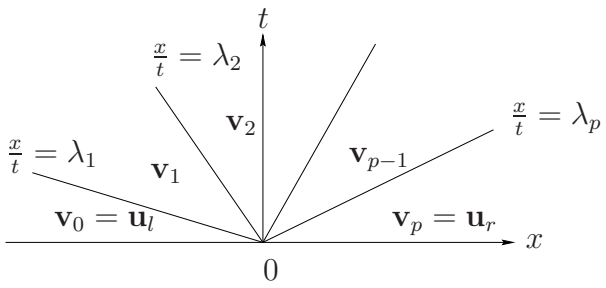

Figure 3.1: Solution of Riemann problem for a linear system.

3.1.3 Linearization of Nonlinear Systems

Consider the one-dimensional nonlinear hyperbolic system

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0\,, \quad x \in \mathbb{R}\,. \tag{3.6}$$

In the quasi linear form system (3.6) can be written as

$$\mathbf{u}_t + \mathcal{A}\,\mathbf{u}_x = 0,$$

where $\mathcal{A}(\mathbf{u}) = \frac{df_i}{du_k}(\mathbf{u})$ is the Jacobian matrix. We can define characteristics by integrating the eigenvalues of $\mathcal{A}(\mathbf{u})$. There are p characteristic curves through each point. The curve $x(t)$ in the m-th family satisfies

$$\begin{aligned} x'(t) &= \lambda_m(\mathbf{u}(x(t),t)), \\ x(0) &= x_0, \end{aligned}$$

for some x_0. Note that the characteristic speed λ_m now depends on the solution $\mathbf{u}$. Thus we cannot proceed by determining the characteristics and then solving a system of ODEs along the characteristics. It is known that the characteristics yield valuable information about what happens locally for smooth data. In particular, if we linearize the problem about a constant state $\mathbf{u}'$, see LeVeque [34] for further details, we obtain a constant coefficient linear system, with the Jacobian frozen at $\mathcal{A}(\mathbf{u}')$. This is relevant if we consider propagation of small disturbances. Thus assume an expansion of the solution of the form

$$\mathbf{u}(x,t) = \mathbf{u}' + \epsilon\mathbf{u}^{(1)}(x,t) + \epsilon^2\mathbf{u}^{(2)}(x,t) + \cdots\,, \tag{3.7}$$

where $\mathbf{u}'$ is a constant and ϵ is small. In order to freeze the Jacobian matrices at constant states $\mathbf{u}'$, we expand it in a Taylor series as follows

$$\mathcal{A}(\mathbf{u}) = \mathcal{A}(\mathbf{u}') + (\mathbf{u} - \mathbf{u}')\frac{\partial \mathcal{A}}{\partial \mathbf{u}}(\mathbf{u}') + \cdots\,,$$

Substituting $(\mathbf{u} - \mathbf{u}') = \epsilon\mathbf{u}^{(1)}$ from (3.7), we obtain

$$\mathcal{A}(\mathbf{u}) = \mathcal{A}(\mathbf{u}') + \epsilon\mathbf{u}^{(1)}\frac{\partial \mathcal{A}}{\partial \mathbf{u}}(\mathbf{u}')\,. \tag{3.8}$$

Using (3.7) and (3.8) in the one-dimensional nonlinear hyperbolic system, we obtain the following linearized Euler equation system.

$$\mathbf{u}_t^{(1)}(x,t) + \mathcal{A}(\mathbf{u}')\mathbf{u}_x^{(1)} = 0. \tag{3.9}$$

Thus small disturbances propagate approximately along characteristic curves of the form $x_m(t) = x_0 + \lambda_m(\mathbf{u}')t$. In this thesis we consider the linearized Euler equations in two-space dimensions . In this case the Jacobian matrices which are frozen at the constant state $(\rho, u, v, p)^T = (\rho', u', v', p')^T$ are given as

$$\mathcal{A}_1 := \begin{pmatrix} u' & \rho' & 0 & 0 \\ 0 & u' & 0 & \frac{1}{\rho'} \\ 0 & 0 & u' & 0 \\ 0 & \rho' c'^2 & 0 & u' \end{pmatrix}, \quad \mathcal{A}_2 := \begin{pmatrix} v' & 0 & \rho' & 0 \\ 0 & v' & 0 & 0 \\ 0 & 0 & v' & \frac{1}{\rho'} \\ 0 & 0 & \rho' c'^2 & v' \end{pmatrix},$$

where c' is the local speed of sound, i.e. $c' = \sqrt{\frac{\gamma p'}{\rho'}}$.

3.2 Approximate Evolution Operators For Linearized Euler Equations in 2D

In this section we present the approximate evolution operators for the linearized Euler equations, namely the EG3-Euler and the EG4-Euler approximate evolution operators. These operators have been derived in [43] and [71]. The 2D linearized Euler equations read

$$\mathbf{U}_t + \mathcal{A}_1(\mathbf{U}')\mathbf{U}_x + \mathcal{A}_2(\mathbf{U}')\mathbf{U}_y = 0, \quad \mathbf{x} = (x,y)^T \in \mathbb{R}^2, \tag{3.10}$$

where

$$\mathbf{U} := \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}, \quad \mathbf{U}' := \begin{pmatrix} \rho' \\ u' \\ v' \\ p' \end{pmatrix}, \quad \mathcal{A}_1 := \begin{pmatrix} u' & \rho' & 0 & 0 \\ 0 & u' & 0 & \frac{1}{\rho'} \\ 0 & 0 & u' & 0 \\ 0 & \rho'(c')^2 & 0 & u' \end{pmatrix}$$

$$\text{and} \quad \mathcal{A}_2 := \begin{pmatrix} v' & 0 & \rho' & 0 \\ 0 & v' & 0 & 0 \\ 0 & 0 & v' & \frac{1}{\rho'} \\ 0 & 0 & \rho'(c')^2 & v' \end{pmatrix}.$$

Here ρ indicates the density, u and v are the components of the velocity in x and y directions and p denotes the pressure. Symbols ρ', u', v' and $p' = \frac{c'^2\rho'}{\gamma}$ are the local variables at a point (x', y'), c' is the local speed of the sound there and γ is isotropic exponent ($\gamma = 1.4$ for dry air).

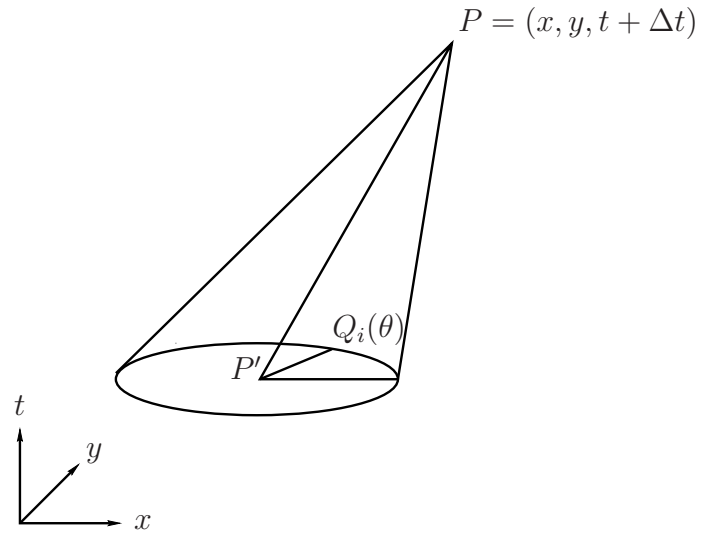

Figure 3.2: Bicharacteristics along the Mach cone through P and $Q_i(\theta)$, supersonic case.

Now the approximate evolution operator for EG3-Euler scheme reads

$$\begin{aligned}
\rho(P) &= \rho(P') - \frac{p(P')}{c'^2} + \frac{1}{2\pi}\int_0^{2\pi}\left[\frac{p(Q)}{c'^2} - 2\frac{\rho'}{c'}u(Q)\cos\theta - 2\frac{\rho'}{c'}v(Q)\sin\theta\right]d\theta \\
&\quad +\mathcal{O}(\Delta t^2)\,, \qquad (3.11)\\
u(P) &= \frac{1}{2}u(P') + \frac{1}{2\pi}\int_0^{2\pi}\left[-2\frac{p(Q)}{\rho' c'}\cos\theta + u(Q)(3\cos^2\theta - 1) + 3v(Q)\sin\theta\cos\theta\right]d\theta \\
&\quad +\mathcal{O}(\Delta t^2)\,, \qquad (3.12)\\
v(P) &= \frac{1}{2}v(P') + \frac{1}{2\pi}\int_0^{2\pi}\left[-2\frac{p(Q)}{\rho' c'}\sin\theta + 3u(Q)\sin\theta\cos\theta + v(Q)(3\sin^2\theta - 1)\right]d\theta \\
&\quad +\mathcal{O}(\Delta t^2)\,, \qquad (3.13)\\
p(P) &= \frac{1}{2\pi}\int_0^{2\pi}\left[p(Q) - 2\rho' c' u(Q)\cos\theta - 2\rho' c' v(Q)\sin\theta\right]d\theta + \mathcal{O}(\Delta t^2)\,. \qquad (3.14)
\end{aligned}$$

while the approximate evolution operator for EG4-Euler scheme is

$$\begin{aligned}
\rho(P) &= \rho(P') - \frac{p(P')}{c'^2} + \frac{1}{2\pi}\int_0^{2\pi}\left[\frac{p(Q)}{c'^2} - 2\frac{\rho'}{c'}u(Q)\cos\theta - 2\frac{\rho'}{c'}v(Q)\sin\theta\right]d\theta \\
&\quad +\mathcal{O}(\Delta t^2)\,, \qquad (3.15)\\
u(P) &= \frac{1}{2\pi}\int_0^{2\pi}\left[-2\frac{p(Q)}{\rho' c'}\cos\theta + 2u(Q)\cos^2\theta + 2v(Q)\sin\theta\cos\theta\right]d\theta + \mathcal{O}(\Delta t^2), \qquad (3.16)\\
v(P) &= \frac{1}{2\pi}\int_0^{2\pi}\left[-2\frac{p(Q)}{\rho' c'}\sin\theta + 2u(Q)\sin\theta\cos\theta + 2v(Q)\sin^2\theta\right]d\theta + \mathcal{O}(\Delta t^2), \qquad (3.17)\\
p(P) &= \frac{1}{2\pi}\int_0^{2\pi}\left[p(Q) - 2\rho' c' u(Q)\cos\theta - 2\rho' c' v(Q)\sin\theta\right]d\theta + \mathcal{O}(\Delta t^2). \qquad (3.18)
\end{aligned}$$

where $P = (x, y, t+\Delta t)$ and $P' = (x-(u'-c'\cos\theta)\Delta t, y-(v'-c'\cos\theta)\Delta t)$, see Figure 3.2. P' is the center of the characteristic cone base which does not correspond to a quadrature point but is displaced from it due to the advection effect. For more details see [43], [45] and [71].

3.3 Numerical Algorithms

In case of Euler equations, we note that due to the effect of advection the center of the sonic circle which constitutes the base of the Mach cone does not coincide with a vertex or a midpoint in the mesh. This means that any vertex or midpoint will be shifted according to local speeds. for linearized Euler equations the evolution as well as the finite volume update steps are carried out using primitive variables. Thus the first order algorithm for linearized Euler equations reads

- Input the initial data: $\rho_i^0,\ u_i^0,\ v_i^0,\ p_i^0$.
- Determine the center of the sonic circles using the local velocities.
- Determine the time step Δt.
- Do the time loop
  1. Determine the intermediate values: $\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}},\ \tilde{u}_{s_{i_j}}^{n+\frac{1}{2}},\ \tilde{v}_{s_{i_j}}^{n+\frac{1}{2}},\ \tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}$.
  2. Update the primitive variables:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3}\left(n_{x_{s_{i_j}}}\left(u'\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}} + \rho'\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}\right) + n_{y_{s_{i_j}}}\left(v'\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}} + \rho'\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}\right)\right), \tag{3.19}$$

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\left(u'\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}} + \frac{1}{\rho'}\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\right) + n_{y_{s_{i_j}}} v'\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}, \tag{3.20}$$

$$v_i^{n+1} = v_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}} u'\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}} + n_{y_{s_{i_j}}}\left(v'\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}} + \frac{1}{\rho'}\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\right), \tag{3.21}$$

$$\begin{aligned} p_i^{n+1} &= p_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\left(\rho'(c')^2\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}} + u'\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\right) \\ &+ n_{y_{s_{i_j}}}\left(\rho'(c')^2\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}} + v'\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\right). \end{aligned} \tag{3.22}$$

  3. Apply the boundary conditions.
- End the time loop.

To have a second order algorithm we have to carry out a recovery stage before applying the approximate evolution operator. We use the same recovery procedure as given in Section 2.5. The second order algorithm reads

- Input the initial data: ρ_i^0, u_i^0, v_i^0, p_i^0.
- **Carry out the recovery stage**.
- Determine the center of the sonic circles using the local velocities.
- Determine the time step Δt.
- Do the time loop
    1. Determine the intermediate values: $\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}$.
    2. Update using equations (3.19)-(3.22).
    3. Apply the boundary conditions.
    4. **Carry out the recovery stage**.
- End the time loop.

It is important to point out that in the case of linearized Euler equations, local variables are kept constant. This means that all local variables have same value at all vertices and midpoints and this value once assigned outside the time loop does not change with time. With these known values of the local variables, it is easy to find the new centers of the bases of the Mach cones. Since local velocities are constant, the circle position does not change with time. Determination of the time step also depends upon the local velocities i.e.

$$\Delta t = \frac{h_{\min}\,\mathrm{CFL}}{\max(|u'|+c', |v'|+c')}\,.$$

The determination of $h_{\min}$ is carried out in the same manner as mentioned in Chapter 2 in case of wave equation system. This choice of Δt prevents the sonic circle from crossing the close neighbors. Which means that the circle is confined only to the close neighbors. The approximate evolution operators are applied at the new centers for the sonic circles which require the angular contribution of the neighbors at that point. The angles are therefore computed with respect to the new origin at all quadrature points. In case of EG3-Euler scheme one has to evaluate the values $\rho(P')$, $u(P')$, $v(P')$ and $P(P')$ at all vertices and midpoints of the edges. This is done by taking weighted average of the piecewise constant values from the neighboring elements at the point. The length of the arc cut from the sonic circle at that point by a neighboring element is multiplied as a weight to the piecewise constant value from that element.

The advection wave equation system: Taking $\rho' = c' = 1$ and neglecting the first

equation in sytem (3.10) gives the advection wave equation system with propagation speed equal to 1.

$$\begin{aligned} p_t + u'\, p_x + v'\, p_y + u_x + v_y &= 0, \\ u_t + u'\, u_x + v'\, u_y + p_x &= 0, \\ v_t + u'\, v_x + v'\, v_y + p_y &= 0. \end{aligned} \tag{3.23}$$

3.4 Numerical Experiments

Example 3.1

In this numerical experiment we consider the advection wave equation system (3.23). We take the following smooth initial data, see [71]

$$\begin{aligned} \phi(x,y,0) &= -(\sin(2\pi x) + \sin(2\pi y))\,, \\ u(x,y,0) &= 0\,, \\ v(x,y,0) &= 0\,. \end{aligned}$$

The exact solution is

$$\begin{aligned} \phi(x,y,t) &= -\cos(2\pi t)(\sin 2\pi(x-u't) + \sin 2\pi(y-v't))\,, \\ u(x,y,t) &= \sin(2\pi t)\cos 2\pi(x-u't)\,, \\ v(x,y,t) &= \sin(2\pi t)\cos 2\pi(y-v't)\,. \end{aligned}$$

Two physical situations are considered here , the subsonic and the supersonic cases. For the subsonic case we take $u' = v' = 0.5$ and for the supersonic case we set $u' = v' = 0.8$ together with CFL $= 0.4$ and absolute time $T = 0.2$. The computational domain is $\Omega = [-1,1] \times [-1,1]$. If N_x and N_y are the number of rectangular cells along x and y-axis respectively in a regular triangular mesh then $N = 2N_xN_y$ where N is the total number of triangular cells in the mesh. In Tables 3.1 and 3.2 we present the results for the first order EG4-Euler scheme for subsonic and supersonic cases respectively. Here we use exact boundary conditions and the structured mesh 2 as shown in Figure A.1. The comparison of the exact solution with the numerical solution for the first order scheme is shown in Figure 3.3 and Figure 3.4 for the subsonic and supersonic cases respectively. This is a 1D plot along the line $y = 0$. The plot shows that the numerical solution is in good agreement with the exact solution. The error analysis of second order EG4-Euler scheme is given in Tables 3.3 and 3.4 for the subsonic and supersonic cases respectively. The error in the second order EG scheme is several times smaller than for the first order. Figures 3.5 and 3.6 give a graphical representation of the second order numerical scheme and its comparison with the exact solution. With the increase of mesh size the numerical solution becomes closer to the exact solution and gives a better approximation. In Tables 3.5 and 3.6, we compare the accuracy of EG4-Euler scheme for unstructured and structured meshes for approximately the same number of points which indicates that the EG schemes are suitable for the unstructured grids also. Similarly second order results for unstructured and structured meshes are compared in Tables 3.7 and 3.8.

Example 3.2

We simulate numerically the propagation of acoustic, vorticity and entropy pulses in a uniform mean flow. This experiment has been studied by Zahaykah [71] for rectangular mesh. We assume that the flow of the fluid is along the diagonal of the mesh that is in the direction making 45^o angle with the x-axis. We consider the linearized Euler equation system (3.10) together with the following initial data that describes the location of acoustic, entropy and vorticity pulses:

$$\begin{aligned}
\rho(x,y,0) &= 2.5\exp(-40((x-x_a)^2+(y-y_a)^2)) + 0.5\exp(-40((x-x_b)^2+(y-y_b)^2)),\\
u(x,y,0) &= 0.05\exp(-40((x-x_b)^2+(y-y_b)^2)),\\
v(x,y,0) &= -0.05\exp(-40((x-x_b)^2+(y-y_b)^2)),\\
p(x,y,0) &= 2.5\exp(-40((x-x_a)^2+(y-y_a)^2)).
\end{aligned}$$

Assume $u' = v' = 0.5\sin(\frac{\pi}{4})$ which means that the local flow is subsonic. At time $T = 0$ an acoustic pulse is generated at $(x_a, y_a) = (-0.31, -0.31)$, and combined entropy and vorticity pulses originate at $(x_b, y_b) = (0.39, 0.39)$. The mean flow interacts with these pulses and stretches the acoustic, entropy and density pulses. The intensity, shape and profile of the propagating waves are also affected by the mean flow. The vorticity pulse causes the energy transfer between the propagating disturbances and the mean flow. We set CFL=0.45, take a mesh consisting of $2 \times 200 \times 200$ cells and employ extrapolated boundary conditions. We examine the propagation and physical locations of the pulses after time $T = 0.166$. Isolines of the computed density ρ, velocities and pressure using first order EG3-Euler and EG4-Euler schemes are shown in Figures 3.7 and 3.9. Plots for the second order EG3-Euler and EG4-Euler schemes are shown in Figures 3.8 and 3.10. The pulses move away from their original positions and get closer to each other as time passes. The second order plot gives a very refined picture of the physical mechanisms going on. In Figure 3.12, we plot the density ρ along the line $y = x$ and compare the first order EG4-Euler scheme with the first order Lax-Friedrichs scheme [18] and second order Lax-Wendroff scheme [64]. Both the Lax-Friedrichs and Lax-Wendroff schemes are plotted on a rectangular mesh with 200×200 rectangles while EG4-Euler scheme is plotted on a triangular mesh with $2 \times 200 \times 200$ triangles in the mesh. It is evident that EG4-Euler scheme is several times accurate than the Lax-Friedrichs scheme. Compared with the second order Lax-Wendroff scheme, first order EG4-Euler scheme has relatively more dissipation, particularly at maxima and minima.

Table 3.1: First order EG4-Euler scheme (subsonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.2287231442	0.3032928501	0.3032928501	0.4860940058
$2\times 40\times 40$	0.1417699108	0.1750108389	0.1750108389	0.2852302493
$2\times 80\times 80$	0.0797841969	0.0944292656	0.0944292656	0.1555612113
$2\times 160\times 160$	0.0424906677	0.0491286468	0.0491286468	0.0814414190
$2\times 320\times 320$	0.021946113	0.0250684675	0.0250684675	0.0416951797

Table 3.2: First order EG4-Euler scheme (supersonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.3377097939	0.2820635521	0.2820635521	0.5226543788
$2\times 40\times 40$	0.1648985862	0.1648985862	0.1648985862	0.3099082328
$2\times 80\times 80$	0.1131940327	0.0900183550	0.0900183550	0.1703511008
$2\times 160\times 160$	0.0597638213	0.0472191810	0.0472191810	0.0896159386
$2\times 320\times 320$	0.0307372664	0.0242101758	0.0242101758	0.0460113549

Table 3.3: Second order EG4-Euler scheme (subsonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.0609746674	0.0423121461	0.0423121461	0.0854315252
$2\times 40\times 40$	0.0300827178	0.0187751436	0.0187751436	0.0401245804
$2\times 80\times 80$	0.0151395292	0.0092255711	0.0092255711	0.0199856866
$2\times 160\times 160$	0.0077054695	0.0046439500	0.0046439500	0.0101245644
$2\times 320\times 320$	0.0038690553	0.0023307357	0.0023307357	0.0050827401

Table 3.4: Second order EG4-Euler scheme (supersonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.0839318451	0.0603927266	0.0603927260	0.1197460541
$2\times 40\times 40$	0.0423139761	0.0290946385	0.0290946385	0.0590209163
$2\times 80\times 80$	0.0214903760	0.0145847519	0.0145847519	0.0297870146
$2\times 160\times 160$	0.0107642136	0.0073262480	0.0073262480	0.0149404188
$2\times 320\times 320$	0.0053985865	0.0036867388	0.0036867388	0.0075052529

Table 3.5: First order EG4 scheme (unstructured mesh, subsonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
1264	0.2090766795	0.2534964164	0.2514140449	0.4137420852
5056	0.1245254626	0.1430963486	0.1411816251	0.2364643887
20224	0.0683283498	0.0761475007	0.0748860241	0.1267877039
80896	0.0350018581	0.0386493459	0.0379515678	0.0644920422
323584	0.0179821922	0.0195552012	0.0191868480	0.0327704176

Table 3.6: First order EG4 scheme (structured mesh, subsonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
1352	0.1932509434	0.2489122966	0.2489122966	0.4015726460
5408	0.1150930994	0.1393751021	0.1393751021	0.2282482415
21632	0.0631700200	0.0739731112	0.0739731112	0.1222067666
81608	0.0341088947	0.0392421733	0.0392421733	0.0651407171
323584	0.0175897118	0.0200398019	0.0200398019	0.0333554386

Table 3.7: Second order EG4 scheme (unstructured mesh, subsonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
1264	0.0401337546	0.0279533773	0.0280653784	0.0563894939
5056	0.0191832227	0.0120537083	0.0124758214	0.0258637592
20224	0.0096962424	0.0058479148	0.0061719865	0.0128960709
80896	0.0053019776	0.0030817698	0.0032668629	0.0069484290
323584	0.0026916558	0.0015917740	0.0016671514	0.003543747

Table 3.8: Second order EG4 scheme (structured mesh, subsonic case).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{u}(T)-\mathbf{U}^n\|_{L^2}$
1352	0.0446340257	0.0301024798	0.0301024798	0.0616807493
5408	0.0234135868	0.0143161142	0.0143161142	0.0309531630
21632	0.0117561746	0.0071168622	0.0071168622	0.0154760168
81608	0.0061080577	0.0036823330	0.0036823330	0.0080266756
326432	0.0030789474	0.0018503797	0.0018503797	0.0040407583

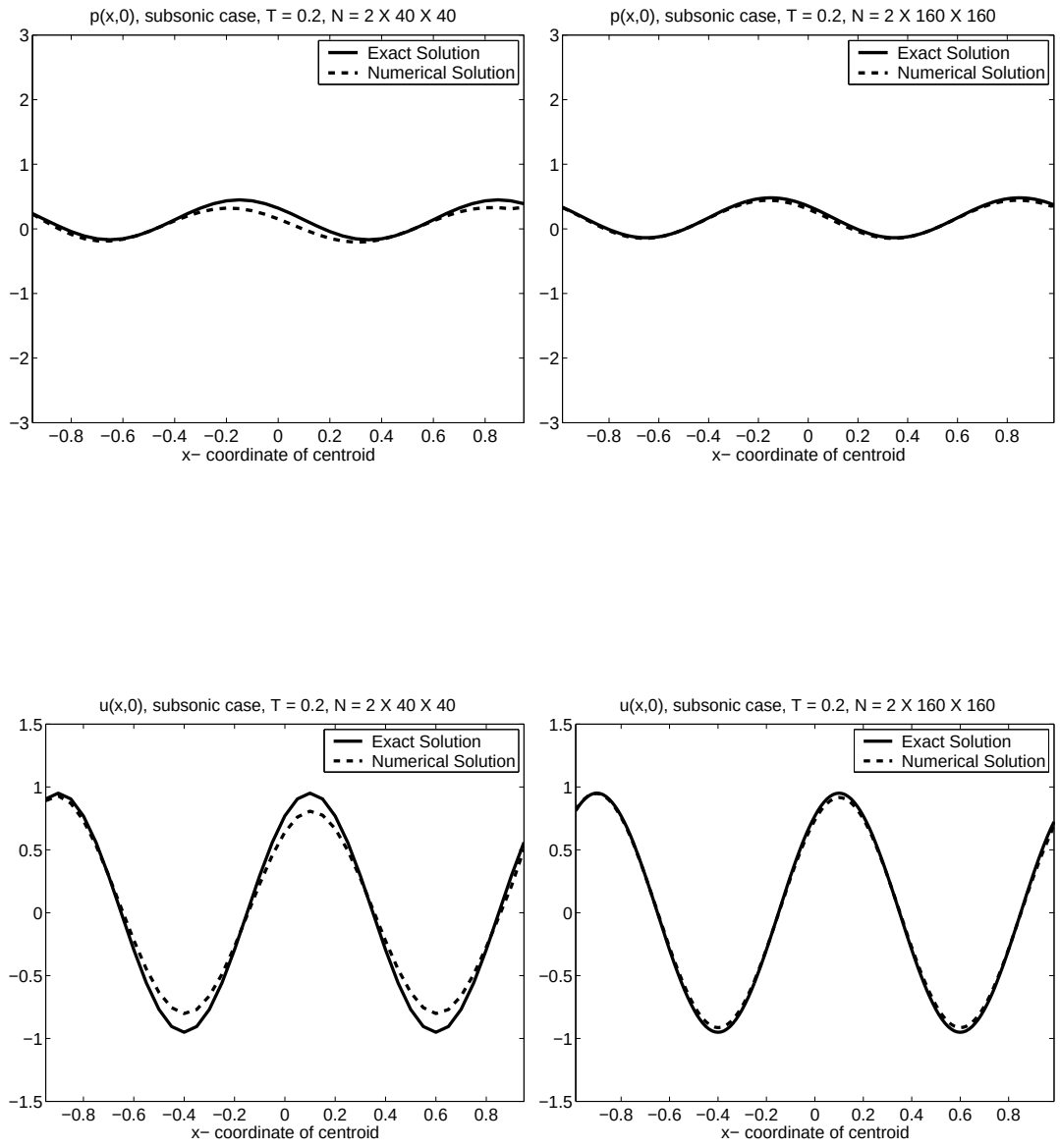

Figure 3.3: First order EG4-Euler scheme (subsonic case).

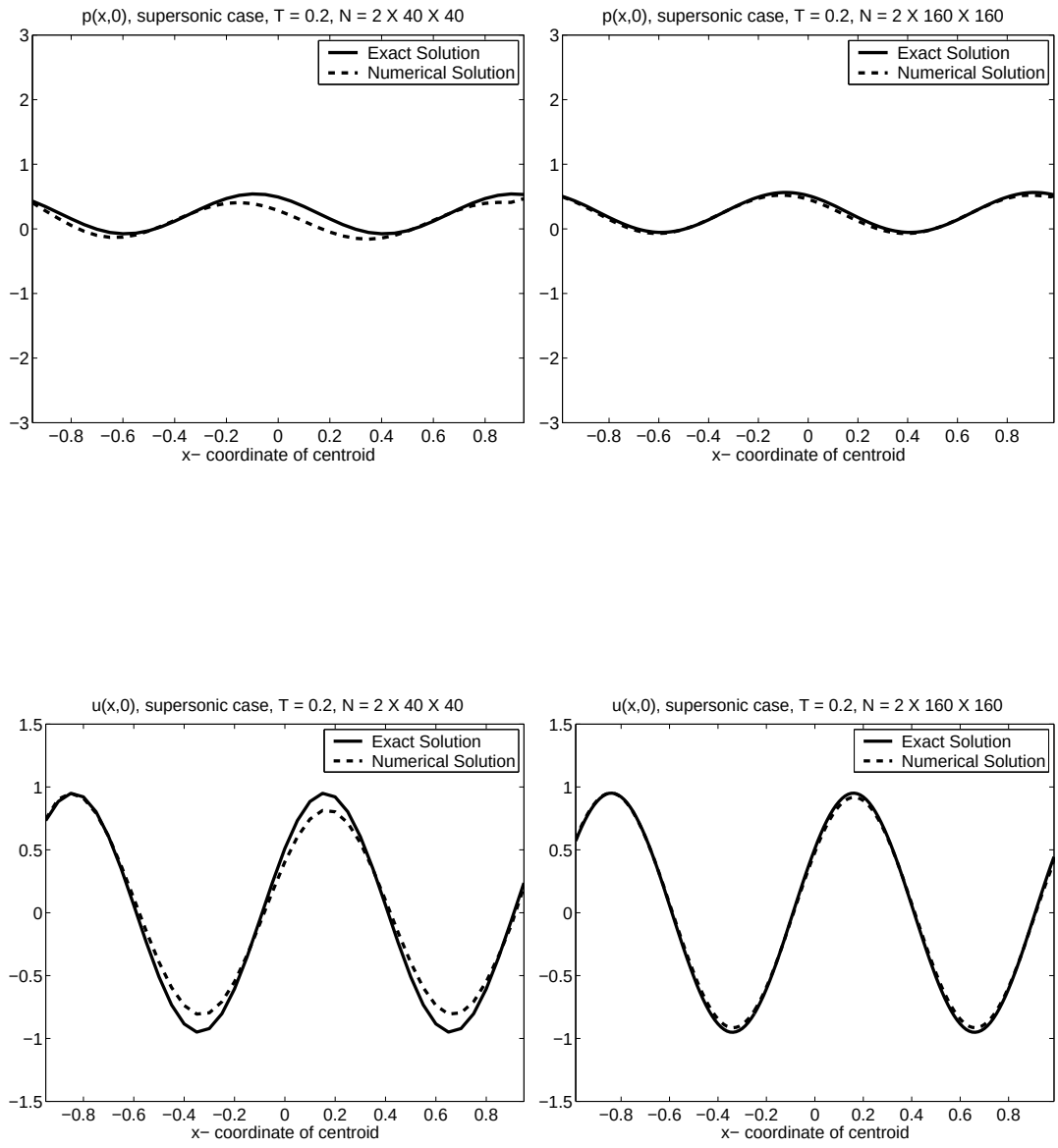

Figure 3.4: First order EG4-Euler scheme (supersonic case).

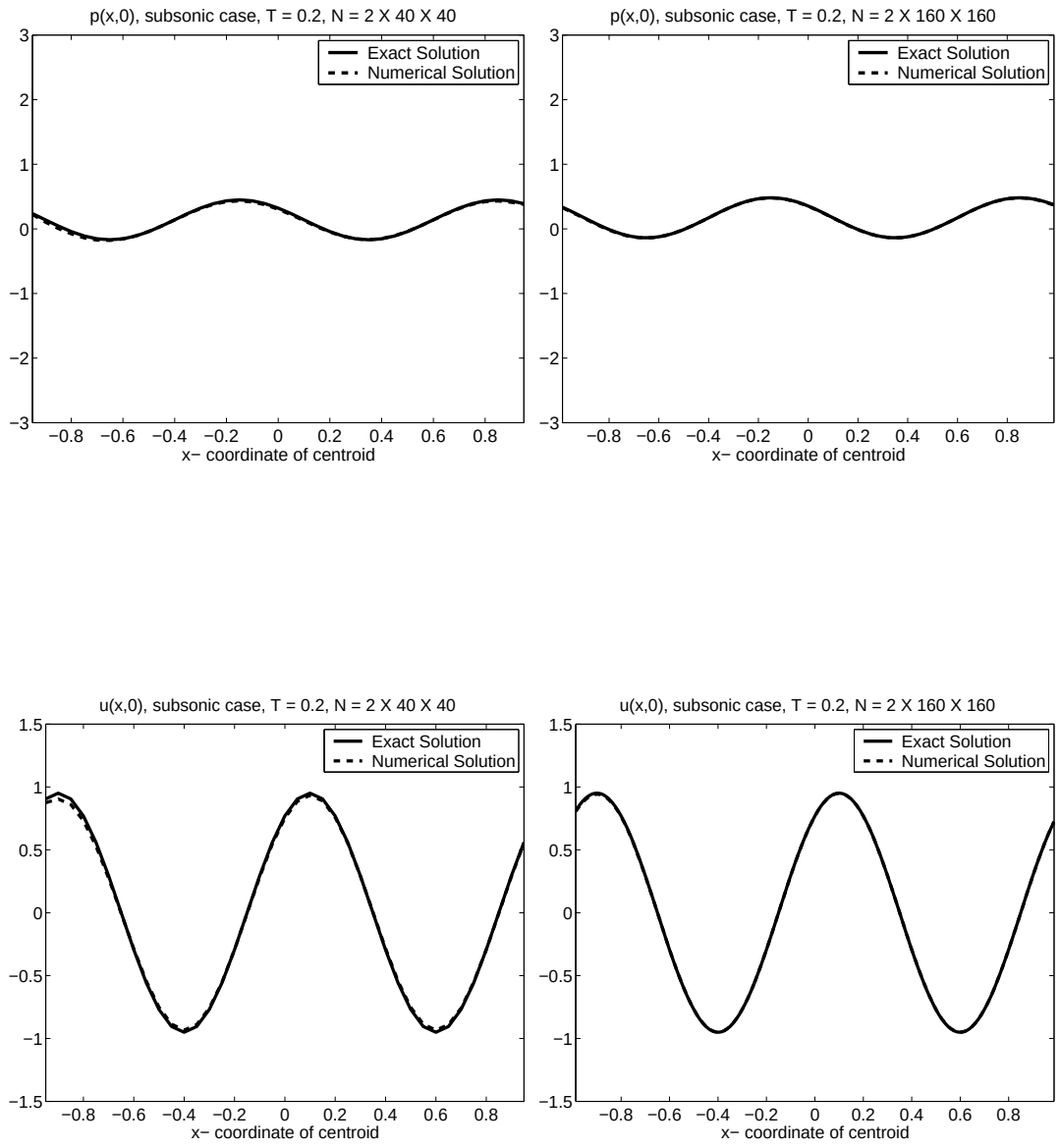

Figure 3.5: Second order EG4-Euler scheme (subsonic case).

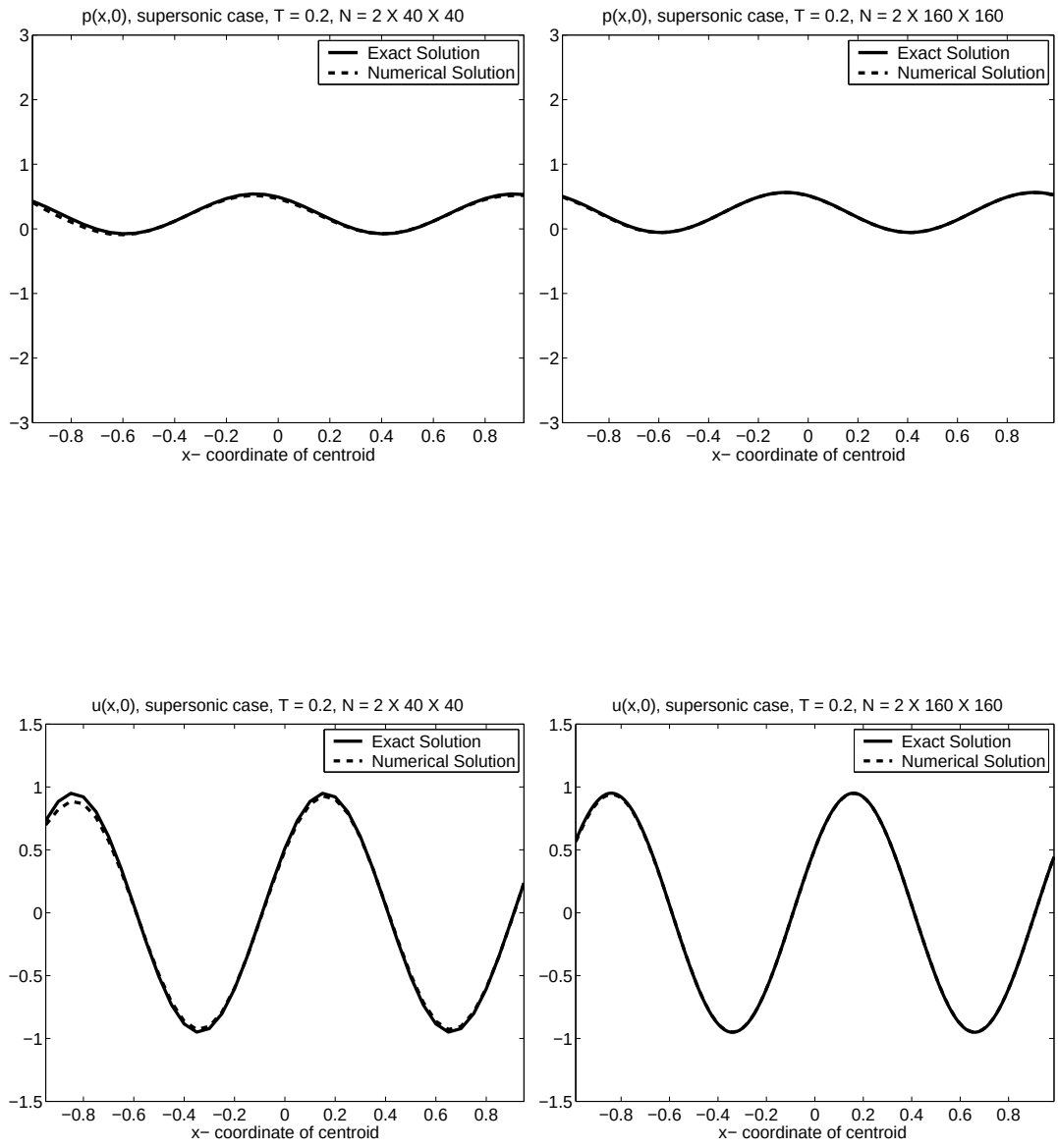

Figure 3.6: Second order EG4-Euler scheme (supersonic case).

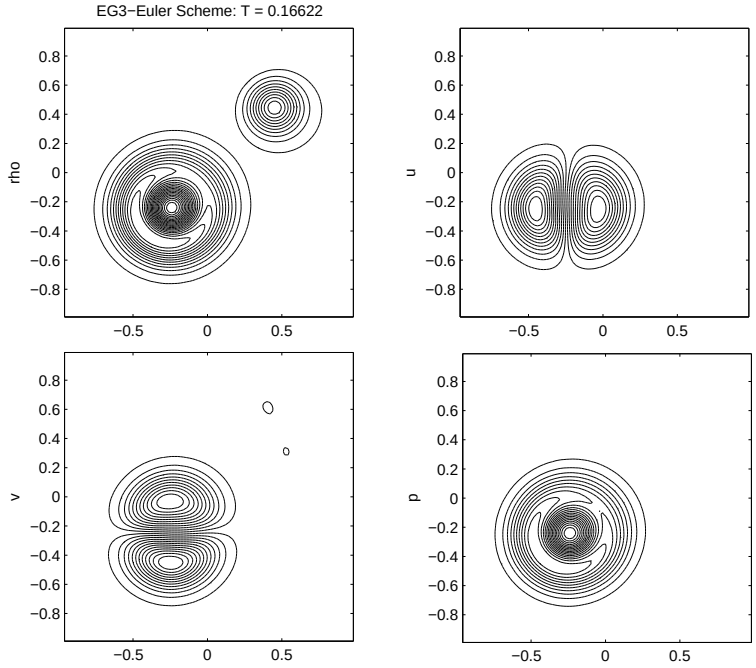

Figure 3.7: First order EG3-Euler Scheme at time $T = 0.166$.

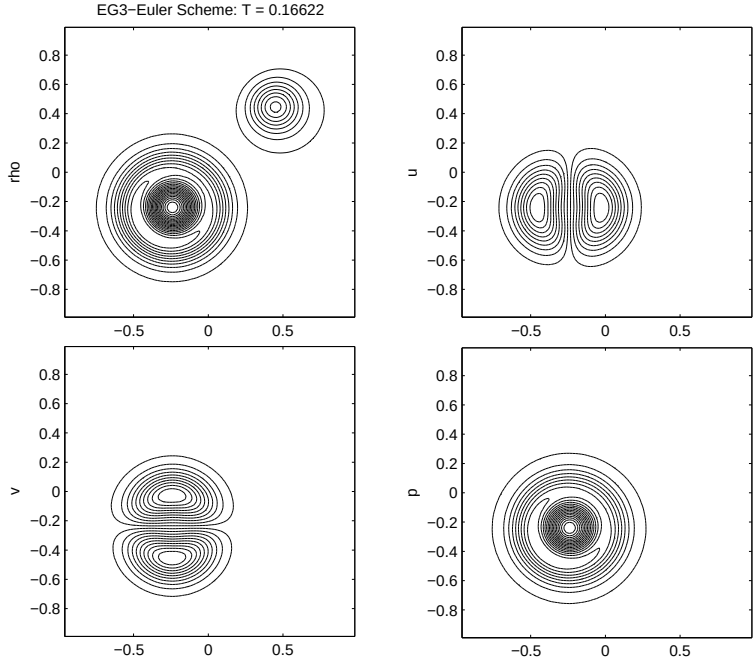

Figure 3.8: Second order EG3-Euler scheme at time $T = 0.166$.

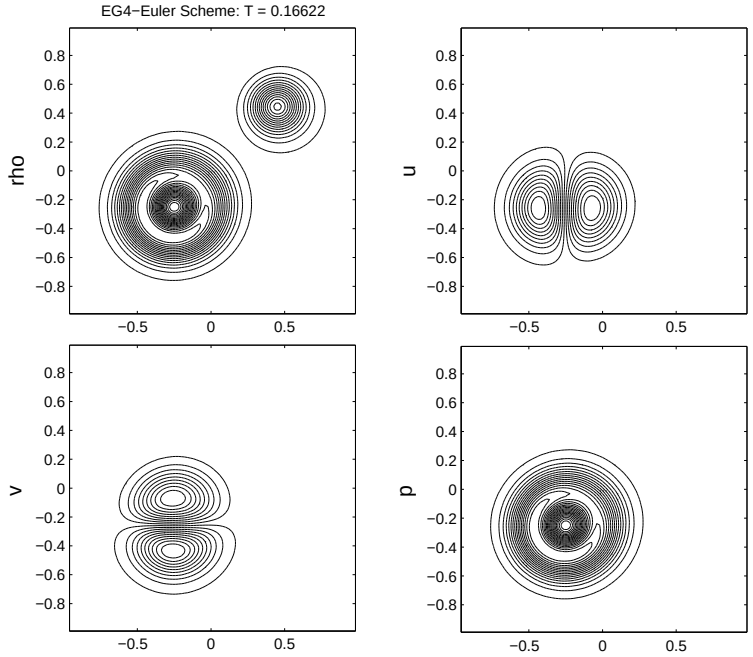

Figure 3.9: First order EG4-Euler Scheme at time $T = 0.166$.

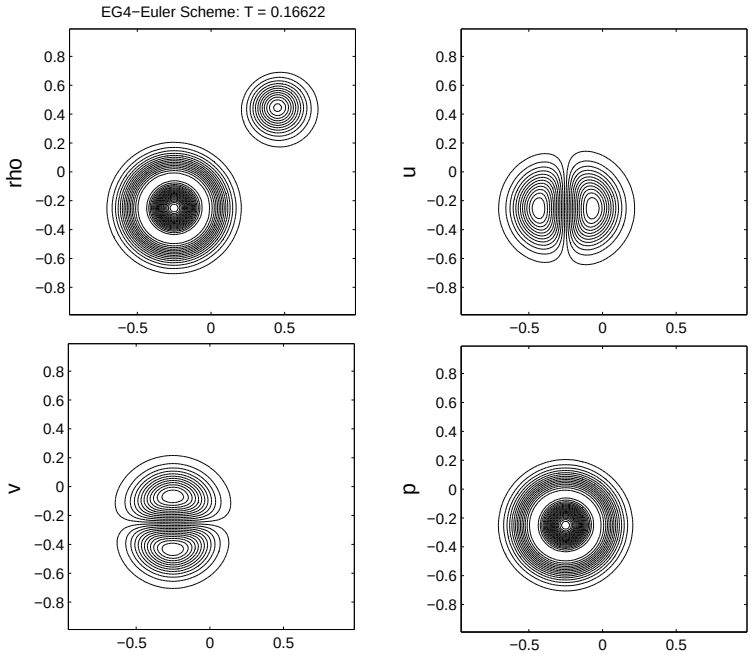

Figure 3.10: Second order EG4-Euler scheme at time $T = 0.166$.

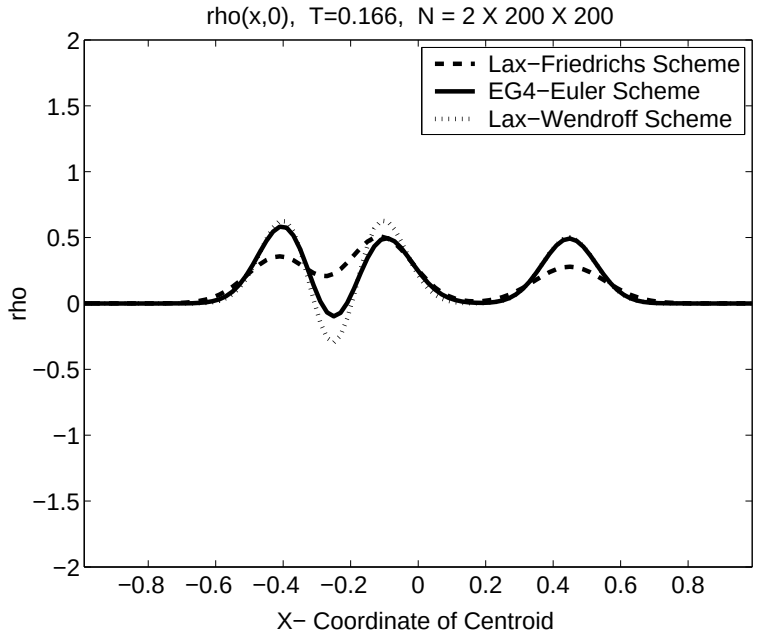

Figure 3.11: Plot of density along the line $y = x$.

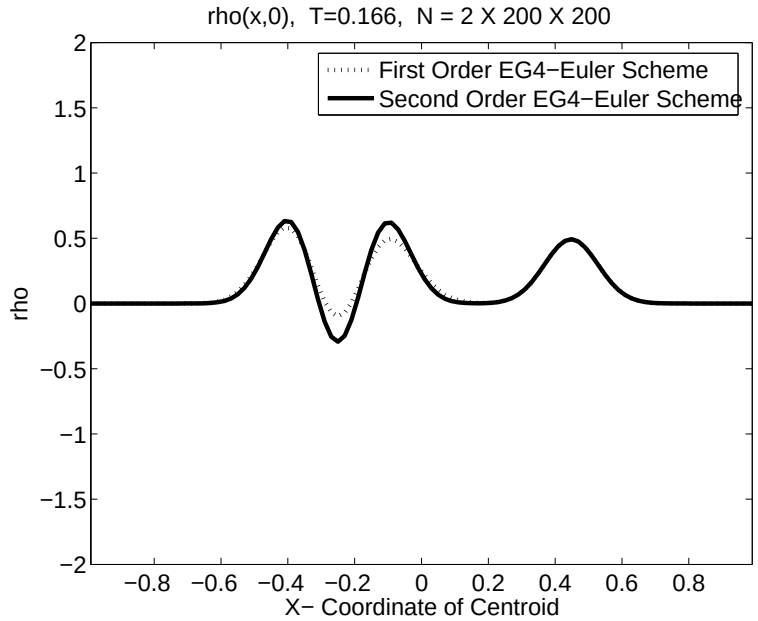

Figure 3.12: Comparison of first order and second order EG4-Euler schemes.

Chapter 4

EG Schemes for the Nonlinear Euler Equations

The EG schemes that have been established in Chapter 3 for linearized Euler equations are extended to the solution of nonlinear Euler equations. In Chapter 3 we have linearized the nonlinear Euler equations by considering small perturbations in density (ρ), velocities (u, v), sound speed (c) and pressure (p). The numerical example demonstrated the propagation of acoustic, entropy and vorticity pulses along the diagonal of the mesh. In that case the local variables stayed constant at all points of the mesh and for all time steps. This introduces a considerable simplification to the nonlinear phenomenon and hence contributes to the error in evaluation of the physical quantities like density, velocities and pressure. Many numerical schemes treating nonlinear Euler equations carry out no linearization for example kinetic schemes, central type schemes, Godunov type upwind schemes and many others. However EG schemes do need some kind of linearization at early stage, see [43]. In this chapter we shall use the same approximate evolution operators as we used for linearized Euler equations, however the local variables will have different values at different quadrature points. This means that each vertex and midpoint of an edge will have a particular value which will be different than the other points. These values will be calculated by an appropriate averaging procedure at these points. In the next time steps these variables will have to be re-calculated at all points. In this way the updated values of the physical quantities are involved in the calculation of local variables at each time step which brings the nonlinear effects into play.

4.1 Numerical Algorithms

In case of the Euler equations, we note that due to the effect of advection the center of the sonic circle which constitutes the base of the Mach cone does not coincide with a vertex or a midpoint in the mesh. This means that any vertex or midpoint will be shifted according to local speeds. Thus the first order algorithm for Euler equations reads

- Input the initial data: ρ_i^0, u_i^0, v_i^0, p_i^0 for the whole mesh.

- Compute the conservative variables.
- Do the time loop
  1. Find the local variables: ρ', u', v', p' at all quadrature points.
  2. Find the centers of the sonic circles using the local velocities.
  3. Find the global maximum of $(|u'|+c', |v'|+c')$.
  4. Find the time step $\Delta t = \frac{h_{\min}\,\mathrm{CFL}}{\max(|u'|+c', |v'|+c')}$.
  5. Compute the intermediate values: $\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}$.
  6. Update the conservative variables:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3}\left(n_{x_{s_{i_j}}}\left[\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}\right] + n_{y_{s_{i_j}}}\left[\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}\right]\right), \tag{4.1}$$

$$(q_1)_i^{n+1} = (q_1)_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\left[\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}(\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}})^2 + \tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\right] + n_{y_{s_{i_j}}}\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}, \tag{4.2}$$

$$(q_2)_i^{n+1} = (q_2)_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}} + n_{y_{s_{i_j}}}\left[\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}(\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}})^2 + \tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\right], \tag{4.3}$$

$$\begin{aligned} E_i^{n+1} = {} & E_i^n - \frac{\Delta t}{|T_i|}\sum_{j=1}^{3} n_{x_{s_{i_j}}}\left[\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}\frac{(\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}})^2 + (\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}})^2}{2} + \tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\left(\frac{\gamma}{\gamma-1}\right)\right]\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}} \\ & + n_{y_{s_{i_j}}}\left[\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}\frac{(\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}})^2 + (\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}})^2}{2} + \tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}\left(\frac{\gamma}{\gamma-1}\right)\right]\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}. \end{aligned} \tag{4.4}$$

  7. Compute the primitive variables.
  8. Apply the boundary conditions.
- End the time loop.

Note that the local variables are not constant. We compute the local variables inside the time loop at each vertex and edge midpoint. This is carried out by using a simple average of the piecewise constant values at the neighbouring elements of that point. In this manner the values of the local variables are different at each vertex or midpoint and in each time cycle these values are computed from the current values. Since the local velocities also vary with time, the time step which is a function of these velocities is also variable i.e. for each time cycle we have a different time step. In case of the linearized Euler equations

the time step was constant because the local velocities were also constant. Furthermore the displacement of the sonic circle is different at each point. This displacement depends upon the values of the local velocities at that point. Likewise the angular contributions from the neighboring elements also vary at each point and each time cycle hence the angles and the number of neighbors intersected are updated with time. The code switches form subsonic to sonic or supersonic situation depending upon the values of the current local velocities at a particular point. The computational time increases considerably because those computations which were carried out once for the linearized Euler equations, are performed in each time cycle. To have second order algorithm we have to carry a recovery stage before applying the approximate evolution operator. We use the same recovery procedure described in 2.5.

- Input the initial data: ρ_i^0, u_i^0, v_i^0, p_i^0.
- Compute the conservative variables.
- **Carry out the recovery stage**.
- Apply the boundary conditions.
- Do the time loop
  1. Find the local variables: ρ', u', v', p' at all vertices and midpoints.
  2. Find new centers of the sonic circles at all vertices and midpoints of the edges.
  3. Find the global maximum of $(|u'| + c', |v'| + c')$ for all quadrature points in the mesh.
  4. Find the time step Δt.
  5. Find the intermediate values: $\tilde{\rho}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{u}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{v}_{s_{i_j}}^{n+\frac{1}{2}}$, $\tilde{p}_{s_{i_j}}^{n+\frac{1}{2}}$.
  6. Update using equations (4.1)-(4.4).
  7. Compute the primitive variables.
  8. Apply the boundary conditions.
  9. **Carry out the recovery stage**.
- End the time loop.

As mentioned in Chapter 2, we use evolution operator to determine the intermediate value $\mathbf{U}^{n+\frac{1}{2}}$, see equation (2.27), at the quadrature points. The piecewise constant $\mathbf{U}^n$ is taken from the close neighbors at the vertex in the following way

$$\mathbf{U}^n = \begin{cases} \mathbf{U}_{i_6}^n & , 0 < \theta < \alpha_1, \\ \mathbf{U}_{i_1}^n & , \alpha_1 < \theta < \alpha_2, \\ \mathbf{U}_{i_2}^n & , \alpha_2 < \theta < \alpha_3, \\ \mathbf{U}_{i_3}^n & , \alpha_3 < \theta < \alpha_4, \\ \mathbf{U}_{i_4}^n & , \alpha_4 < \theta < \alpha_5, \\ \mathbf{U}_{i_5}^n & , \alpha_5 < \theta < \alpha_6, \\ \mathbf{U}_{i_6}^n & , \alpha_6 < \theta < 2\pi. \end{cases}$$

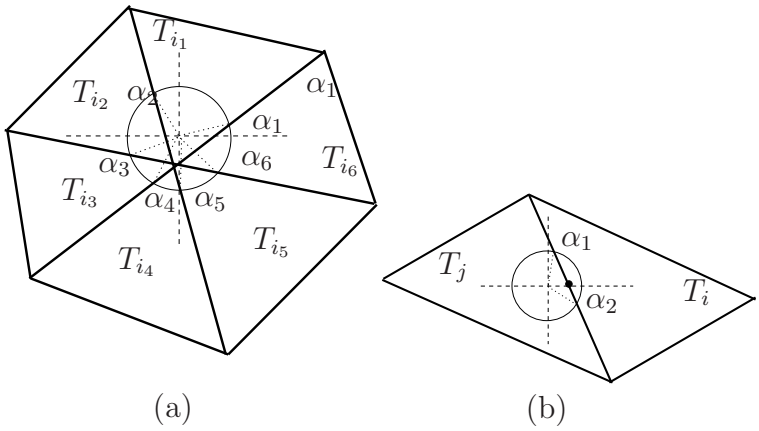

Figure 4.1: Subsonic case: A possible orientation of the sonic circle at (a) vertex (b) midpoint.

See Figure 4.1 (a). The cone base integral of $0 \longrightarrow 2\pi$ splits accordingly

$$0 \longrightarrow \alpha_1 \longrightarrow \alpha_2 \longrightarrow \alpha_3 \longrightarrow \alpha_4 \longrightarrow \alpha_5 \longrightarrow \alpha_6 \longrightarrow 2\pi$$

For midpoint, see Figure 4.1 (b), the piecewise constant $\mathbf{U}^n$ is taken to be

$$\mathbf{U}^n = \begin{cases} \mathbf{U}_i^n & , 0 < \theta < \alpha_1, \\ \mathbf{U}_j^n & , \alpha_1 < \theta < \alpha_2, \\ \mathbf{U}_i^n & , \alpha_2 < \theta < 2\pi. \end{cases}$$

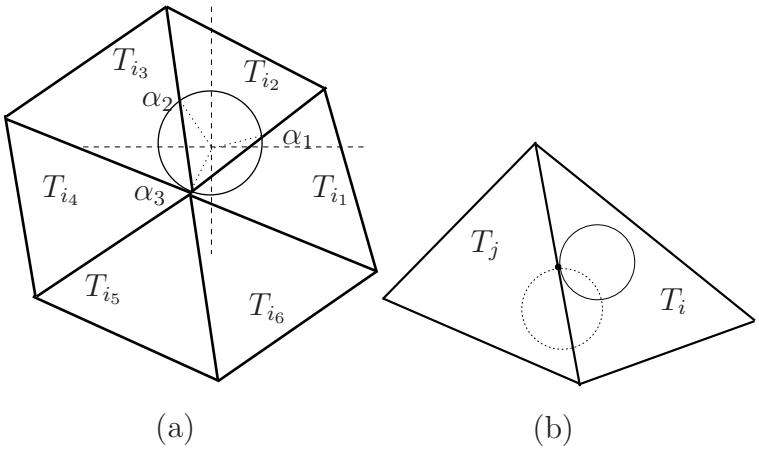

Figure 4.2: Sonic case: Position of the sonic circle at (a) vertex (b) midpoint.

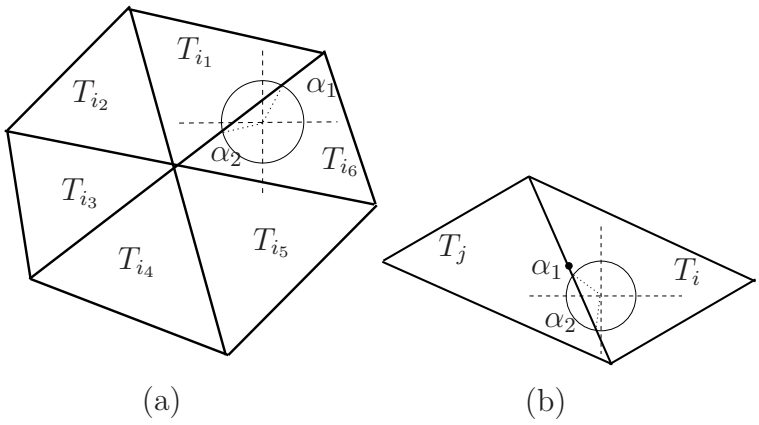

Figure 4.3: Supersonic case: Position of the sonic circle at (a) vertex (b) midpoint.

For the sonic case the displacement of the sonic circle is equal to the radius of the circle. Hence the circle is tangent to the quadrature point, see Figure 4.2, where possible positions of the sonic circle are shown for a vertex and midpoint of the edge. In Figure 4.2 (a), $\mathbf{U}^n$ is distributed in the following manner among different neighbours:

$$\mathbf{U}^n = \begin{cases} \mathbf{U}^n_{i_1} & , 0 < \theta < \alpha_1, \\ \mathbf{U}^n_{i_2} & , \alpha_1 < \theta < \alpha_2, \\ \mathbf{U}^n_{i_3} & , \alpha_2 < \theta < \alpha_3, \\ \mathbf{U}^n_{i_1} & , \alpha_3 < \theta < 2\pi\,. \end{cases}$$

See Figure 4.2 (b), two positions can be seen. For the first position of the circle where it is lying completely inside a triangle, the piecewise constant $\mathbf{U}^n$ is given by

$$\mathbf{U}^n = \left\{ \begin{array}{ll} \mathbf{U}^n_i & , 0 < \theta < 2\pi\,. \end{array} \right.$$

For the second position the circle is shared by two edge neighbours, hence angles are to be computed in this case.

Similarly one can compute $\mathbf{U}^n$ for the supersonic case. A possible position of the sonic circle is shown in 4.3 (a) for a vertex and in 4.3 (b) the midpoint of the edge. There can be many other possible orientations. In this case the advection effect is more pronounced and hence the center of the sonic circle is more displaced from the quadrature point as compared to the previous two cases. The computation of $\mathbf{U}^n$ is more involved because one has to determine the right position of the circle among many possibilities. After finding the exact position of the sonic circle, the angular contribution from the neighbors can be computed.

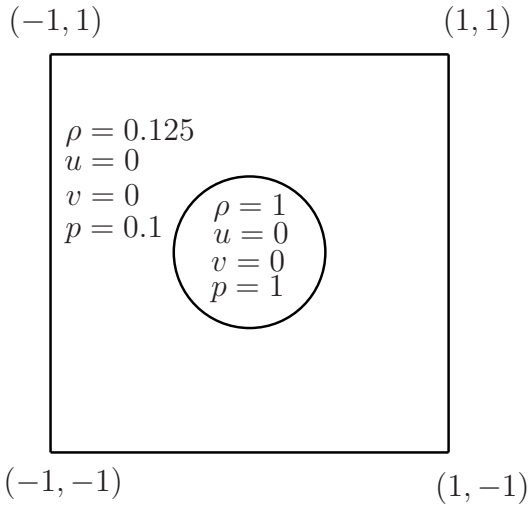

Figure 4.4: Domain for the cylindrical explosion problem.

4.2 Numerical Examples

Example 4.1

We consider the nonlinear Euler equations and a cylindrical explosion problem in a square domain $\Omega = [-1, 1] \times [-1, 1]$ as shown in Figure 4.4. This problem has also been studied in [43] for rectangular meshes using EG schmes. We consider the initial data given in example 2.2. The pressure and density inside the circular region of radius 0.4 is greater than the outside region. This pressure difference generates a shock wave expanding towards the boundary of the domain. A contact discontinuity moves along with the shock while a circular rarefaction wave travels towards the origin at (0,0). We compare the first and second order solutions of this problem in Figures 4.5 and 4.6 using EG4-Euler scheme for a mesh containing $2 \times 200 \times 200$ triangular cells after $T = 0.2$. We use extrapolated boundary conditions in both the first and second order schemes.

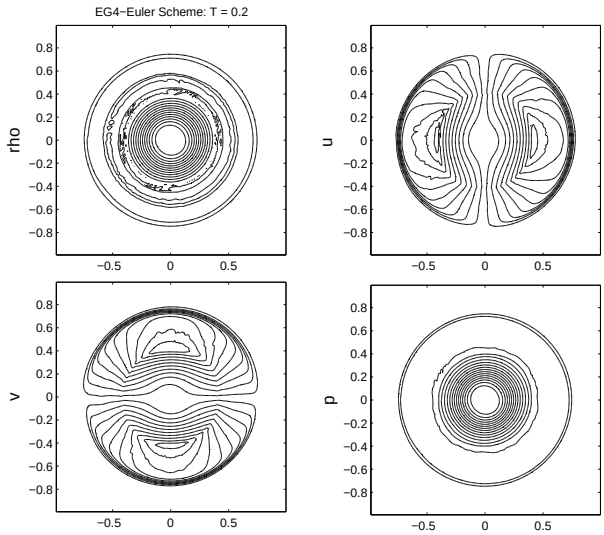

Figure 4.5: First order EG4-Euler scheme.

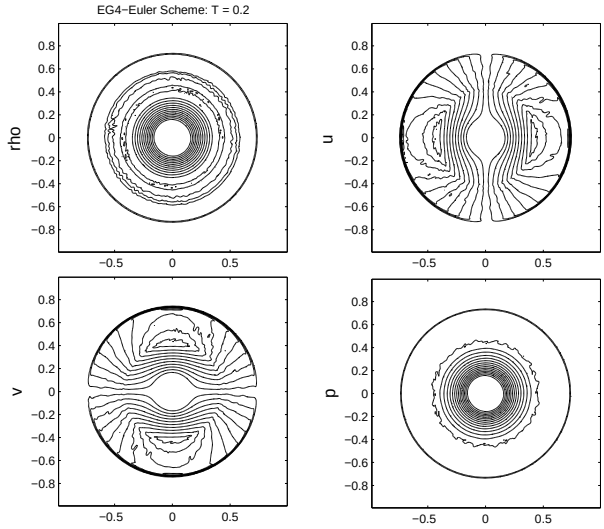

Figure 4.6: Second order EG4-Euler scheme.

Chapter 5

A Slope Propagation Method

In this chapter we propose a new second order scheme which uses space-time control volumes as conservation elements. The scheme is Jacobian free and hence can be applied to any hyperbolic system. We refer to this scheme as the slope propagation method (SP-method). The scheme uses the basic concepts of both, the CE/SE method of Chang et al. [6, 7, 8, 9, 10, 65, 66] and the central schemes of Nessyahu and Tadmor [48].

The central schemes [25, 48] use the staggered grid, well-known MUSCL-type initial reconstruction and the min-mod nonlinear limiter as basic entities to calculate slopes at each time step. These schemes are very simple, efficient and Jacobian-free.

On the other hand the CE/SE method of Chang uses the space-time control elements (CE) as conservation elements and treats the conservative flow variables and their slopes in the same way. The CE/SE method is a family of schemes, i.e., the a scheme, the $a-\epsilon$ scheme, and the $a-\alpha$ scheme. The a scheme derived for linear advection, determines the space-time geometry of the numerical mesh employed. The $a-\epsilon$ and the $a-\alpha$ schemes are extensions of the a scheme for nonlinear equations and for shock capturing. In the a scheme the number of control elements are equal to the number of unknowns. These control elements are used to derive the formulation for the conservative flow variables and their slopes. This scheme has a better resolution and is as efficient as central schemes. The main disadvantage of the CE/SE method is that the calculation of the flux derivatives with respect to time is required and hence the Jacobian matrices are involved in the derivation of the scheme. The space-time derivatives of the fluxes in the scheme come from the assumption of linear variation of the fluxes in space-time. Hence the method is no longer Jacobian-free, which is an important property of the central schemes.

The main aim of our scheme is to retain the advantages of the both the CE/SE method for linear problems, the a scheme, and central schemes. We derive the scheme in a simple and straightforward way by using the basic concepts of finite volume schemes and conservation laws. The scheme uses space-time control volumes in order to compute the conservative flow variables and their slopes. In our scheme, unlike the CE/SE method, we do not as-

sume the space-time linear variation of fluxes in each element. We assume instead the linear variation of the conservative flow variables only. The fluxes are calculated from the flow variables at the midpoint of the faces of the space-time control volumes. We approximate the time integrals of the fluxes by using the midpoint rule. This procedure eliminates the use of Jacobian matrices in our scheme. In the one-dimensional case our staggered mesh stencil is similar to that of central schemes [48] and Chang's method [6]. Note that Breuss [4] showed that a staggered central scheme produces less oscillations at local extrema in the data. In the 2D triangular mesh case we use the same staggered stencil which is used in the CE/SE method [7]. For linear equations, our scheme reduces to the a scheme of Chang [6] and differs from the central schemes since reconstructions are not used.

The main features of the our scheme are as follows:

(i) Space and time are treated in a unified manner.

(ii) The discrete space-time control volumes are the basic conservation regions.

(iii) The derivatives of the dependent variables are also treated as independent variables, even in the case of nonlinear conservation laws.

(iv) The mesh is staggered in time.

(v) A multidimensional scheme is derived on triangular meshes.

(vi) The scheme enjoys the advantages of both central schemes [48] and the CE/SE method [6].

(vii) The second order accuracy of the scheme is verified inh numerical experiments with standard test cases..

5.1 The One-Dimensional SP-Method

In this section we derive the SP-method for one-dimensional hyperbolic systems

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = \mathbf{0}, \quad x \in \mathbb{R}\,, \quad t \in \mathbb{R}^+\,, \tag{5.1}$$

where $\mathbf{u} \in \mathbb{R}^m\,, \quad \mathbf{0} \in \mathbb{R}^m\,, \quad \mathbf{f} : \mathbb{R}^m \longrightarrow \mathbb{R}^m\,, \quad \text{for} \quad m \geq 1\,.$
Here $\mathbf{u}(x,t)$ represent a vector of conserved variables and $\mathbf{f}(x,t)$ is the corresponding vector of fluxes. Let $x_1 = x,$ and $x_2 = t$ be the coordinates of two-dimensional Euclidean space $\mathbb{R}^2$. The integral form of the above equation is given by

$$\oint_{S(V)} \mathbf{h}_k \cdot d\mathbf{S} = 0\,, \quad k = 1, 2, \cdots m\,, \tag{5.2}$$

where S is the boundary of an arbitrary space-time region V in $\mathbb{R}^2$. The space-time vector $\mathbf{h}_k \stackrel{\text{def}}{=} (f_k, u_k)$, where f_k and u_k are the components of vectors $\mathbf{f}$ and $\mathbf{u}$, respectively. Here $d\mathbf{S} = \mathbf{n}d\sigma$, where $d\sigma$ is area of the face of the control volume and $\mathbf{n}$ is an outward unit normal. Equation (5.2) is enforced over a space-time region, called control volume (CV), in which discontinuities of the flow variables are allowed. Here we can see that $\mathbf{h}_k \cdot d\mathbf{S}$ is the space-time flux coming out of the region through the surface element $d\mathbf{S}$.

In the present approach, like the original a scheme of Chang [6], the number of CV's associated with each grid point is identical to the number of unknowns. In one dimension the unknowns are $\mathbf{u}$ and $\mathbf{u}_x$, therefore we need two control volumes to determine them. To proceed we divide the entire computational spatial domain into non-overlapping cells, which are line elements of mesh length Δx in this case, see Figure 5.1. The dashed lines

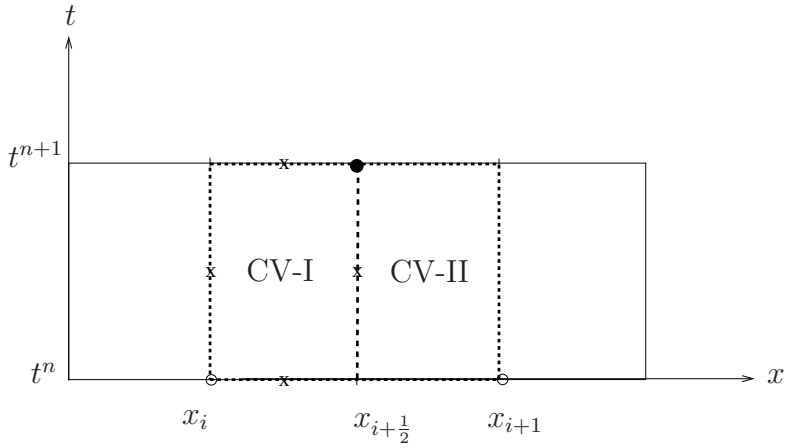

Figure 5.1: Geometrical representation of the control volumes.

indicate the boundaries of the CVs. We introduce the grid points

$$x_i = i \cdot \Delta x \quad \text{for} \quad i \in \mathbb{Z} \quad \text{and} \quad x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}\,.$$

The grid points actually used are denoted by small dots, where the hollow dots represent the grid points at the previous time step, while the filled dot is the node at the updated time. The mesh is staggered in time. Consider the case that $n \in \mathbb{N}_0$ is an even number. At each grid point, we construct two CV's as

$$\text{CV-I} = [x_i, x_{i+\frac{1}{2}}] \times [t^n, t^{n+1}]\,, \quad \text{CV-II} = [x_{i+\frac{1}{2}}, x_{i+1}] \times [t^n, t^{n+1}]\,.$$

For n odd everything is shifted by $\frac{\Delta x}{2}$. Let us denote the exact solutions by vector $\mathbf{u} \in \mathbb{R}^m$ and the numerical solution by $\mathbf{U} \in \mathbb{R}^m$, then we approximate the flow variables in each cell $I_i^n := \{\xi \mid |\xi - x_i| \le \frac{\Delta x}{2}\}$ by

$$\mathbf{U}(x,t) = \sum \left[\mathbf{U}_i^n + (\mathbf{U}_x)_i^n (x - x_i) + (\mathbf{U}_t)_i^n (t - t^n)\right] \chi_i(x)\,. \tag{5.3}$$

Here, $\chi_i(x)$ is the characteristic function of the cell I_i^n, centered around $x_i = i \cdot \Delta x$. Using (5.1), we can rewrite (5.3) as

$$\mathbf{U}(x,t) = \sum \left[\mathbf{U}_i^n + (\mathbf{U}_x)_i^n (x - x_i) - (\mathbf{F}_x)_i^n (t - t^n)\right] \chi_i(x)\,. \tag{5.4}$$

For flows in one spatial dimension using a fixed spatial domain for CV's, the left hand side of (5.2) can be converted into line integral. In vector form it can be represented as

$$\oint_{S(V)} (\mathbf{F}dt - \mathbf{U}dx) = \mathbf{0}\,, \tag{5.5}$$

Here $\mathbf{U}$ and $\mathbf{F}$ are vectors with components U_k and F_k for $k = 1, 2, \cdots, m$, respectively. We apply this equation to the CV-I and compute the flux balance counterclockwise

$$-\int_{x_i}^{x_{i+\frac{1}{2}}} \mathbf{U}(x,t^n)dx + \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_{i+\frac{1}{2}},t))dt + \int_{x_i}^{x_{i+\frac{1}{2}}} \mathbf{U}(x,t^{n+1})dx - \int_{t^n}^{t^{n+1}} \mathbf{F}(\mathbf{U}(x_i,t))dt = \mathbf{0}\,. \tag{5.6}$$

Here the first integral corresponds to the bottom edge and uses the linear function (5.4) at (x_i, t^n). The second integral gives the flux at the right edge with such a linear function at node $(x_{i+\frac{1}{2}}, t^{n+1})$. The third integral corresponds to the top edge using (5.4) at $(x_{i+\frac{1}{2}}, t^{n+1})$ and the last one corresponds to the left edge using (5.4) at (x_i, t^n) of the CV-I. Using (5.4) and the midpoint rule for the time integral in (5.6) we get after dividing by Δx

$$-\left[\mathbf{U}_i^n + \frac{\Delta x}{4}(\mathbf{U}_x)_i^n\right] + 2\lambda \mathbf{F}(\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}) + \left[\mathbf{U}_{i+\frac{1}{2}}^{n+1} - \frac{\Delta x}{4}(\mathbf{U}_x)_{i+\frac{1}{2}}^{n+1}\right] - 2\lambda \mathbf{F}(\mathbf{U}_i^{n+\frac{1}{2}}) = \mathbf{0}\,. \tag{5.7}$$

where $\lambda = \Delta t/\Delta x$. The midpoint value $\mathbf{U}_i^{n+\frac{1}{2}}$ may be predicted via (5.4) from the known data, while $\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ must still be determined. Similarly the flux balance over CV-II gives

$$-\left[\mathbf{U}_{i+1}^n - \frac{\Delta x}{2}(\mathbf{U}_x)_{i+1}^n\right] + \lambda \mathbf{F}(\mathbf{U}_{i+1}^{n+\frac{1}{2}}) + \left[\mathbf{U}_{i+\frac{1}{2}}^{n+1} + \frac{\Delta x}{2}(\mathbf{U}_x)_{i+\frac{1}{2}}^{n+1}\right] - \lambda \mathbf{F}(\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}) = \mathbf{0}\,. \tag{5.8}$$

Now we can find $\mathbf{U}_{i+\frac{1}{2}}^{n+1}$ and $(\mathbf{U}_x)_{i+\frac{1}{2}}^{n+1}$ from the above two equations. Adding (5.7) and (5.8) we get

$$\mathbf{U}_{i+\frac{1}{2}}^{n+1} = \frac{1}{2}(\mathbf{U}_i^n + \mathbf{U}_{i+1}^n) + \frac{\Delta x}{8}\left[(\mathbf{U}_x)_i^n - (\mathbf{U}_x)_{i+1}^n\right] + \lambda\left[\mathbf{F}(\mathbf{U}_i^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{i+1}^{n+\frac{1}{2}})\right]\,. \tag{5.9}$$

Subtracting (5.7) from (5.8), we get

$$\begin{aligned}\Delta x(\mathbf{U}_x)_{i+\frac{1}{2}}^{n+1} =& 2(\mathbf{U}_{i+1}^n - \mathbf{U}_i^n) - \frac{\Delta x}{2}\left[(\mathbf{U}_x)_i^n + (\mathbf{U}_x)_{i+1}^n\right] \\ &- 4\lambda\left[\mathbf{F}(\mathbf{U}_i^{n+\frac{1}{2}}) - 2\mathbf{F}(\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{i+1}^{n+\frac{1}{2}})\right]\,.\end{aligned} \tag{5.10}$$

This will give the update of the slopes. We still need in (5.9) and (5.10) the predicted values $\mathbf{U}_i^{n+\frac{1}{2}}$ and $\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ which can be obtained by using Taylor expansion and by approximating the flux derivative as follows. First we define

$$\mathbf{U}_i^{n+\frac{1}{2}} = \mathbf{U}_i^n + \frac{\Delta t}{2}(\mathbf{U}_t)_i^n = \mathbf{U}_i^n - \frac{\lambda}{2}\Delta\mathbf{F}(\mathbf{U}_i^n)\,, \quad \text{for any } i \in \mathbb{Z}\,. \tag{5.11}$$

The computation of flux differences $\Delta\mathbf{F}_i^n = \Delta\mathbf{F}(\mathbf{U}_i^n)$ can be done by using the min-mod nonlinear limiter, see [25, 48]. In this case for any grid function $\{\mathbf{F}_i^n\}$ and parameter $1 \le \alpha \le 2$ we set

$$\Delta\mathbf{F}_i^n = MM\left(\alpha\Delta_-\mathbf{F}_i^n, \frac{(\Delta_-\mathbf{F}_i^n + \Delta_+\mathbf{F}_i)}{2}, \alpha\Delta_+\mathbf{F}_i^n\right)\,, \quad \Delta_\pm\mathbf{F}_i^n = \pm\left(\mathbf{F}_{i\pm1}^n - \mathbf{F}_i^n\right)\,, \tag{5.12}$$

and MM denotes the min-mod nonlinear limiter

$$MM\{x_1, x_2, ...\} = \begin{cases} \min_i\{x_i\} & \text{if } x_i > 0 \quad \forall i\,, \\ \max_i\{x_i\} & \text{if } x_i < 0 \quad \forall i\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{5.13}$$

Note that (5.11) gives the predicted value at the forward half time step with respect to the data at the initial time step. Next we set

$$\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \mathbf{U}_{i+\frac{1}{2}}^{n+1} - \frac{\Delta t}{2}(\mathbf{U}_t)_{i+\frac{1}{2}}^{n+1} = \mathbf{U}_{i+\frac{1}{2}}^{n+1} + \frac{\lambda}{2}\tilde{\Delta}\mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^{n+1})\,. \tag{5.14}$$

The updated values of the flow variables $\mathbf{U}_{i+\frac{1}{2}}^{n+1}$ have been calculated in (5.9) and our formula (5.14) is only needed in (5.10) afterwards. The approximation (5.14) is the predicted value at the backward half time step with respect to the data at the updated time step. Here

$$\tilde{\Delta}\mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^{n+1}) = MM\left(\alpha\tilde{\Delta}_-\mathbf{F}_{i+\frac{1}{2}}^{n+1}, \frac{(\tilde{\Delta}_-\mathbf{F}_{i+\frac{1}{2}}^{n+1} + \tilde{\Delta}_+\mathbf{F}_{i+\frac{1}{2}}^{n+1})}{2}, \alpha\tilde{\Delta}_+\mathbf{F}_{i+\frac{1}{2}}^{n+1}\right)\,, \tag{5.15}$$

where

$$\tilde{\Delta}_-\mathbf{F}_{i+\frac{1}{2}}^{n+1} = \mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^{n+1}) - \mathbf{F}((\mathbf{U}')_i^{n+1})\,, \quad \tilde{\Delta}_+\mathbf{F}_{i+\frac{1}{2}}^{n+1} = \mathbf{F}((\mathbf{U}')_{i+1}^{n+1}) - \mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^{n+1})\,, \tag{5.16}$$

and

$$(\mathbf{U}')_i^{n+1} = \mathbf{U}_i^n - \lambda\Delta\mathbf{F}(\mathbf{U}_i^n)\,. \tag{5.17}$$

The values $(\mathbf{U}')_i^{n+1}$ are predicted at the updated time step in which $\Delta\mathbf{F}(\mathbf{U}_i^n)$ are calculated from the known initial data through equation (5.12) and MM is the minmod limiter (5.13). The equations (5.9)-(5.17) form a complete scheme.

Like any second order scheme in case of problems with discontinuities we need more dissipation for smooth transition of shocks. Hence we have to limit the slopes given by (5.10). Here we use two different procedures to limit our slopes. In the first procedure we limit directly the slopes calculated in (5.10), while in the second procedure instead of (5.10) we use a central finite difference procedure to calculate the derivatives of the flow variables, [6, 8, 73]. Let us define

$$(V_k)^{n+1}_{i+\frac{1}{2}} = \left((U'_k)^{n+1}_{i+1} - (U_k)^{n+1}_{i+\frac{1}{2}}\right), \quad (W_k)^{n+1}_{i+\frac{1}{2}} = \left((U_k)^{n+1}_{i+\frac{1}{2}} - (U'_k)^{n+1}_{i}\right), \quad \forall\, k = 1, 2, \cdots m\,, \tag{5.18}$$

where for each cell I_i the variables $(U'_k)^{n+1}_i$ are the components of vector $(\mathbf{U}')^{n+1}_i$ in (5.17) and $(U_k)^{n+1}_{i+\frac{1}{2}}$ are the components of vector $\mathbf{U}^{n+1}_{i+\frac{1}{2}}$ at updated time t^{n+1} which are given by (5.9). The slope limiters are given as follows

limiter 1: Direct Procedure for Limiting $\mathbf{U}_x$

For the parameter $1 < \theta < 2$, the discrete slopes at updated time are given by the following expression

$$\Delta x(U_{kx})^{n+1}_{i+\frac{1}{2}} = \Delta x(U_{kx})^{n+1}_{i+\frac{1}{2}} \cdot \varphi\left(\frac{\theta\,(W_k)^{n+1}_{i+\frac{1}{2}}}{\Delta x(U_{kx})^{n+1}_{i+\frac{1}{2}}}\right) \cdot \varphi\left(\frac{\theta\,(V_k)^{n+1}_{i+\frac{1}{2}}}{\Delta x(U_{kx})^{n+1}_{i+\frac{1}{2}}}\right), \tag{5.19}$$

where for each cell I_i $(U_{kx})^{n+1}_{i+\frac{1}{2}}$ are the components of vector $(\mathbf{U}_x)^{n+1}_{i+\frac{1}{2}}$ as given by (5.10) and $\varphi(r)$ is some limiting function, e.g. the minmod function [26] as given by

$$\varphi(r) = \begin{cases} 0\,, & \text{for } r < 0\,, \\ r\,, & \text{for } 0 < r < 1\,, \\ 1\,, & \text{for } r > 1\,. \end{cases} \tag{5.20}$$

Note that the smaller θ is, the smaller are the oscillations, but at the same time, only taking reasonably large θ leads to a uniformly second order accuracy. The optimal θ depends on the problem at hand. However, our experiments indicate that the value of θ between 1.5 and 1.7 is a good choice. This limiter gives oscillations sometimes which is expected since our scheme is still non-dissipative and uniformly second order and hence more oscillatory than the (formally first order) min-mod reconstruction. Therefore for strong shock problems one may need to take further smaller values of the parameter θ.

limiter 2: Central Difference Procedure for $\mathbf{U}_x$

This procedure has also been used for the CE/SE methods in [6, 66, 73]. In this case instead of equation (5.10), we use finite difference procedure to calculate the slopes U_{kx}. Using (5.18) we get the following relation for the slopes at updated time

$$\Delta x \, (U_{kx})_i^{n+1} = \frac{|V_k|^\theta \, W_k \, + \, |W_k|^\theta \, V_k}{|V_k|^\theta \, + \, |W_k|^\theta} \quad \forall \, k = 1, 2, \cdots, m. \tag{5.21}$$

Again θ is adjustable parameter and can be between 1 or 2. However, in our calculation we take $\theta = 1$.

5.1.1 Numerical Algorithm in 1D

We describe the numerical algorithm for the SP-method to solve the one-dimensional hyperbolic equation system

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0 \, .$$

- Discretize the computational domain.
- Take the initial data in each cell I_i^n at time t^n: $\mathbf{U}_i^n$, $(\mathbf{U}_x)_i^n \; \forall \, i \in \mathbb{Z}$.
- Apply the boundary conditions.
- Compute $\mathbf{F}(\mathbf{U}_i^n)$ using the above initial data.
- Compute the time step Δt.
- Do the time loop

  1. Compute the flow variables at half time step i.e. $\mathbf{U}_i^{n+\frac{1}{2}}$.
  2. Compute the flux at half time step: $\mathbf{F}(\mathbf{U}_i^{n+\frac{1}{2}})$.
  3. Compute the predicted values $\mathbf{U}'^{n+1}_i$ required in step 5 and 6.
  4. Update: Obtain $\mathbf{U}_{i+\frac{1}{2}}^{n+1}$ by using equation (5.9).
  5. Compute: $\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ and $\mathbf{F}(\tilde{\mathbf{U}}_{i+\frac{1}{2}}^{n+\frac{1}{2}})$ using (5.14).
  6. Compute the slopes $(\mathbf{U}_x)_{i+\frac{1}{2}}^{n+1}$ using equation (5.10).
  7. Shift the grid by $\frac{\Delta x}{2}$ and assign the updated values to the cells centers in order to use it as initial data for the next time step. This means the mesh is staggered in time.

- Repeat the above procedure for the next time step by updating the values at t^{n+2} with initial data at time t^{n+1}.
- End the time loop after the final time is reached.

5.2 Two-dimensional SP-Method

In this section we derive a two-dimensional SP-method for a regular triangular mesh. A hyperbolic system of conservation laws in two space dimensions is given by

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{0}, \quad (x, y) \in \mathbb{R}^2, \quad t \in \mathbb{R}^+, \tag{5.22}$$

where $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{f} : \mathbb{R}^m \longrightarrow \mathbb{R}^m$, and $\mathbf{g} : \mathbb{R}^m \longrightarrow \mathbb{R}^m$, $m \geq 1$.
Here $\mathbf{u}(x, y, t)$ is the vector of conserved variables and $\mathbf{f}(x, y, t)$ and $\mathbf{g}(x, y, t)$ are the corresponding vectors of fluxes in x and y directions respectively. Let $x_1 = x$, $x_2 = y$ and $x_3 = t$ be the coordinates of a three-dimensional Euclidean space $\mathbb{R}^3$. Once again we use the integral form of conservation law (5.2). Here S is the boundary of the space-time region V in $\mathbb{R}^3$ which we refer to as control volume CV and $\mathbf{h}_k = (f_k, g_k, u_k)$, $d\mathbf{S} = da\,\mathbf{n}$ where da and $\mathbf{n}$ are the area and the outward drawn unit normal respectively of a surface element in S. In two space dimensions we have three unknowns $\mathbf{u}$, $\mathbf{u}_x$ and $\mathbf{u}_y$. Therefore we need three control volumes (CV's) to determine their values. For a triangle Q we construct these CV's from the three close neighbours, each of which shares an edge with triangle Q. In a regular mesh such as that shown in Figure 5.2(a), each lower triangle has all the three neighbours which are upper triangles and vice versa. Thus ΔAGE, ΔAHC and ΔECI are the close neighbours indicated by A_1, A_2 and A_3 respectively as shown in Figure 5.2(a). The bigger dots indicate the centroids of these triangles while the smaller dots indicate the midpoints of the lines which join the centroid of a triangle with the vertex. These points are numbered as A_{jk} where $j = 1, 2, 3$ and $k = 1, 2, 3$. Each control volume is a parallelogram. Parallelograms $AQEA_1A$, AA_2CQA and QCA_3EQ constitute the control volumes CV-I, CV-II and CV-III respectively. The full stencil in space-time is shown in Figure 2(b). Due to staggered grid, we have two consecutive time marching steps. In the first step given the initial data at the lower triangles, we calculate the solution on the upper triangles at updated time step. While in the second marching step we use these newly calculated data on the upper triangles to get the conserved variables at lower triangles at the next updated time step. This procedure of alternating the upper and lower triangles is continued until the final time has arrived. Let us denote the numerical solution vector by $\mathbf{U} \in \mathbb{R}^m$. We can discretize the flow variables by a first order Taylor series expansion as follows

$$\mathbf{U}_Q^n(x, t) = \mathbf{U}_Q^n + (\mathbf{U}_x)_Q^n(x - x_Q) + (\mathbf{U}_y)_Q^n(y - y_Q) + (\mathbf{U}_t)_Q^n(t - t^n). \tag{5.23}$$

The conservation law states that the flux entering a control volume is equal to the flux leaving a control volume. Let us define $\lambda = \frac{\Delta t}{\Delta x}$ and $\mu = \frac{\Delta t}{\Delta y}$, where $\Delta x = x_E - x_C$ and $\Delta y = y_H - y_E$. Considering the control volume CV-I, we get the following equation

$$\begin{aligned} &\mathbf{U}_Q^{n+1} + \frac{\Delta x}{3}(\mathbf{U}_x)_Q^{n+1} + \frac{\Delta y}{6}(\mathbf{U}_y)_Q^{n+1} - \lambda\left[2\mathbf{F}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) + \mathbf{F}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}})\right] \\ &+ \mu\left[\mathbf{G}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) - \mathbf{G}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}})\right] = \mathbf{U}_{A_1}^n - \frac{\Delta x}{3}(\mathbf{U}_x)_{A_1}^n - \frac{\Delta y}{6}(\mathbf{U}_y)_{A_1}^n \\ &- \lambda\left[2\mathbf{F}(\mathbf{U}_{A_{11}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{A_{12}}^{n+\frac{1}{2}})\right] + \mu\left[\mathbf{G}(\mathbf{U}_{A_{11}}^{n+\frac{1}{2}}) - \mathbf{G}(\mathbf{U}_{A_{12}}^{n+\frac{1}{2}})\right]. \end{aligned} \tag{5.24}$$

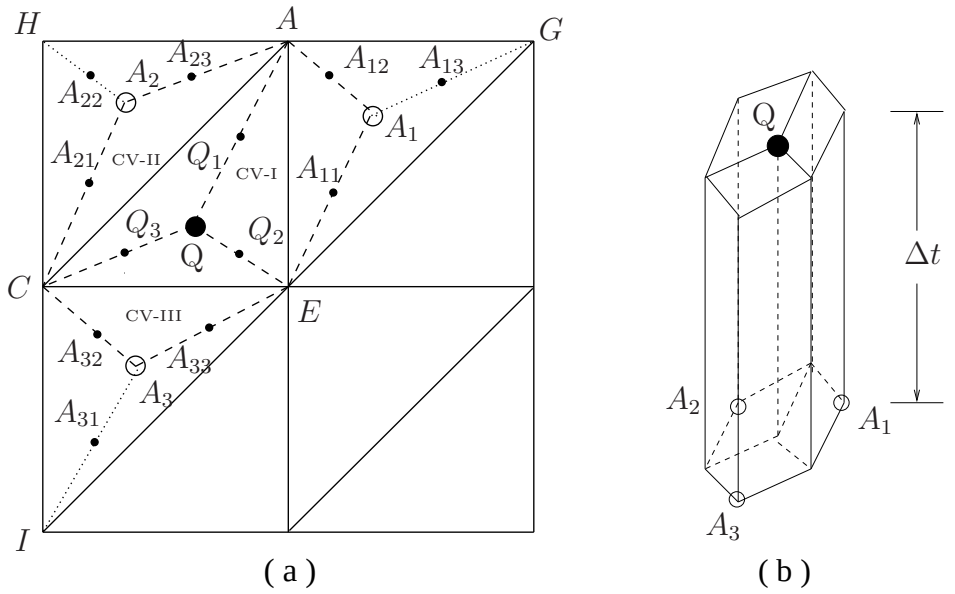

Figure 5.2: Representative triangles and the corresponding control volume.

For the control volume CV-II, the conservation law gives us

$$\begin{aligned}
&\mathbf{U}_Q^{n+1} - \frac{\Delta x}{6}(\mathbf{U}_x)_Q^{n+1} + \frac{\Delta y}{6}(\mathbf{U}_y)_Q^{n+1} + \lambda\left[2\mathbf{F}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) + \mathbf{F}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}})\right] \\
&- \mu\left[\mathbf{G}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) + 2\mathbf{G}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}})\right] = \mathbf{U}_{A_2}^n + \frac{\Delta x}{6}(\mathbf{U}_x)_{A_2}^n - \frac{\Delta y}{6}(\mathbf{U}_y)_{A_2}^n \\
&+ \lambda\left[2\mathbf{F}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}})\right] - \mu\left[2\mathbf{G}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}}) + \mathbf{G}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}})\right].
\end{aligned} \tag{5.25}$$

Applying the flux balance to the control volume CV-III, we obtain

$$\begin{aligned}
&\mathbf{U}_Q^{n+1} - \frac{\Delta x}{6}(\mathbf{U}_x)_Q^{n+1} - \frac{\Delta y}{3}(\mathbf{U}_y)_Q^{n+1} + \lambda\left[\mathbf{F}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}}) - \mathbf{F}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}})\right] \\
&+ \mu\left[\mathbf{G}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}}) + 2\mathbf{G}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}})\right] = \mathbf{U}_{A_3}^n + \frac{\Delta x}{6}(\mathbf{U}_x)_{A_3}^n + \frac{\Delta y}{3}(\mathbf{U}_y)_{A_3}^n \\
&+ \lambda\left[\mathbf{F}(\mathbf{U}_{A_{32}}^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{A_{33}}^{n+\frac{1}{2}})\right] + \mu\left[\mathbf{G}(\mathbf{U}_{A_{32}}^{n+\frac{1}{2}}) + 2\mathbf{G}(\mathbf{U}_{A_{33}}^{n+\frac{1}{2}})\right].
\end{aligned} \tag{5.26}$$

Adding equations (5.24), (5.25) and (5.26), we obtain

$$\begin{aligned}
\mathbf{U}_Q^{n+1} =& \frac{1}{3}(\mathbf{U}_{A_1}^n + \mathbf{U}_{A_2}^n + \mathbf{U}_{A_3}^n) \\
&\pm \frac{\Delta x}{18}\left[(\mathbf{U}_x)_{A_2}^n + (\mathbf{U}_x)_{A_3}^n - 2(\mathbf{U}_x)_{A_1}^n\right] \pm \frac{\Delta y}{18}\left[2(\mathbf{U}_y)_{A_3}^n - (\mathbf{U}_y)_{A_1}^n - (\mathbf{U}_y)_{A_2}^n\right] \\
&\pm \frac{\lambda}{3}\left[2(\mathbf{F}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{A_{11}}^{n+\frac{1}{2}})) + \mathbf{F}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{A_{32}}^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{A_{12}}^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{A_{33}}^{n+\frac{1}{2}})\right] \\
&\pm \frac{\mu}{3}\left[2(\mathbf{G}(\mathbf{U}_{A_{33}}^{n+\frac{1}{2}}) - \mathbf{G}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}})) + \mathbf{G}(\mathbf{U}_{A_{11}}^{n+\frac{1}{2}}) + \mathbf{G}(\mathbf{U}_{A_{32}}^{n+\frac{1}{2}}) - \mathbf{G}(\mathbf{U}_{A_{12}}^{n+\frac{1}{2}}) - \mathbf{G}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}})\right].
\end{aligned} \tag{5.27}$$

Here the + sign indicates the expression for the lower triangle and − sign indicates the expression for the upper triangle. Subtracting (5.25) from (5.24), we get

$$\begin{aligned}\frac{\Delta x}{2}(\mathbf{U}_x)_Q^{n+1} = &\pm (\mathbf{U}_{A_1}^n - \mathbf{U}_{A_2}^n) - \frac{\Delta x}{6}\left[2(\mathbf{U}_x)_{A_1}^n + (\mathbf{U}_x)_{A_2}^n\right] - \frac{\Delta y}{6}\left[(\mathbf{U}_y)_{A_1}^n - (\mathbf{U}_y)_{A_2}^n)\right] \\ &+ \lambda\Big[4\mathbf{F}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) + \mathbf{F}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}}) + \mathbf{F}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}}) - 2(\mathbf{F}(\mathbf{U}_{A_{11}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}})) \\ &-\mathbf{F}(\mathbf{U}_{A_{12}}^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}})\Big] + \mu\Big[\mathbf{G}(\mathbf{U}_{A_{11}}^{n+\frac{1}{2}}) - \mathbf{G}(\mathbf{U}_{A_{12}}^{n+\frac{1}{2}}) + 2\mathbf{G}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}}) \\ &+\mathbf{G}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}}) - 2\mathbf{G}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) + \mathbf{G}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}}) - 2\mathbf{G}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}})\Big]\,. \end{aligned} \tag{5.28}$$

Subtracting (5.25) from (5.26), we get

$$\begin{aligned}\frac{\Delta y}{2}(\mathbf{U}_y)_Q^{n+1} = &\pm (\mathbf{U}_{A_2}^n - \mathbf{U}_{A_3}^n) + \frac{\Delta x}{6}\left[(\mathbf{U}_x)_{A_2}^n - (\mathbf{U}_x)_{A_3}^n\right] - \frac{\Delta y}{6}\left[(\mathbf{U}_y)_{A_2}^n + 2(\mathbf{U}_y)_{A_3}^n\right] \\ &+ \lambda\Big[2\mathbf{F}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}}) - \mathbf{F}(\mathbf{U}_{A_{32}}^{n+\frac{1}{2}}) + \mathbf{F}(\mathbf{U}_{A_{33}}^{n+\frac{1}{2}}) - 2\mathbf{F}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) \\ &-2\mathbf{F}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}}) + \mathbf{F}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}})\Big] - \mu\Big[2\mathbf{G}(\mathbf{U}_{A_{23}}^{n+\frac{1}{2}}) + \mathbf{G}(\mathbf{U}_{A_{21}}^{n+\frac{1}{2}}) + \mathbf{G}(\mathbf{U}_{A_{32}}^{n+\frac{1}{2}}) \\ &+2\mathbf{G}(\mathbf{U}_{A_{33}}^{n+\frac{1}{2}}) - \mathbf{G}(\tilde{\mathbf{U}}_{Q_1}^{n+\frac{1}{2}}) - 4\mathbf{G}(\tilde{\mathbf{U}}_{Q_3}^{n+\frac{1}{2}}) - \mathbf{G}(\tilde{\mathbf{U}}_{Q_2}^{n+\frac{1}{2}})\Big]\,. \end{aligned} \tag{5.29}$$

We still need to calculate the predicted values $\mathbf{U}^{n+\frac{1}{2}}$ and $\tilde{\mathbf{U}}^{n+\frac{1}{2}}$ which can be obtained by using Taylor expansion and by approximating the flux derivative as follows. First we define

$$\mathbf{U}_{A_{jk}}^{n+\frac{1}{2}} = \mathbf{U}_{A_j}^n + (\mathbf{U}_x)_{A_j}^n\,\Delta x_{A_{jk}} + (\mathbf{U}_y)_{A_j}^n\,\Delta y_{A_{jk}} - \frac{\Delta t}{2}\left[\mathbf{F}_x(\mathbf{U}_{A_j}^n) + \mathbf{G}_y(\mathbf{U}_{A_j}^n)\right]\,, \tag{5.30}$$

where $\Delta x_{A_{jk}} = x_{A_{jk}} - x_{A_j}$ and $\Delta y_{A_{jk}} = y_{A_{jk}} - y_{A_j}$ for $j = 1, 2, 3$ and $k = 1, 2, 3$. Here (x_{A_j}, y_{A_j}) are the coordinates of the centroids of the mesh cells and $(x_{A_{jk}}, y_{A_{jk}})$ are the coordinates of the midpoints of the edges of the control volumes, see Figure 5.2(a). For any triangle A_2, consult Figure 5.3, the $x-$component of flux slope $\mathbf{F}_x(\mathbf{U}_{A_2}^n)$ is given by

$$\mathbf{F}_x(U_{A_2}^n) = \left(\mathbf{F}(\mathbf{U}_{A_1}^n) - \mathbf{F}(\mathbf{U}_{A_2}^n)\right)/\Delta x\,. \tag{5.31}$$

Similarly for any triangle A_3 the $y-$component of flux slope $\mathbf{G}_y(\mathbf{U}_{A_3}^n)$ is given by

$$\mathbf{G}_x(U_{A_3}^n) = \left(\mathbf{G}(\mathbf{U}_{A_2}^n) - \mathbf{G}(\mathbf{U}_{A_3}^n)\right)/\Delta y\,. \tag{5.32}$$

Next we define

$$\tilde{\mathbf{U}}_{Q_k}^{n+\frac{1}{2}} = \mathbf{U}_Q^{n+1} + (\mathbf{U}_x)_Q^{n+1}\,\Delta x_{Q_k} + (\mathbf{U}_y)_Q^{n+1}\,\Delta y_{Q_k} + \frac{\Delta t}{2}\left[\mathbf{F}_x(\mathbf{U}_Q^{n+1}) + \mathbf{G}_y(\mathbf{U}_Q^{n+1})\right]\,. \tag{5.33}$$

where $\Delta x_{Q_k} = x_{Q_k} - x_Q$ and $\Delta y_{Q_k} = y_{Q_k} - y_Q$. Here the slopes $(\mathbf{U}_x)_Q^{n+1}$ are not treated implicitly but can be directly calculated using a finite difference approach and a minmod limiter. For that purpose one has to use the updated value $\mathbf{U}_Q^{n+1}$ of the cell under consideration and the predicted values at updated time of the neighbouring cells, as this solpe

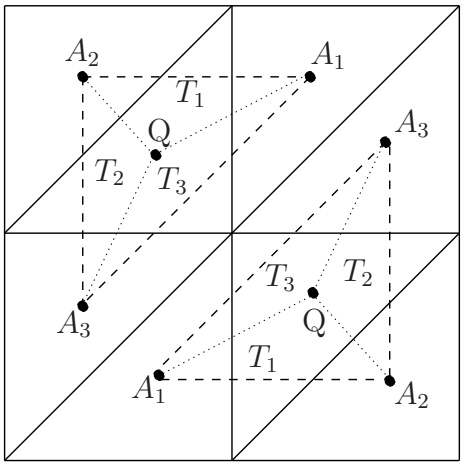

Figure 5.3: Representative triangles and their neighbours.

is only needed at the same cell which is under consideration. The predicted values on the neighbouring cells at updated time are given by

$$\mathbf{U}'^{n+1}_{A_j} = \mathbf{U}^n_{A_j} - \Delta t \left[\mathbf{F}_x(\mathbf{U}^n_{A_j}) + \mathbf{G}_y(\mathbf{U}^n_{A_j}) \right] . \tag{5.34}$$

The x- and y-components of flux slopes $\mathbf{F}_x(\mathbf{U}^{n+1}_Q)$ and $\mathbf{G}_y(\mathbf{U}^{n+1}_Q)$ can be calculated as follow. Let (x_Q, y_Q) be the centroid of the triangle Q under consideration and (x_{A_j}, y_{A_j}) for $j = 1, 2, 3$ be the coordinates of the centroid of the neigbouring triangles A_j, see Figure 5.3. Using the points Q, A_1, A_2, we can write for triangle T_1

$$D := \begin{vmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \end{vmatrix} , \qquad \text{for } \Delta x_j = x_{A_j} - x_Q , \quad \Delta y_j = y_{A_j} - y_Q . \tag{5.35}$$

Then for each components F_k, G_k of vectors $\mathbf{F}$ and $\mathbf{G}$

$$D^{(1)}_x := \begin{vmatrix} \Delta F^{(1)}_k & \Delta y_1 \\ \Delta F^{(2)}_k & \Delta y_2 \end{vmatrix} , \quad D^{(1)}_y := \begin{vmatrix} \Delta x_1 & \Delta G^{(1)}_k \\ \Delta x_2 & \Delta G^{(2)}_k \end{vmatrix} \quad \forall\, k = 1, 2, \cdots m , \tag{5.36}$$

where $m \geq 1$ represents number of equations in the system. Similarly we can use Q, A_2, A_3 for triangle T_2 to find $D^{(2)}_x$, $D^{(2)}_y$ and Q, A_3, A_1 for triangle T_3 to find $D^{(3)}_x$, $D^{(3)}_y$. Here

$$\Delta F^{(j)}_k = F_k(\mathbf{U}'^{n+1}_{A_j}) - F_k(\mathbf{U}^{n+1}_Q), \qquad \Delta G^{(j)}_k = G_k(\mathbf{U}'^{n+1}_{A_j}) - G_k(\mathbf{U}^{n+1}_Q) , \tag{5.37}$$

and

$$F^{(j)}_{kx} = \frac{D^{(j)}_x}{D}, \qquad G^{(j)}_{ky} = \frac{D^{(j)}_y}{D} , \text{ for } \; j = 1, 2, 3. \tag{5.38}$$

Then the components F_{kx} and G_{kx} of flux slopes vectors $\mathbf{F}_x$ and $\mathbf{G}_x$ are given by

$$F_{kx}(\mathbf{U}_Q^{n+1}) = \frac{1}{3}\sum_{j=1}^{3} F_{kx}^{(j)}, \qquad G_{ky}(\mathbf{U}_Q^{n+1}) = \frac{1}{3}\sum_{j=1}^{3} G_{ky}^{(j)}\,. \tag{5.39}$$

To suppress spurious oscillations further, we can modify by a re-weighting procedure

$$F_{kx}(\mathbf{U}_Q^{n+1}) = \frac{\sum\limits_{j=1}^{3} W_k^{(j)} F_{kx}^{(j)}}{\sum\limits_{j=1}^{3} W_k^{(j)}}\,, \quad G_{ky}(\mathbf{U}_Q^{n+1}) = \frac{\sum\limits_{j=1}^{3} W_k^{(j)} G_{ky}^{(j)}}{\sum\limits_{j=1}^{3} W_k^{(j)}}\,, \tag{5.40}$$

where

$$W_k^{(j)} = \prod_{i\neq j,\, i=1}^{3} \theta_{ki}\,, \quad \text{and} \;\; \theta_{ki} = \sqrt{(F_{kx}^{(i)})^2 + (G_{ky}^{(i)})^2} \quad .$$

Note that the weighted average technique applied above is identical to the one used in [7, 8, 73]. We also calculate the components $(U_{kQ})_x^{n+1}$ and $(U_{kQ})_y^{n+1}$ of the slopes vectors $(\mathbf{U}_Q)_x^{n+1}$ and $(\mathbf{U}_Q)_y^{n+1}$ in equation (5.33). For that purpose we use the above procedure in similar manner by just replacing F_k and G_k by U_k. Apart from the above procedure, there are other procedures as well which can be used for the calculation of flux derivatives in each cell, namely by a least square procedure [26].

For problems with discontinuities we have to limit our calculated slopes at updated time in a similar manner as discussed in the one-dimensional case. For that purpose we use similar relations as given in (5.35)-(5.37) with following modifications. Instead of $\Delta F_k^{(j)}$ and $\Delta G_k^{(j)}$ in (5.36) we use $\Delta U_k^{(j)}$, where

$$\Delta U_k^{(j)} = (U_k')_{A_j}^{n+1} - U_{kQ}^{n+1}\,, \quad k = 1, 2, \cdots m\,. \tag{5.41}$$

Here $(U_k')_{A_j}^{n+1}$ and U_{kQ}^{n+1} are the components of the vectors given by (5.34) and (5.27), respectively. The relations in equation (5.38) are replaced by

$$U_{kx}^{(j)} = \frac{D_x^{(j)}}{D}, \qquad U_{ky}^{(j)} = \frac{D_y^{(j)}}{D}\,, \;\text{for} \;\; j = 1, 2, 3. \tag{5.42}$$

Limiter 1: This limiter limits the slopes given by (5.28) and (5.29). Using the relations (5.41) and (5.42) we get the following limited slopes at updated time

$$(U_{kx})_Q^{n+1} = (U_{kx})_Q^{n+1} \cdot \prod_{j=1}^{3} \varphi\left(\frac{\theta\, U_{kx}^{(j)}}{(U_{kx})_Q^{n+1}}\right), \quad \text{for} \quad (U_{kx})_Q^{n+1} \neq 0\,, \tag{5.43}$$

$$(U_{ky})_Q^{n+1} = (U_{ky})_Q^{n+1} \cdot \prod_{j=1}^{3} \varphi\left(\frac{\theta\, U_{ky}^{(j)}}{(U_{ky})_Q^{n+1}}\right), \quad \text{for} \quad (U_{ky})_Q^{n+1} \neq 0\,. \tag{5.44}$$

Here the limiting function $\varphi(r)$ is given by (5.20) and $1 < \theta < 2$ is an adjustable parameter. Normally the factors of $\varphi(r)$ obtained from the contribution of all three triangles T_1, T_2, T_3 in (5.43) and (5.44) give a little bit more dissipative solutions. Hence it is recommended to use only two of them, for example use the triangles T_1 and T_2 for both lower and upper triangles Q as shown in Figure 5.3. This is what we have used in all of our two-dimensional numerical computations.

Limiter 2: Finite Difference Method for Slopes
In this method instead of limiting the original slopes formulas (5.28) and (5.29), a central difference type reconstruction approach is employed to calculate $(U_{kx})_Q^{n+1}$ and $(U_{ky})_Q^{n+1}$. By using equations (5.42) we get:

$$(U_{kx})_Q^{n+1} = \left(\sum_{j=1}^{3} U_{kx}^{(j)}\right)/3\,, \qquad (U_{ky})_Q^{n+1} = \left(\sum_{j=1}^{3} U_{ky}^{(j)}\right)/3\,, \quad \forall\, k = 1, 2, 3, \cdots, m\,. \tag{5.45}$$

For flows with steep gradients or discontinuities, equation (5.45) is modified by re-weighting procedure:

$$(U_{kx})_Q^{n+1} = \frac{\sum\limits_{j=1}^{3} W_k^{(j)} U_{kx}^{(j)}}{\sum\limits_{j=1}^{3} W_k^{(j)}}\,, \quad (U_{ky})_Q^{n+1} = \frac{\sum\limits_{j=1}^{3} W_k^{(j)} U_{ky}^{(j)}}{\sum\limits_{j=1}^{3} W_k^{(j)}}\,, \tag{5.46}$$

where

$$W_k^{(j)} = \prod_{i\neq j,\, i=1}^{3} \theta_{ki}\,, \quad \text{and} \quad \theta_{ki} = \sqrt{(U_{kx}^{(i)})^2 + (U_{ky}^{(i)})^2} \quad .$$

The above modified function is simple and effective to suppress spurious oscillations near shocks. This concludes the formulation of the SP-method for regular triangular mesh 2 shown in Figure A.1.

5.2.1 Numerical Algorithm

We now present the numerical algorithm for solving the two-dimensional wave equation system (2.7) using the SP-method for a triangular mesh.

- Take the initial data on upper triangular elements: $\mathbf{U}_{A_i}^n$, $(\mathbf{U}_x)_{A_i}^n$ and $(\mathbf{U}_y)_{A_i}^n$.
- Do the time loop
    1. Apply the boundary conditions.
    2. Compute time step Δt.
    3. Compute the fluxes $\mathbf{F}(\mathbf{U}_{A_i}^n)$ and $\mathbf{G}(\mathbf{U}_{A_i}^n)$ using the available data $\mathbf{U}_{A_i}^n$.

4. Compute the flux slopes $\mathbf{F}_x(\mathbf{U}^n_{A_i})$ and $\mathbf{G}_y(\mathbf{U}^n_{A_i})$ using $\mathbf{F}(\mathbf{U}^n_{A_i})$ and $\mathbf{G}(\mathbf{U}^n_{A_i})$.
5. Compute the flow variables at half time step: $\mathbf{U}^{n+\frac{1}{2}}_{A_i}$ at the centers of the faces of the space time control volumes.
6. Compute the fluxes at half time step: $\mathbf{F}(\mathbf{U}^{n+\frac{1}{2}}_{A_i})$ and $\mathbf{G}(\mathbf{U}^{n+\frac{1}{2}}_{A_i})$ at the center of the faces of the space-time control volumes.
7. Update: Obtain $\mathbf{U}^{n+1}_Q$ on the upper triangular elements using equations (5.27).
8. Compute the predicted value $\mathbf{U}'^{\,n+1}_{A_i}$ at the centroids required in step 10-12.
9. Compute $\mathbf{F}(\mathbf{U}'^{n+1}_{A_i})$ and $\mathbf{G}(\mathbf{U}'^{n+1}_{A_i})$.
10. Compute the fluxes at half time step: $\tilde{\mathbf{U}}^{n+\frac{1}{2}}_Q$.
11. Compute the fluxes at half time step: $\mathbf{F}(\tilde{\mathbf{U}}^{n+\frac{1}{2}}_Q)$ and $\mathbf{G}(\tilde{\mathbf{U}}^{n+\frac{1}{2}}_Q)$ at the centers of the faces of the space time control volumes with respect to the updated values $\mathbf{U}^{n+1}_Q$.
12. Compute the the slopes $(\mathbf{U}_x)^{n+1}_Q$, $(\mathbf{U}_y)^{n+1}_Q$ from (5.28) and (5.29).
13. Repeat the procedure by alternating the type of mesh elements. This means that we are staggering the mesh with respect to the lower and upper triangular elements.

- End the time loop after the final time is reached.

5.3 Numerical Tests in 1D

5.3.1 1D Wave Equation

In the following example we consider the one-dimensional wave equations in order to test the convergence behavior of the scheme.

Example 5.1 *Smooth initial data*

We implement the SP-method for the following one-dimensional wave equation system

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0\,.$$

$$\mathbf{u} := \begin{pmatrix} \phi \\ u \end{pmatrix}, \qquad \mathbf{f} := \begin{pmatrix} c\,u \\ c\,\phi \end{pmatrix}. \tag{5.47}$$

Where c is a given constant. Let us take the computational domain as $\Omega = [-1, 1]$ and take the following sinusoidal initial data

$$\phi(x, 0)) = -\frac{1}{c}\sin 2\pi x, \;\; u(x, 0) = 0\,.$$

The exact solution is given by

$$\phi(x,t) = -\frac{1}{c}\cos(2\pi ct)\sin(2\pi x)\,,$$
$$u(x,t) = \frac{1}{c}\sin(2\pi ct)\cos(2\pi x)\,.$$

In Table 5.1 and 5.2, we demonstrate respectively the L^1 and L^2-errors between the exact and numerical solutions, the experimental order of convergence (EOC) and the effect of grid refinement on the accuracy of the numerical solution. The scheme gives us second order accuracy which is evident from the last column where EOC of 2 is easily obtained. In Figure 5.4, the plot for components ϕ and u are shown for 40 mesh points for absolute time $T = 0.2$ using exact boundary conditions at the two boundaries. The scheme has a second order accuracy and a very good agreement with the exact solution even for this smaller number of points.

5.3.2 1D Euler Equations

Here we want to solve the one-dimensional Euler equations given by

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0\,, \tag{5.48}$$

where

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix}\,,\ \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ u(E+p) \end{pmatrix}\,, \tag{5.49}$$

where ρ is the density, u is the velocity, p the pressure and $E = \frac{p}{(\gamma-1)} + \frac{1}{2}\rho\,(u^2)$ is the total energy. The CFL condition is given by

$$\Delta t \leq \Delta x \left(\frac{1}{\max|u| + c}\right)\,, \tag{5.50}$$

where the speed of sound is $c = \sqrt{\gamma \frac{p}{\rho}}$. In the following examples for Euler equations we have tested both kinds of limiters for the slopes. The plots are given for both limiters and one can see that for all examples, limiter 2 is more dissipative as compared to limiter 1. Although in some tests the plots with limiter 1 show small oscillations, however limiter 1 has better resolution than limiter 2. In all figures the left hand plots indicate the results with limiter 1, while the right hand side are with limiter 2.

Example 5.2 *Collision of two shocks*

This problem [64] describes the collision of two shocks. The initial data are

$$(\rho, u, p) = \begin{cases} (5.99924, 19.5975, 460.94) & \text{if } x < 0.4\,, \\ (5.99242, -6.19633, 46.0950) & \text{if } x \geq 0.4\,. \end{cases}$$

Where the spatial domain is $0 \leq x \leq 1$. Here $\gamma = 1.4$, the number of mesh points is 200 and the final time is $T = 0.035$ with CFL= 0.4. We employ extrapolated boundary conditions. The results for both limiters of the SP-method are shown in Figure 5.5. Here we have used the parameter $\theta = 1.6$ for limiter 1 while $\theta = 1$ for limiter 2. The numerical solution using SP-method has been compared with exact Riemann solution and the central scheme [48].

Example 5.3 *Sod problem*

This Riemann problem was proposed by Sod [60]. The initial data are

$$(\rho, u, p) = \begin{cases} (1.0, 0.0, 1.0) & \text{if } x < 0.5\,, \\ (0.125, 0.0, 0.1) & \text{if } x \geq 0.5\,. \end{cases}$$

Where the spatial domain is $0 \leq x \leq 1$. We choose $\gamma = 5/3$, the number of mesh points as 200 and the final time is $T = 0.25$ with CFL= 0.4. At the two boundaries outflow boundary conditions are used. The results of SP-method are compared in Figure 5.6 with the central scheme [48] for the same mesh points and other numerical parameters while the exact solution is a second order central scheme on 4000 mesh points. Here we have used the parameter $\theta = 1.5$ for limiter 1 while $\theta = 1$ for limiter 2.

Example 5.4

This problem contains a shock wave originated due to a difference of pressure on the left and right side of a diaphragm separating two regions. We use 200 mesh points and take $\gamma = 5/3$. The initial data are

$$(\rho, u, p) = \begin{cases} (1.0, -19.59745, 1000) & \text{if } x < 0.8\,, \\ (1.0, -19.59745, 0.01) & \text{if } x \geq 0.8\,. \end{cases}$$

Where the spatial domain is $0 \leq x \leq 1$ and the final time is $T = 0.012$ with CFL= 0.3. The resulting density, velocity and pressure profiles are shown in Figure 5.7. Here we have used the parameter $\theta = 1.7$ for limiter 1 while $\theta = 1$ for limiter 2. The exact solution is a second order central scheme [48] on 4000 mesh points.

Example 5.5 *Blast wave problem*

In this example, we test the scheme for the blast wave problem which was carefully studied by Woodward and Colella [69]. The initial data are taken as

$$(\rho, u, p) = \begin{cases} (1.0, 0.0, 1000.0) & \text{if } x < 0.1\,, \\ (1.0, 0.0, 0.01) & \text{if } 0.1 \leq x < 0.9\,, \\ (1.0, 0.0, 100.0) & \text{if } 0.9 \leq x \leq 1.0\,. \end{cases}$$

The reflective boundary conditions are applied at both $x = 0$ and $x = 1$. We choose $\gamma = 1.4$, 500 mesh points and CFL= 0.4. The results are shown in Figure 5.8 at time is $T = 0.0385$. The exact solution is a second order central scheme on 4000 mesh points. Again the results of our scheme are better resolved as compared to the central scheme [48]. Here we use reflected boundary conditions. Again we have used the parameter $\theta = 1.7$ for limiter 1 while $\theta = 1$ for limiter 2.

5.4 Numerical Tests in 2D

In this section we present two-dimensional numerical test cases.In the 2D case we use regular triangular mesh cells. Hence if $N_x \times N_y$ are the total number of rectangles in our computational domain then $2 \times N_x \times N_y$ are the number of triangles in the same domain of computation. However, due to staggering in time, we are using half of the triangles at each time step. This means that at each time step the updated solution is either available on upper or lower triangles. For a regular grid the number grid points at which solution is available at each time step are the same as the number of rectangles.

5.4.1 2D Wave Equation

In the following example we consider the two-dimensional wave equations in order to test the experimental order of convergence (EOC) of the scheme.

Example 5.6 *Smooth initial data*

Let us take the computational domain as $\Omega = [-1, 1] \times [-1, 1]$ and consider the initial data of Example 2.7 for two-dimensional wave equation system (2.7). We implement exact boundary conditions in this example and take end time T = 0.2 and CFL = 0.4. Tables 5.3 and 5.4 show the L^1 and L^2 errors between the exact and the approximated solutions using the SP-method for a regular triangular mesh 2 shown in Figure A.1. The first column of the tables show the number of triangular elements in the mesh. If N_x and N_y are the number of rectangular cells along x and y-axis respectively, in a regular triangular mesh then $2N_xN_y$ will be the number of triangular cells in the mesh. However, as mentioned above we are using half of them at each time step. Other columns indicate the error for individual solution components and the combined error. The last column of both the tables shows the experimental order of convergence (EOC) for the scheme which is 2 indicating that the scheme is truly a second order scheme. The components u and v retain their symmetry in both L^1 and L^2-errors for this example. If we compare the SP-method with EG scheme, see tables 5.4, 2.4 and 2.2, we notice that the combined error of SP-method is considerably lower than EG schemes and that SP-method gives correct EOC. These tables show the effect of grid refinement on the accuracy of the solution which increases as the number of mesh points increase.

In Figure 5.9, the centerline profiles of the solution components ϕ and u are shown with SP-method and the finite volume EG4 scheme for $2 \times 40 \times 40$ triangular cells. The component v is a function of y coordinate like the component u which is a function of x. Therefore the plot for v with respect to y has not been shown as it is similar to that of u with respect to x. The exact solution is also plotted along with the numerical solution for different mesh sizes. The numerical solution with SP-method is in good agreement with the exact solution and has better accuracy than the EG4 scheme. However at maxima and minima of the solution, the scheme gives more dissipation as compared to other regions. The graph shows that the error in component ϕ is greater than u. This can also be seen from the above tables that the error in u is less than half of the error in ϕ.

5.4.2 2D Euler Equations

In this section we consider few numerical test cases for the Euler equations which are available in the literature.

Example 5.7 *A two-dimensional quadratic pulse*

We study two-dimensional test problems for the Euler equations of gas dynamics given by equation (3.2). We take a mesh consisting of 300×300 triangles and consider a CFL equal to 0.3. It is an initial value problem studied by Dreyer et al. [15] and Qamar [53]. The domain consists of an outer square box (1×1) units and a smaller inner box placed at position $0.4 \leq x \leq 0.6$, $0.4 \leq y \leq 0.6$. Initially the velocities are zero. The particle density and pressure are equal to 4 inside the box and equal to 1 elsewhere. The pressure and density gradients stimulate the propagation of shock waves which travel from the center to the boundary of the region. The profiles of the physical quantities can be seen in Figure 5.10 after time $T = 0.1$.

Example 5.8 *Interaction of four shocks*

This is a two-dimensional Riemann problem with four shocks. We choose the initial data of the following form

$$(\rho, u, v, p) = \begin{cases} (1.1, 0.0, 0.0, 1.1) & \text{if } x > 0.5, \quad y > 0.5\,, \\ (0.5065, 0.8939, 0.0, 0.35) & \text{if } x < 0.5, \quad y > 0.5\,, \\ (1.1, 0.8939, 0.8939, 1.1) & \text{if } x < 0.5, \quad y < 0.5\,, \\ (0.5065, 0.0, 0.8939, 0.35) & \text{if } x > 0.5, \quad y < 0.5\,, \end{cases}$$

which shows that a left forward shock, a right backward shock, an upper backward shock and a lower forward shock move towards the center of the region. We consider a square region $\Omega = [0, 1] \times [0, 1]$. We take the adiabatic constant $\gamma = 1.4$, CFL $= 0.3$ with 300×300 triangles in the mesh. The profiles of particle density, velocities and pressure are shown in Figure 5.11 after time $T = 0.25$. Here we employ outflow boundary conditions.

Example 5.9 *Explosion in a square box*

In this example, we consider a 2-D Riemann problem inside a square box (2×2) units with reflecting walls. Initially the density is unity everywhere and the velocities are zero. The pressure is equal 1000 inside a square (0.5×0.5) units in the center of the box and equal to 10 elsewhere. The geometrical representation of the initial data is shown in Figure 5.12. The solution is computed using SP-method with a mesh consisting of 300×300 grid points with reflected boundary conditions at the four boundaries. The snapshots of temperature, density, kinetic energy, and pressure are shown in Figures 5.13 and 5.14 for end times $T = 0.03$ and $T = 0.1$.

Example 5.10

Here we numerically solve 2D-Riemann problem for the Euler equations, subject to the initial data

$$(\rho, u, v, p) = \begin{cases} (0.5313, 0.0, 0.0, 0.4) & \text{if } x > 0.5, \quad y > 0.5\,, \\ (1.0, 0.7276, 0.0, 1.0) & \text{if } x < 0.5, \quad y > 0.5\,, \\ (0.8, 0.0, 0.0, 1.0) & \text{if } x < 0.5, \quad y < 0.5\,, \\ (1.0, 0.0, 0.7276, 1.0) & \text{if } x > 0.5, \quad y < 0.5\,, \end{cases}$$

which shows right and top forward shock waves and left and lower contact waves. We consider a square region $\Omega = [0, 1] \times [0, 1]$ and apply outflow boundary conditions at all boundaries. We take the adiabatic constant $\gamma = 1.4$. The profiles of particle density, velocities and pressure are shown in Figure 5.11 after time $T = 0.25$.

Example 5.11 *Interaction of two spherically symmetric fields*

This example was also studied by Dreyer et al. [15]. Here we consider the interaction of two spherically symmetric fields with initial data: $\rho_0(x_1, x_2) = 4$, $p_0(x_1, x_2) = 4$ for $(x_1 - 0.4)^2 + (x_2 - 0.4)^2 \leq 0.015$ and for $(x_1 - 0.6)^2 + (x_2 - 0.6)^2 \leq 0.015$. Otherwise $\rho(x_1, x_2) = 1$, $p_0(x_1, x_2) = 1$. Where velocities are zero everywhere, i.e. $u = v = 0$. Our computational domain is $(x, y) \in [0, 1] \times [0, 1]$. We employ outflow boundary conditions and plot the solution after time $T = 0.15$ shown in Figure 5.15.

Example 5.12 *Shock bubble interaction*

The simulations in this example show the interactions between a planar shock and various heterogeneities. The inspiration for this examples is taken from the 3-D shock-bubble interaction example used by Langseth and LeVeque [30] to illustrate the induced vorticity and mixing when a shock wave runs through an inhomogeneous media. The setup is as follows. A bubble with radius 0.2 lies at rest at (0.4,0.5) in the domain $[0, 1.6] \times [0, 1]$. The gas is also at rest initially and has unit density and pressure. Inside the bubble the density is 0.1 while the pressure and velocities have similar values as outside. The incoming

shock wave starts at $x = 0.1$ and propagates in the positive x-direction. Behind the shock the density is 3.81, pressure is 10, x-velocity is 2.85 and y-velocity is 0. The results of SP-method are shown in Figure 5.16. Here we use limiter 1 for slopes computation. We have used the reflecting boundary conditions on the top and lower boundary. The right boundary has outflow boundary conditions while the left has inflow boundary conditions.

Example 5.13

This is a two-dimensional Riemann problem with initial data

$$(\rho, u, v, p) = \begin{cases} (1.5, 0.0, 0.0, 1.5) & \text{if } x > 0.8, \quad y > 0.8\,, \\ (0.53226, 1.206045, 0.0, 0.30) & \text{if } x < 0.8, \quad y > 0.8\,, \\ (0.13799, 1.206045, 1.206045, 0.02903) & \text{if } x < 0.8, \quad y < 0.8\,, \\ (0.53226, 0.0, 1.206045, 0.30) & \text{if } x > 0.8, \quad y < 0.8\,, \end{cases}$$

which shows four 1D shocks separating the four regions of constant states. The computational domain is a square region $\Omega = [0, 1] \times [0, 1]$ with 300×300 triangles and outflow boundary conditions at all the four sides. We take the adiabatic constant $\gamma = 1.4$. The results of SP-method are shown in Figure 5.17 after time $T = 0.8$. In this case we also use limiter 1 for the computation of slopes. The results of the SP-method agrees with the published results given in [58]. Here we can also see oscillations in the results of our scheme which are little more what one can see in [58]. However our results are still quite good and shocks are well resolved.

5.5 Conclusions and Remarks

In this chapter we have presented a second order scheme which treats the space and time in a unified manner for the numerical solution of hyperbolic systems. The flow variables and their slopes are the basic unknowns in the scheme. Unlike the CE/SE method of Chang [6] the present scheme is Jacobian-free and hence can also be applied to any hyperbolic system. By introducing a suitable limiter for the slopes of flow variables, we apply the same scheme to linear and non-linear hyperbolic systems with discontinuous initial data. However, in the Chang's scheme they used a finite difference approach for the slope calculation in the case of nonlinear equations with discontinuous initial data. This is an important property of our scheme to maintain the original structure of the scheme for both linear and non-linear hyperbolic equations. The scheme is simple, efficient and has good resolution. We have derived the scheme for one and two space dimensions. In two-space dimension we have used the triangular mesh elements. The second order accuracy of the scheme has been verified by numerical experiments. Several numerical test computations presented validate the accuracy and robustness of the present scheme. The accuracy of SP-method is considerably beeter than EG schemes. EG-schemes do not give EOC of 2 for a triangular mesh while in case of SP-method we get a correct second order accuracy. In addition the computational cost of SP-method is much lower than EG schemes. The numerical implementation of EG schemes is extremely involved as compared to SP-method.

Table 5.1: Accuracy of SP-Method for the 1D linear wave equations

Nr.	L^1-error	EOC	L^2-error	EOC
50	0.00481096430736	-	0.00273908364264	-
100	0.00120490231058	1.9974	0.00068781448579	1.9936
200	0.00030290652968	1.9920	0.00017262756034	1.9944
400	0.00007576456684	1.9993	0.00004322892390	1.9976
800	0.00001893984315	2.0001	0.0000108140957	1.9991
1600	0.00000473450584	2.0001	0.00000270410810	1.9997
3200	0.00000118321200	2.0005	0.00000067604475	2.0000

Table 5.2: Accuracy of SP-Method for the 2D linear wave equations

Nr. of triangles	L^1-error	EOC	L^2-error	EOC
20×20	0.04493444962950	-	0.02579779157879	-
40×40	0.01237596902600	1.8603	0.00700051718312	1.8817
80×80	0.00338629370250	1.8698	0.00192194244088	1.8649
160×160	0.00089572111750	1.9186	0.00051620335388	1.8966
320×320	0.00022897924300	1.9678	0.00013422385207	1.9433
640×640	0.00005765645550	1.9897	0.00003418771697	1.9731

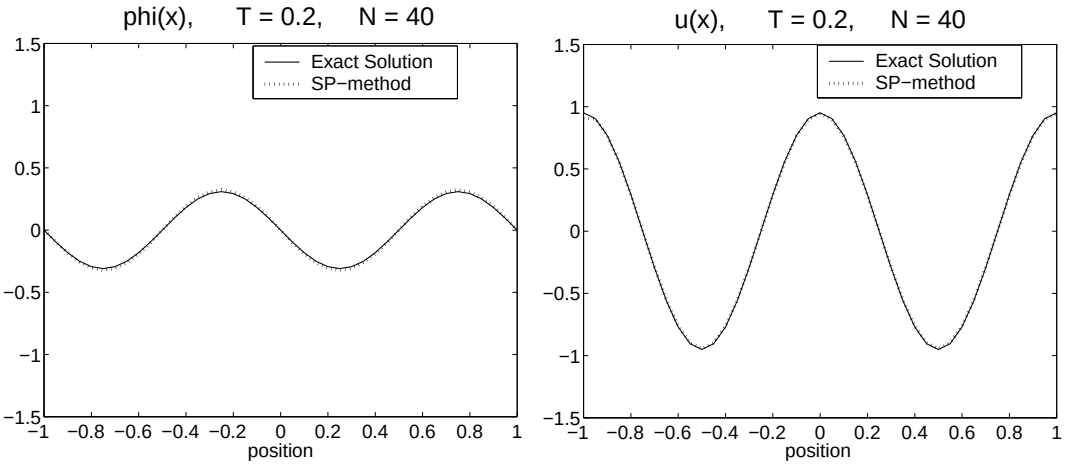

Figure 5.4: One-dimensional SP-method at $T = 0.2$.

Table 5.3: L^1-error for two-dimensional SP-method for triangular mesh.

N	$\|\phi(T) - \phi^n\|_{L^1}$	$\|u(T) - u^n\|_{L^1}$	$\|v(T) - v^n\|_{L^1}$	$\|\mathbf{U}(T) - \mathbf{U}^n\|_{L^1}$	EOC
$2 \times 20 \times 20$	0.0460775847	0.0218956572	0.0218956572	0.0898688992	
$2 \times 40 \times 40$	0.0132187756	0.0057665811	0.0057665811	0.0247519380	1.8603
$2 \times 80 \times 80$	0.0037436771	0.0015144551	0.0015144551	0.0067725874	1.8698
$2 \times 160 \times 160$	0.0010350401	0.0003782010	0.0003782010	0.0017914422	1.9186
$2 \times 320 \times 320$	0.0002742441	0.0000918571	0.0000918571	0.0004579584	1.9678
$2 \times 640 \times 640$	0.0000705335	0.0000223896	0.0000223896	0.0001153129	1.9897

Table 5.4: L^2-error for two-dimensional SP-method for triangular mesh.

N	$\|\phi(T) - \phi^n\|_{L^2}$	$\|u(T) - u^n\|_{L^2}$	$\|v(T) - v^n\|_{L^2}$	$\|\mathbf{U}(T) - \mathbf{U}^n\|_{L^2}$	EOC
$2 \times 20 \times 20$	0.0307315084	0.0139037133	0.0139037133	0.0364835867	
$2 \times 40 \times 40$	0.0085967008	0.0034721186	0.0034721186	0.0099002263	1.8817
$2 \times 80 \times 80$	0.0024079938	0.0008914288	0.0008914288	0.0027180370	1.8649
$2 \times 160 \times 160$	0.0006596862	0.0002210722	0.0002210722	0.0007300217	1.8966
$2 \times 320 \times 320$	0.0001739763	0.0000536857	0.0000536857	0.0001898211	1.9433
$2 \times 640 \times 640$	0.0000446590	0.0000130990	0.0000130990	0.0000483487	1.9731

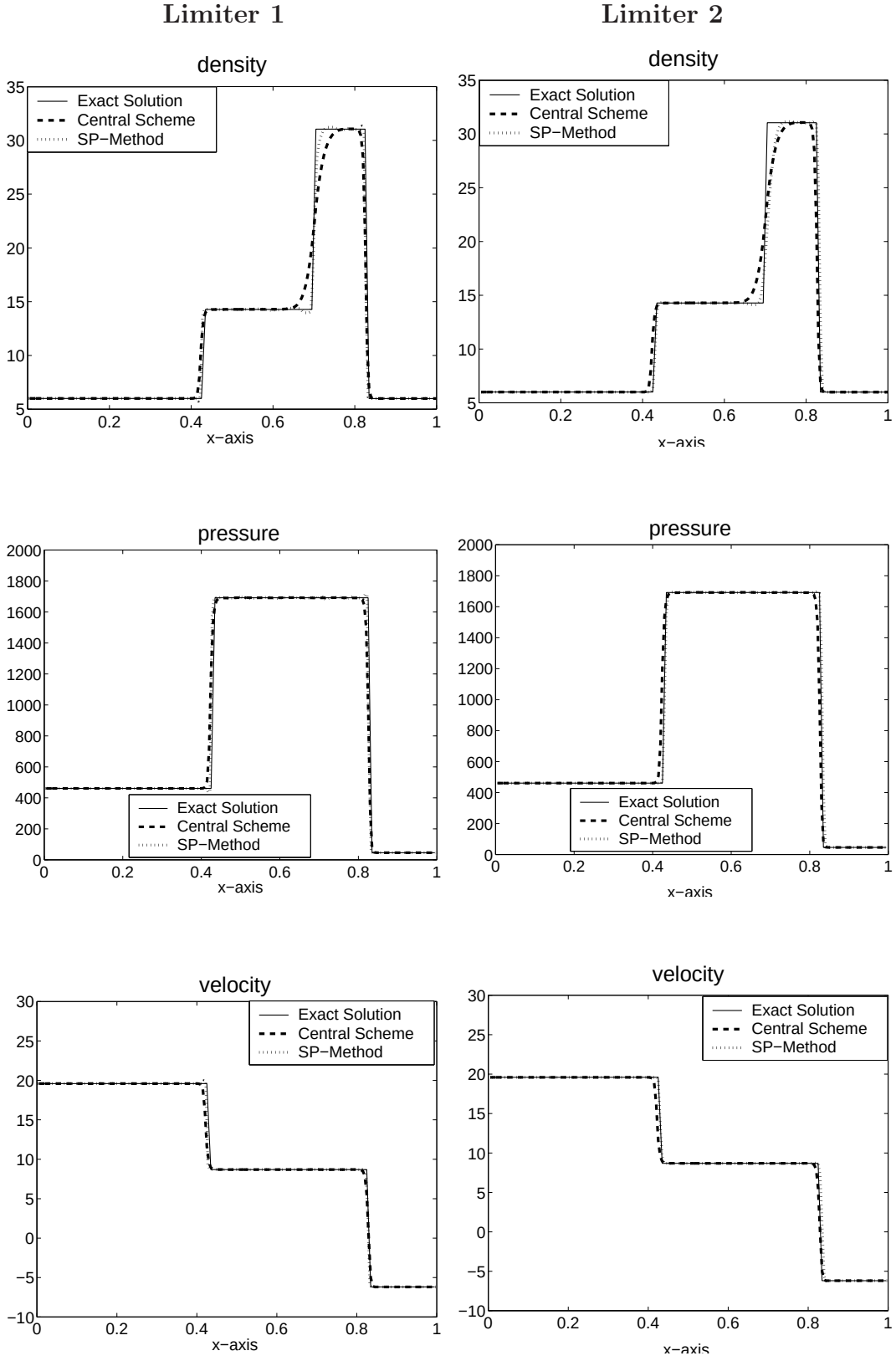

Figure 5.5: Comparison of different limiters for Example 5.2 for 200 mesh points.

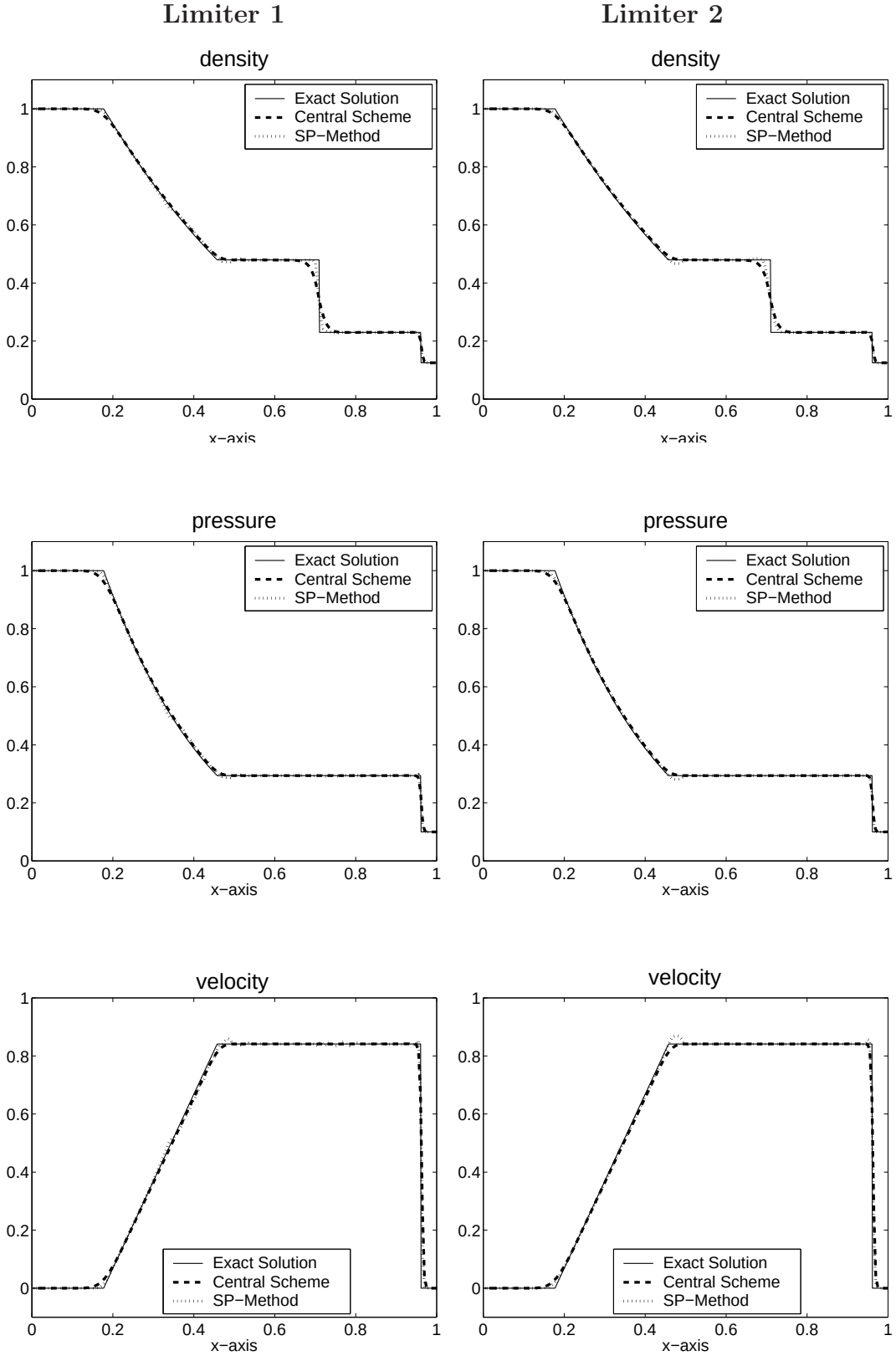

Figure 5.6: Comparison of different limiters for Example 5.3 for 200 mesh points.

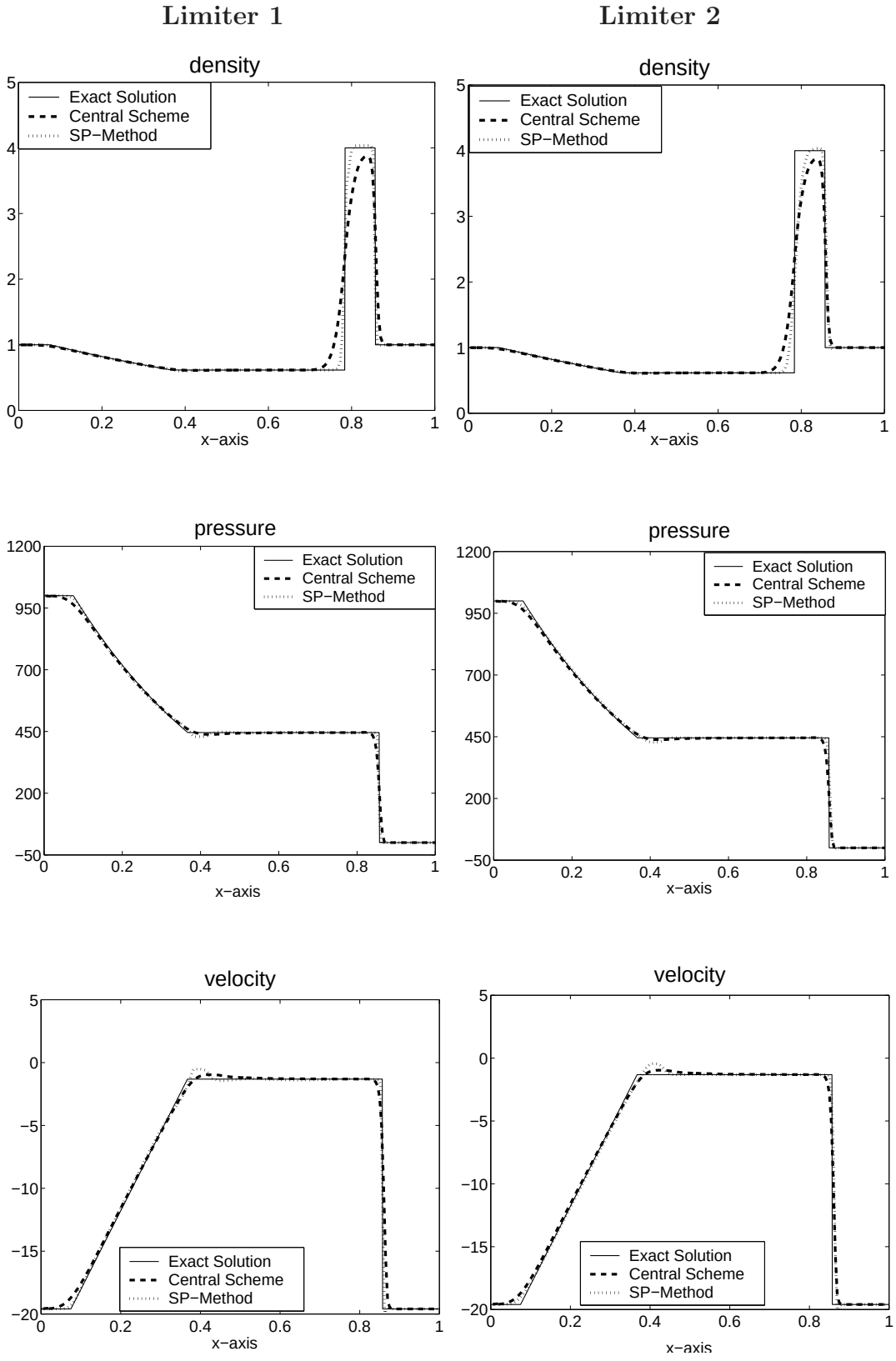

Figure 5.7: Comparison of different limiters for Example 5.4 for 200 mesh points.

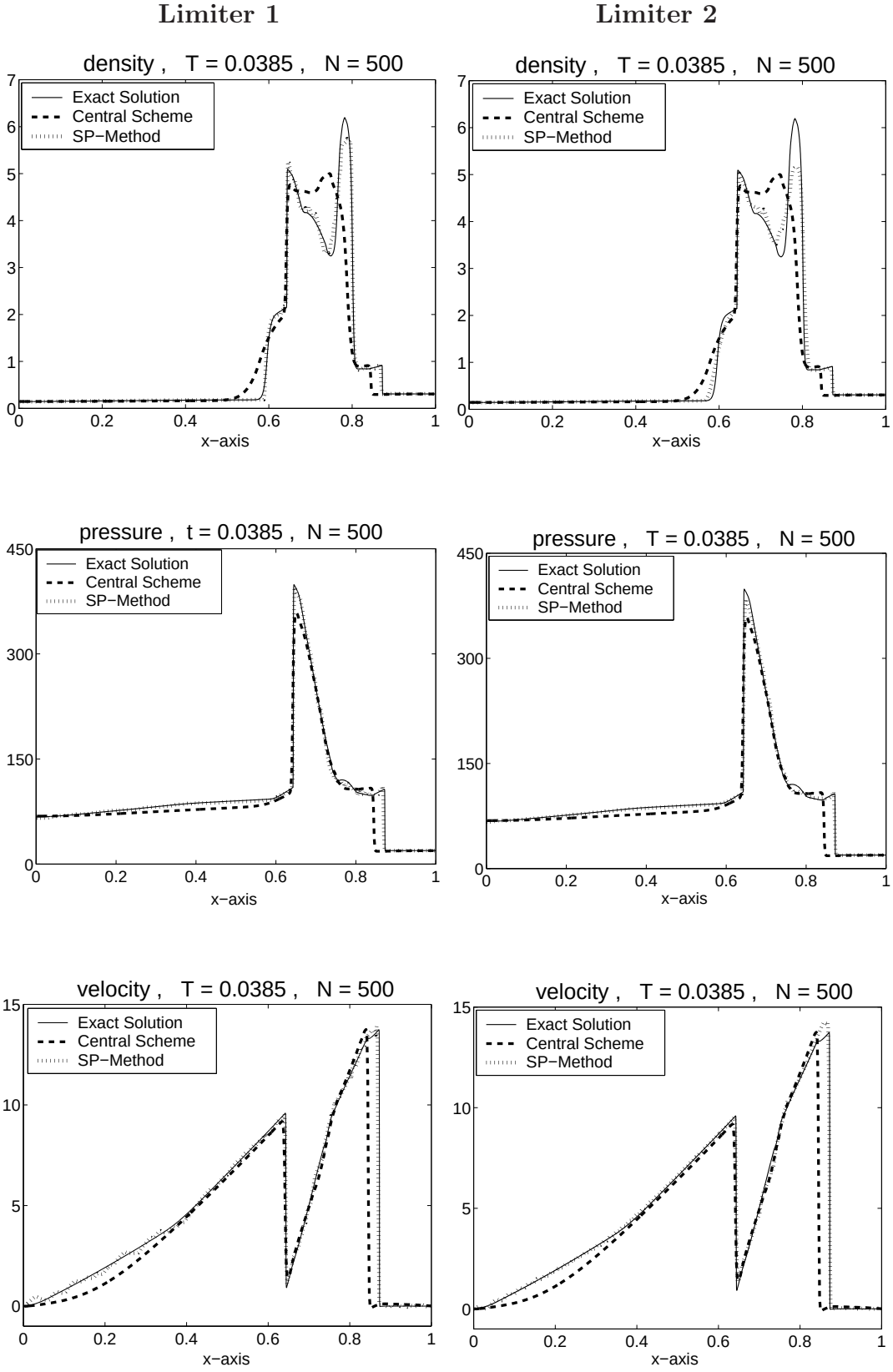

Figure 5.8: Comparison of different limiters for Example 5.5 for $N = 500$ mesh points.

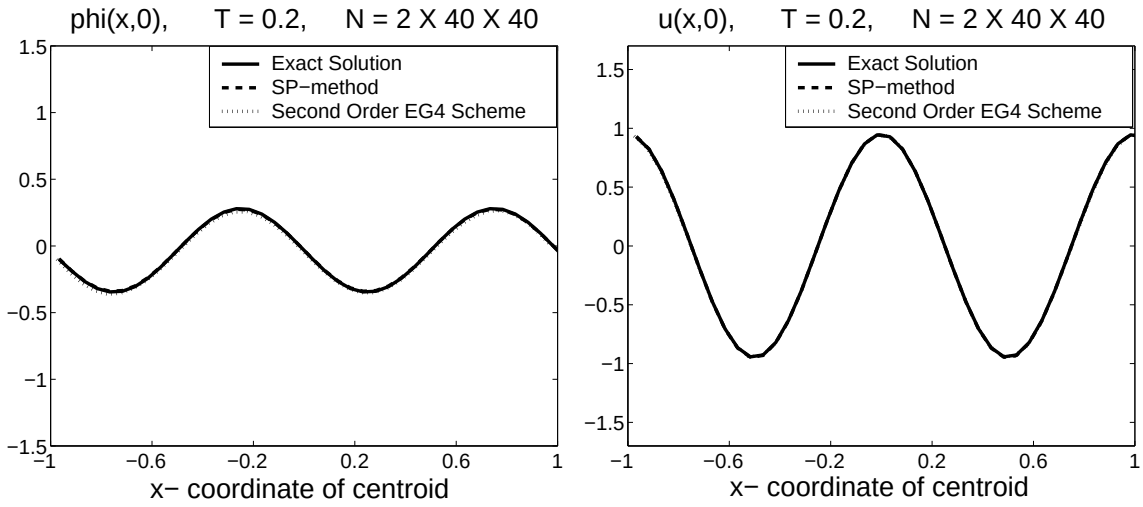

Figure 5.9: Comparison of EG scheme with SP-method for triangular mesh.

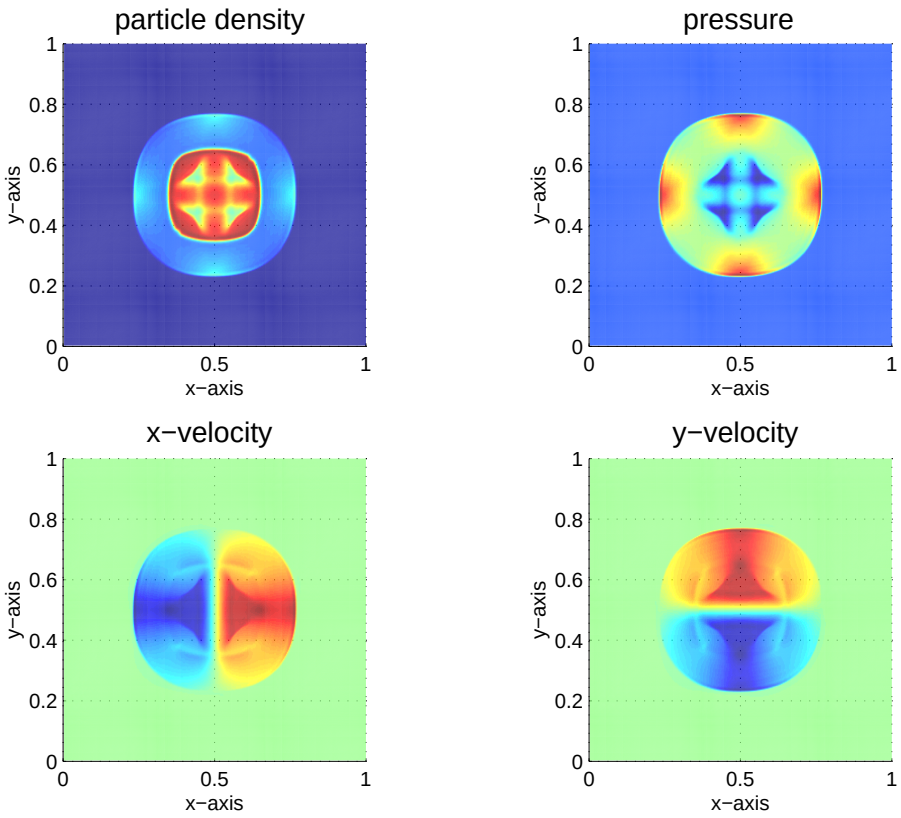

Figure 5.10: Propagation of quadratic pulse after $T = 0.1$ with 300×300 mesh points.

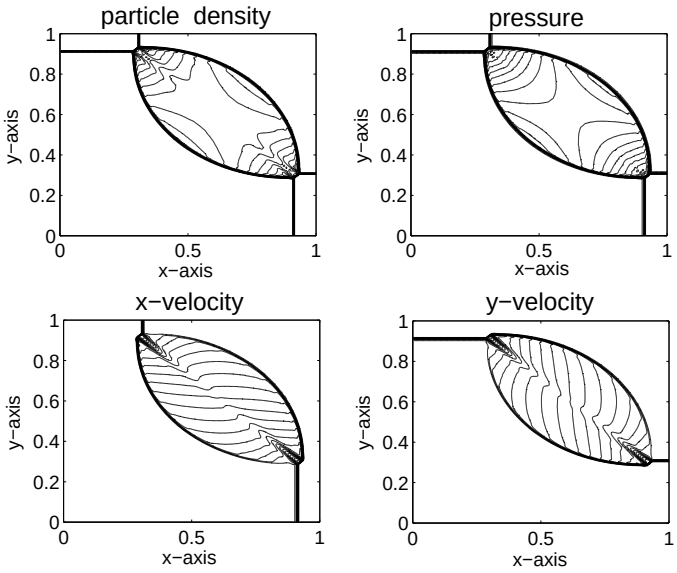

Figure 5.11: Interaction of four shocks at $T = 0.25$ with 300×300 mesh points.

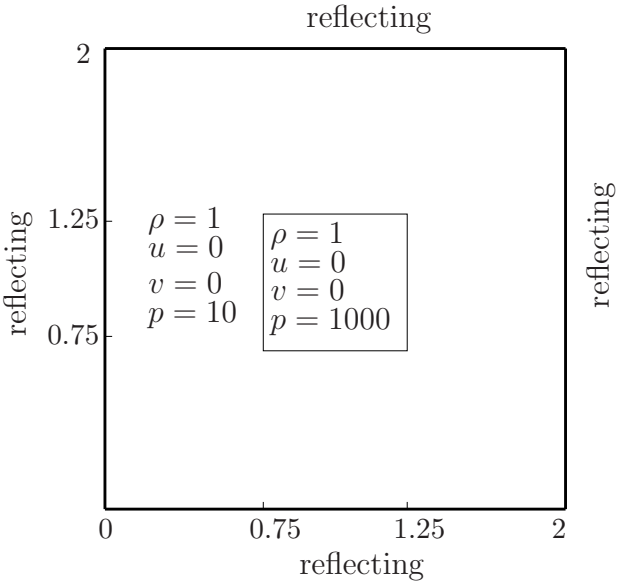

Figure 5.12: Geometrical representation of the initial data for explosion in a box

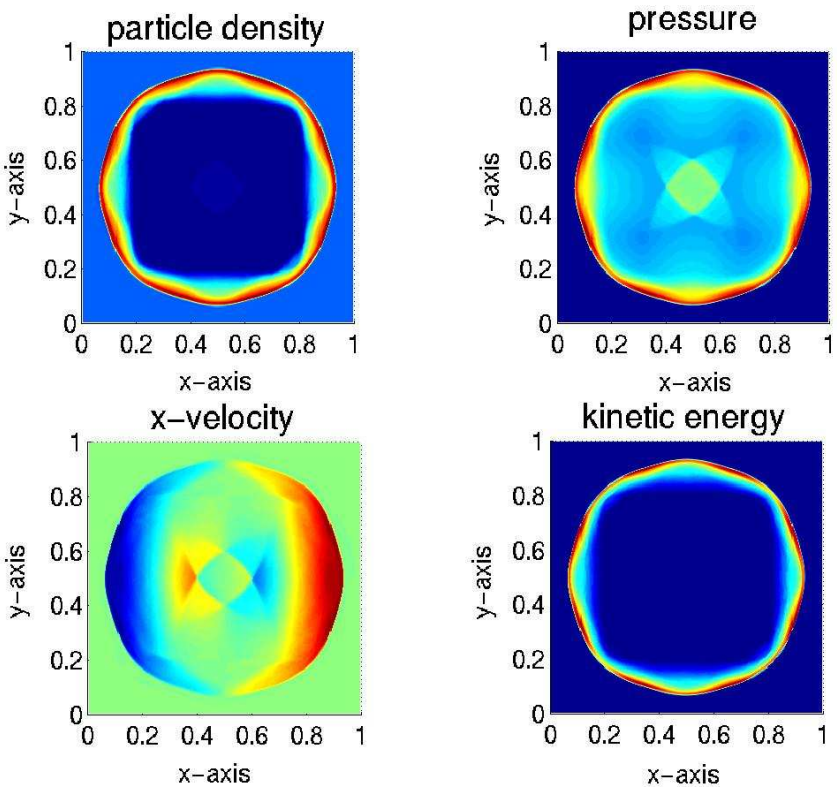

Figure 5.13: Explosion in a box $T = 0.03$ with 300×300 mesh points.

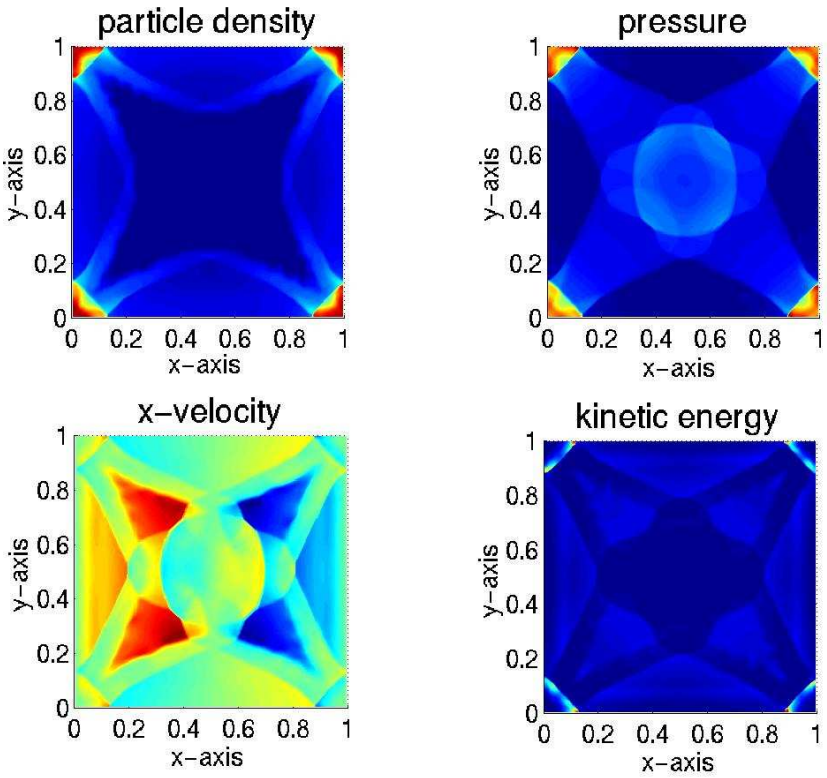

Figure 5.14: Explosion in a box $T = 0.1$ with 300×300 mesh points.

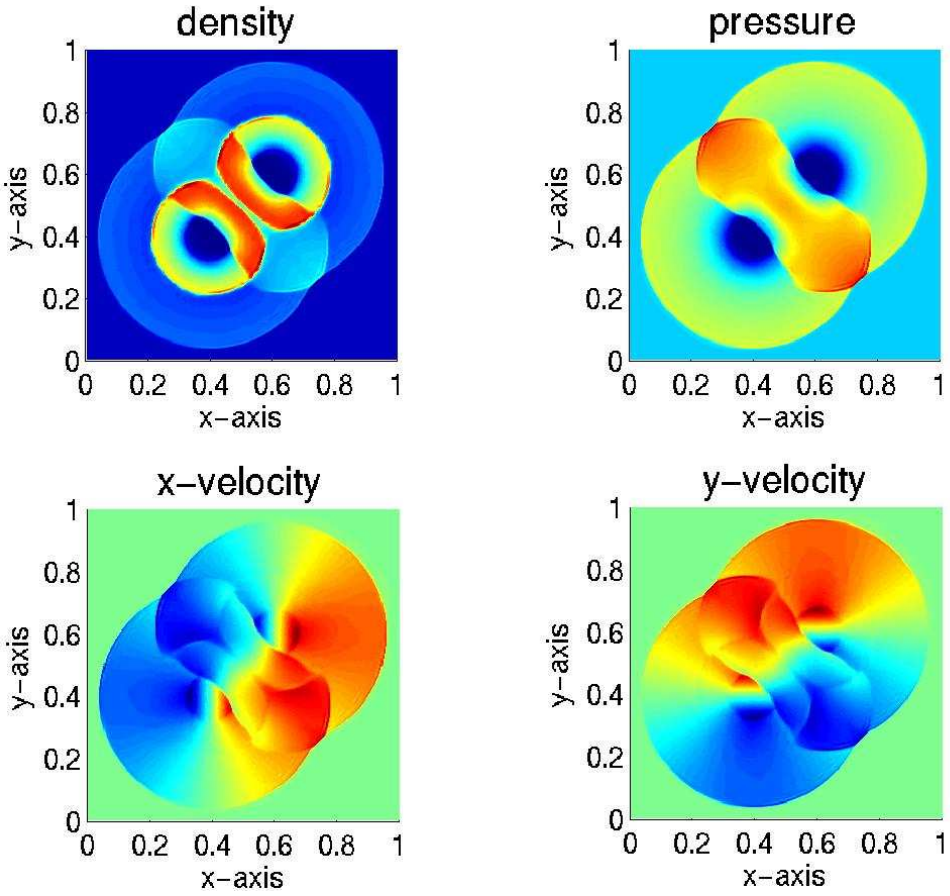

Figure 5.15: Interaction of spherical fields with 300×300 mesh points.

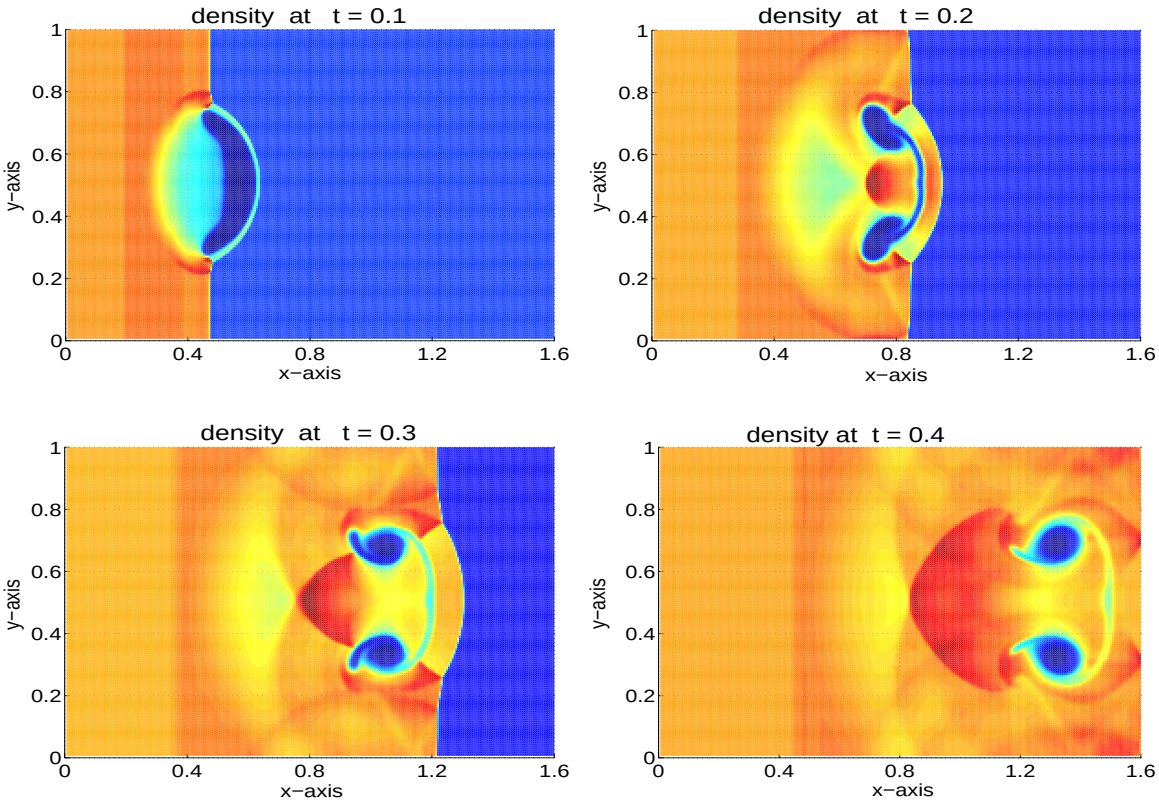

Figure 5.16: Shock bubble interaction

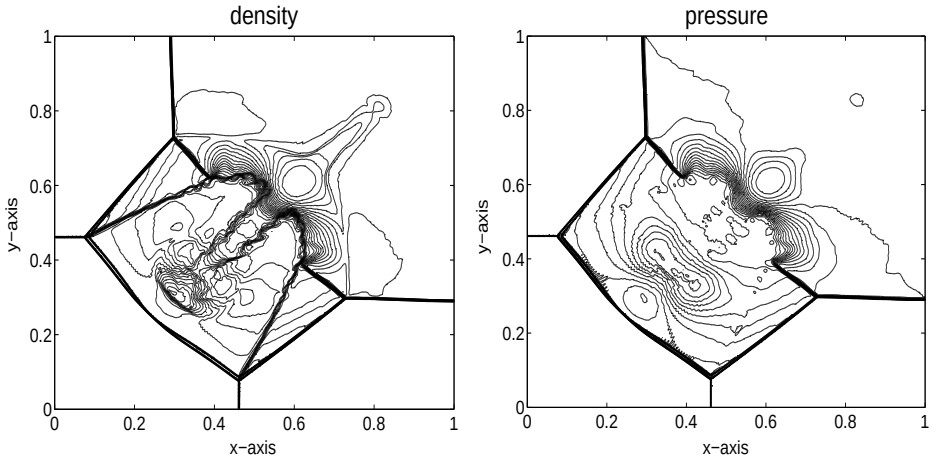

Figure 5.17: Results of Problem 5.13 at $t = 0.8$.

Chapter 6

Summary and Conclusions

In this dissertation we present two types of multidimensional finite volume schemes for hyperbolic systems based on triangular meshes. The first type of schemes are evolution Galerkin schemes and the second type is a new space-time conservative central type method which we name as the slope propagation method or SP-method.

Evolution Galerkin (EG) schemes are genuinely multidimensional schemes since these schemes take into account infinite directions of wave propagation into the calculation of fluxes. We have extended these schemes from rectangular to triangular meshes. The importance of triangular meshes arises from the fact that in practical problems irregular domains are encountered which can be filled by triangular or quadrilateral elements. Triangular mesh is a good choice for an irregular domain. We have developed first and second order EG schemes for triangular meshes and have used them to solve the linear wave equation system, the nonlinear wave equation system, the advection wave equation system, the linearized Euler equations and the nonlinear Euler equations. We have employed these schemes both for structured and unstructured triangular meshes. The accuracy and efficiency of these schemes are equally good for structured and unstructured meshes. For the first order EG schemes, the experimental order of convergence (EOC) is 1, which indicates that the schemes are truly first order with piecewise constant functions. The accuracy of second order EG schemes is considerably improved i.e. the $L2$-error is approximately 9 times smaller than the first order scheme. However the EOC of 2 has not been achieved. In the second order schemes a linear recovery has been used that is locally conservative. This method uses three immediate neighbors of a triangle to calculate slopes of the variables. Other recovery methods which involve two neighbors of a triangle or other procedures such as least squares recovery give equivalent results. The recovery stage is carried out before applying the approximate evolution operator which is used to compute edge fluxes.

The numerical implementation of EG schemes for triangular meshes is quite involved since these schemes consider infinite directions of wave propagation to compute fluxes. The fluxes are computed by using approximate evolution operator which involves integration along the circumference of a circle known as the sonic circle which is the base of the char-

acteristic cone. The center of the circle lies at the quadrature point. The edge fluxes are approximated by using Simpson's rule. In this case our quadrature points are vertices and midpoints of the edges. Trapezoidal rule can also be used for this purpose which requires the approximate evolution operator to be applied only at the vertices. At a vertex the sonic circle is intersected by the neighboring triangles sharing that vertex. To consider the infinite directions of wave propagation lying inside the circle, one has to compute the angular contribution of these neighboring triangles. Similarly at the midpoint of the edges the sonic circle is intersected by two triangles sharing that edge. In this case also the angular contribution of these triangles is necessary to know for the application of approximate evolution operator. This means that a complete knowledge of the geometry of the mesh is required for EG schemes. This data include total number of vertex neighbors, their serial numbers, three immediate neighbors of each triangle, the elements sharing an edge, the normals to each edge, the centroids of all elements, the coordinates of midpoints, vertices, element areas, element vertex numbering, edge lengths, the angles subtended by neighboring triangles at the vertices and at the edges also. We have separated all our codes into two parts. The first part computes all the geometry data and writes it to a file. The second part reads that geometry file and implements the EG schemes. This reduces the computation time considerably.

Another aspect of the EG schemes is its implementation for the nonlinear hyperbolic systems. This is even more cumbersome than for linear systems such as wave equation system. In the nonlinear case linearization of coefficients is required which implies that the higher order perturbation of the flow variables is ignored. For example in case of the Euler equations, we linearize by freezing the Jacobian matrices locally i.e. at each quadrature point. a special case of linearized Euler equations is the advection wave equation system. In this case we completely freeze the pressure, the velocities and the density. The local speed of sound is determined using these values. This means that some constant value is assigned to these variables outside the time loop and this value is same at all quadrature points. These values remain constant with time. Therefore we have a constant advection. The effect of this advection is that now the sonic circle is displaced from the quadrature point. The extent of this displacement depends upon the local velocities. The values of the local velocities determine whether the physical condition is subsonic or supersonic. In subsonic case the displacement of the sonic circle is small and it still intersects all the neighboring triangles both for vertex and edge midpoint. The angles subtended by the neighboring elements are required to be recalculated now. These angles are calculated with respect to a reference axis with its origin at the center of the displaced circle. For this purpose the points of intersections of the circle and elements edges are computed. In the sonic case the displacement of the circle center from the quadrature point is equal to the radius of the circle, hence the circle is tangent at that point. The computation of the angles is similar to the subsonic case. However there are various possible positions of the circle in this case and the program has to search for each possibility. Each possible position involves different vertex neighbors since in this case all neighboring elements of quadrature point are not intersected. The code has to sort out the right serial numbers of the triangles

sharing the circle which is not easy. The supersonic case is even more complicated than the subsonic or sonic cases. Here now the displacement of the circle center is larger than its radius therefore the circle does not cross the quadrature point. Again there are many possible positions of the circle, each involving different number of neighboring elements of the point. In order to determine the angular contribution of different elements to the circle the code has to find the elements intersecting the circle.

To take into account the nonlinear effects we have solved nonlinear Euler equations using EG schemes. In this case also some kind of initial linearization is required. Here we compute local variables inside the time loop at each vertex and edge midpoint. This is carried out by using a simple average of the piecewise constant values at the neighbouring elements of that point. In this manner the values of the local variables are different at each vertex or midpoint and in each time cycle these values are computed from the current values. Since the local velocities also vary with time, the time step which is a function of these velocities is also variable i.e. for each time cycle we have a different time step. Note that in case of the linearized Euler equations the time step was constant because the local velocities were also constant. Furthermore the displacement of the sonic circle is different at each point. This displacement depends upon the values of the local velocities at that point. Likewise the angular contributions from the neighboring elements also vary at each point and each time cycle hence the angles and the number of neighbors intersected are updated with time. The code switches from subsonic to sonic or supersonic situation depending upon the values of the current local velocities at a particular point. The computational time increases considerably because those computations which were carried out once for the linearized Euler equations, are performed in each time cycle.

We conclude that EG schemes for triangular meshes are quite difficult to implement. The first order schemes have good accuracy and EOC but the second order schemes although having several time higher accuracy than the first order schemes do not give EOC equal to 2. The accuracy of EG schemes for structured and unstructured meshes is comparable however EG schemes for unstructured meshes are more expensive computationally than the structured meshes.

Possibilities for the future work in this area include working on the higher order schemes for triangular meshes to get correct order of convergence. A bilinear recovery may be employed. For structured mesh, the integral of the xy term over a triangle, can be exactly computed, however for an unstructured mesh this term is not easy to calculate. For structured mesh all types of boundary conditions are easy to implement, however for an unstructured mesh since the serial numbers of the elements are not in the form of a matrix but randomly distributed, therefore periodic and extrapolated boundary conditions are difficult to apply. We have used only exact boundary conditions for the unstructured mesh. One possibility is that the outer boundary elements are reflected outside. This means that one can have another ghost boundary of elements which are a mirror reflections of the real boundary elements and the number of elements in the ghost and real boundaries will be

exactly equal. Therefore extrapolation boundary conditions can be applied. For periodic boundary conditions this reflection of boundary elements must be carried out twice. One can realize that this procedure is not so straight forward because the ghost elements and their vertices are to be assigned serial numbers and will be included in the actual array of elements in the mesh. Moreover in 3D, in a mesh consisting of tetrahedral elements, EG schemes are not feasible from computational point of view and from the point of view of programming complexities.

Another scheme presented in this dissertation is a space-time conservative finite volume scheme which treats space and time in a unified manner. The scheme is Jacobian free and can be applied to any hyperbolic system. We refer to this scheme as the slope propagation method (SP-method). The scheme uses the basic concepts of both, the CE/SE method originally proposed by Chang [6, 7, 8, 9, 10, 65, 73] and the central schemes of Nessyahu and Tadmor [48].

The central schemes [25, 48] use the staggered grid, well-known MUSCL-type initial reconstruction and the min-mod nonlinear limiter as basic entities to calculate slopes at each time step. These schemes are very simple, efficient and Jacobian-free.

On the other hand the CE/SE method of Chang uses the space-time control elements (CE) as conservation elements and treats the conservative flow variables and their slopes in the same way. The CE/SE method is a family of schemes, i.e., the a scheme, the $a-\epsilon$ scheme, and the $a-\alpha$ scheme. The a scheme derived for linear advection, determines the space-time geometry of the numerical mesh employed. The $a-\epsilon$ and the $a-\alpha$ schemes are extensions of the a scheme for nonlinear equations and for shock capturing. In the a scheme the number of control elements are equal to the number of unknowns. These control elements are used to derive the formulation for the conservative flow variables and their slopes. In our scheme we also use the same basic concept. This scheme has a better resolution and is as efficient as central schemes. The main disadvantage of the CE/SE method is that the calculation of the flux derivatives with respect to time is required and hence the Jacobian matrices are involved in the derivation of the scheme. The space-time derivatives of the fluxes in the scheme come from the assumption of linear variation of the fluxes in space-time. Hence the method is no longer Jacobian-free, which is an important property of the central schemes.

The main aim of our scheme is to retain the advantages of the both the CE/SE method for linear problems, the a scheme, and central schemes. We derive the scheme in a simple and straightforward way by using the basic concepts of finite volume schemes and conservation laws. The scheme uses space-time control volumes in order to compute the conservative flow variables and their slopes. In our scheme, unlike the CE/SE method, we do not assume the space-time linear variation of fluxes in each element. We assume instead the linear variation of the conservative flow variables only. The fluxes are calculated from the flow variables at the midpoint of the faces of the space-time control volumes. We

approximate the time integrals of the fluxes by using the midpoint rule. This procedure eliminates the use of Jacobian matrices in our scheme. In the one-dimensional case our staggered mesh stencil is similar to that of central schemes [48] and Chang's method [6]. In the 2D triangular mesh case we use the same staggered stencil which is used in the CE/SE method [7]. Note that Breuss [4] showed that a staggered central scheme produces less oscillations at local extrema in the data. For linear equations, our scheme reduces to the a scheme of Chang [6] and differs from the central schemes since reconstructions are not used.
The main features of the our scheme are as follows:

(i) Space and time are treated in a unified manner.

(ii) The discrete space-time control volumes are the basic conservation regions.

(iii) The derivatives of the dependent variables are also treated as independent variables.

(iv) The mesh is staggered in time.

(v) A multidimensional scheme is reconstructed on triangular meshes.

(vi) The scheme enjoys the advantages of both central schemes [48] and the CE/SE method [6].

(vii) The second order accuracy of the scheme is verified with numerical experiments.

In future we intend the following extensions in our scheme.

- Extension of the scheme to hyperbolic systems with source terms. This is very interesting area, especially the systems with the stiff source terms. For this purpose we have to rederive the method.

- Extension to unstructured meshes i.e. triangular and quadrilaterals meshes. Furthermore mixed meshes i.e. to use quadrilaterals inside the domain of computation, while triangles on the outer irregular boundaries.

- Application to MHD equations and testing the scheme for divergence-free condition. In case the scheme is not divergence-free then necessary changes can be made in the method/stencil to bring it to divergence-free form.

- Application to different hyperbolic systems, e.g, shallow water and shallow water MHD equations with source terms, multicomponent flows etc.

- Extension of the method to further higher orders.

- Stability analysis of the method.

Appendix A

Boundary Conditions

A.1 Structured Triangular Meshes and Boundary Conditions

Throughout this thesis, we will accompany the structured meshes with three types of boundary conditions, namely periodic, extrapolated and exact boundary conditions. For unstructured meshes we have used only exact boundary conditions which means that the values of exact solution is assumed at the boundary elemenets. The structured meshes that we have considered are based on rectangular meshes. We divide each rectanglular cell into two triangles as follows. Let the domain Ω be given as $\Omega = [a, b] \times [c, d]$ and let N_x, N_y be the number of subintervals along the x- and y-axis respectively. We define $\Delta x := \frac{b-a}{N_x}$ and $\Delta y := \frac{d-a}{N_y}$. Let us denote the lower and the upper triangular elements in a rectangular cell by e_1 and e_2 respectively. Further $(x_j(e), y_j(e))$, $j = 1, 2, 3$, be the coordinates of the vertices of the triangle e. The points a, b, c and d have the coordinates $(x_{\min}, y_{\min})$, $(x_{\max}, y_{\min})$, $(x_{\max}, y_{\max})$ and $(x_{\min}, y_{\max})$ respectively. Where the subscript min and max denote the minimum and maximum values of the coordinate in the rectangular region *abcda*. Then the following algorithms generate the structured triangular meshes 1 and 2 shown in Figure A.1:

Mesh1:

for $j = 0$ to $(N_y - 1)$
for $i = 0$ to $(N_x - 1)$
$e_1 = (2j)N_x + i$
$e_2 = (2j + 1)N_x + i$
if (i is even and j is even or i is odd and j is odd).

$$\begin{array}{rclcrclcrcl} x_1(e_1) & = & x_{\min} + i\Delta x & , & x_2(e_1) & = & x_{\min} + (i+1)\Delta x & , & x_3(e_1) & = & x_{\min} + i\Delta x \\ y_1(e_1) & = & y_{\min} + j\Delta y & & y_2(e_1) & = & y_{\min} + j\Delta y & & y_3(e_1) & = & y_{\min} + (j+1)\Delta y \end{array} .$$

$$\begin{array}{lcl} x_1(e_2) &=& x_{\min} + (i+1)\Delta x \\ y_1(e_2) &=& y_{\min} + (j+1)\Delta y \end{array} , \quad \begin{array}{lcl} x_2(e_2) &=& x_{\min} + i\Delta x \\ y_2(e_2) &=& y_{\min} + (j+1)\Delta y \end{array} , \quad \begin{array}{lcl} x_3(e_2) &=& x_{\min} + (i+1)\Delta x \\ y_3(e_2) &=& y_{\min} + j\Delta y \end{array} .$$

if (i is even and j is odd or i is odd and j is even).

$$\begin{array}{lcl} x_1(e_1) &=& x_{\min} + i\Delta x \\ y_1(e_1) &=& y_{\min} + j\Delta y \end{array} , \quad \begin{array}{lcl} x_2(e_1) &=& x_{\min} + (i+1)\Delta x \\ y_2(e_1) &=& y_{\min} + j\Delta y \end{array} , \quad \begin{array}{lcl} x_3(e_1) &=& x_{\min} + (i+1)\Delta x \\ y_3(e_1) &=& y_{\min} + (j+1)\Delta y \end{array} .$$

$$\begin{array}{lcl} x_1(e_2) &=& x_{\min} + (i+1)\Delta x \\ y_1(e_2) &=& y_{\min} + (j+1)\Delta y \end{array} , \quad \begin{array}{lcl} x_2(e_2) &=& x_{\min} + i\Delta x \\ y_2(e_2) &=& y_{\min} + (j+1)\Delta y \end{array} , \quad \begin{array}{lcl} x_3(e_2) &=& x_{\min} + i\Delta x \\ y_3(e_2) &=& y_{\min} + j\Delta y \end{array} .$$

end for (i)
end for (j).

Mesh2:

for $j = 0$ to $(N_y - 1)$
for $i = 0$ to $(N_x - 1)$
$e_1 = (2j)N_x + i$
$e_2 = (2j+1)N_x + i$

$$\begin{array}{lcl} x_1(e_1) &=& x_{\min} + i\Delta x \\ y_1(e_1) &=& y_{\min} + j\Delta y \end{array} , \quad \begin{array}{lcl} x_2(e_1) &=& x_{\min} + (i+1)\Delta x \\ y_2(e_1) &=& y_{\min} + j\Delta y \end{array} , \quad \begin{array}{lcl} x_3(e_1) &=& x_{\min} + (i+1)\Delta x \\ y_3(e_1) &=& y_{\min} + (j+1)\Delta y \end{array} .$$

$$\begin{array}{lcl} x_1(e_2) &=& x_{\min} + i\Delta x \\ y_1(e_2) &=& y_{\min} + j\Delta y \end{array} , \quad \begin{array}{lcl} x_2(e_2) &=& x_{\min} + (i+1)\Delta x \\ y_2(e_2) &=& y_{\min} + (j+1)\Delta y \end{array} , \quad \begin{array}{lcl} x_3(e_2) &=& x_{\min} + i\Delta x \\ y_3(e_2) &=& y_{\min} + (j+1)\Delta y \end{array} .$$

end for (i)
end for (j).

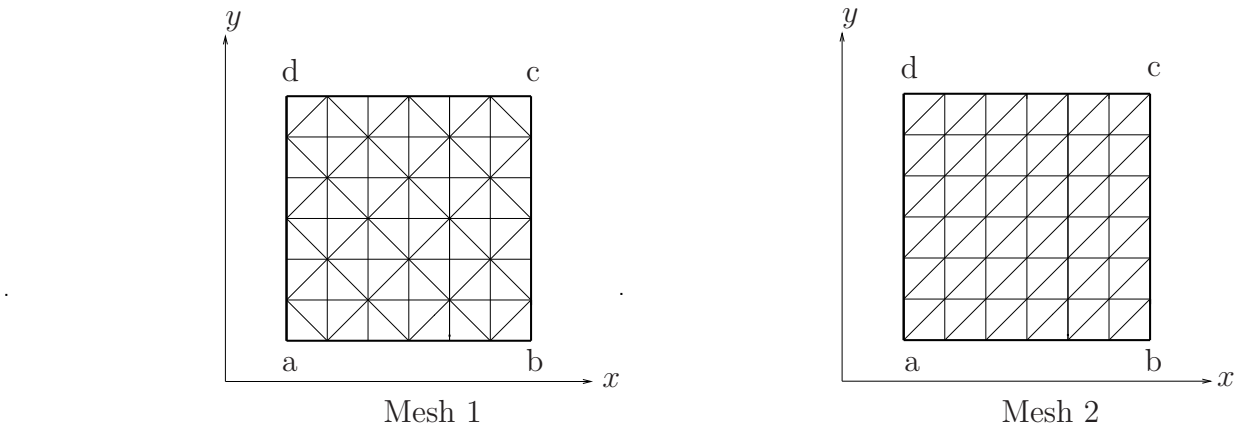

Figure A.1: Structured triangular meshes.

Now the periodic boundary for both types of meshes are:

1. $$\mathbf{U}\Big|_{e_1=(2j)N_x} = \mathbf{U}\Big|_{e1=(2j+1)N_x-2}, \quad \mathbf{U}\Big|_{e_2=(2j+1)N_x} = \mathbf{U}\Big|_{e_2=2(j+1)N_x-2}.$$
   at $x = a$

   $$\mathbf{U}\Big|_{e_1=(2j+1)N_x-1} = \mathbf{U}\Big|_{e1=(2j)N_x+1}, \quad \mathbf{U}\Big|_{e_2=2(j+1)N_x-1} = \mathbf{U}\Big|_{e_2=(2j+1)N_x+1}.$$
   at $x = b$

2. The extrapolation boundary conditions for mesh 1 read

   $$\mathbf{U}\Big|_{e_1=(2j)N_x} = \mathbf{U}\Big|_{e1=(2j)N_x+1}, \quad \mathbf{U}\Big|_{e_2=(2j+1)N_x} = \mathbf{U}\Big|_{e_2=(2j+1)N_x+1}.$$
   at $x = a$
   and

   $$\mathbf{U}\Big|_{e_1=(2j+1)N_x-1} = \mathbf{U}\Big|_{e1=(2j+1)N_x-2}, \quad \mathbf{U}\Big|_{e_2=2(j+1)N_x-1} = \mathbf{U}\Big|_{e_2=2(j+1)N_x-2}.$$

at $x = b$
Analaogous extrapolation boundary conditions can be defined for mesh 2.

A.2 Unstructured Traingular Meshes

The unstructured triangular meshes have been generated using **MATLAB PDETOOL**. Here we select a region and generate the mesh. The mesh data is exported to the matlab evironment where it is written to a file. The mesh data consists of total number of elements, vertices, the triangle vertices serial numbers and the coordinates of the vertices. We split our program into two parts (i) geometry part and (ii) evolution part. The purpose is to save computational time. The first part of the program reads the mesh data and compute the following

- Maximum and minimum points of the domain
- All the inner elements
- Boundary elements
- Inner vertices

- Boundary vertices
- Vertex neighbors
- Edge neighbors of triangles
- Angles at the vertices subtended by the neighbors
- Angles at the midpoint of an edge subtended by the edge neighbor
- Centroids
- Coordinates of the midpoints
- Areas of all elements
- Normals to the edges
- The global minimum $h_{\min}$

All this data is written to a geometry file which is read by the second part of the program. Note that the vertex numbering of the three vertices of an element is anticlockwise in the mesh which is genertaed using **PDETOOL**. Since the numbering of all elements and vertices in an unstructured mesh are random, the angles and the neighbor serial numbers must be sorted according to the reference axis (at the quadrature point). The second part of the program reads the geometry file and solves the wave equation system using EG schemes. The numerical algorithm for this part is similar to the algorithm given in Section 2.4. The advantage of this mesh generator is that most of the triangles are of good quality and very small angles are avoided.

Appendix B

Further Comparison of Errors in EG Schemes

Table B.1: First order EG4 scheme (mesh 2, periodic boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.4592615499	0.4513006456	0.4513006456	0.7862987452	
$2\times 40\times 40$	0.2945924398	0.2156972786	0.2156972786	0.4240699678	0.89
$2\times 80\times 80$	0.1661131959	0.1094441288	0.1094441288	0.2270454327	0.90
$2\times 160\times 160$	0.0884016499	0.0560174243	0.0560174243	0.1187044875	0.94
$2\times 320\times 320$	0.0457802196	0.0286008466	0.0286008466	0.0610888318	0.96

Table B.2: First order EG3 scheme (mesh 2, periodic boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.4524809587	0.4496782804	0.4496782804	0.7804871106	
$2\times 40\times 40$	0.2928463940	0.2151711044	0.2151711044	0.4223224111	0.89
$2\times 80\times 80$	0.1657032214	0.1093533897	0.1093533897	0.2266580801	0.90
$2\times 160\times 160$	0.0884016499	0.0560174243	0.05601742433	0.1187044875	0.93
$2\times 320\times 320$	0.0457802196	0.0286008466	0.0286008466	0.0610888318	0.95

Table B.3: First order EG4 scheme (mesh 1, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$	EOC
$2\times 20\times 20$	0.14460249197	0.31474643442	0.31474643442	0.46801764565	
$2\times 40\times 40$	0.09707353943	0.18205262601	0.18205262601	0.27515375580	0.77
$2\times 80\times 80$	0.05565717473	0.09767360743	0.09767360743	0.14892275942	0.89
$2\times 160\times 160$	0.02968916474	0.05057319797	0.05057319797	0.07743864157	0.94
$2\times 320\times 320$	0.01533564590	0.02573225724	0.02573225724	0.03949025399	0.97

Table B.4: First order EG3 scheme (mesh 1, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.1390210717	0.3133178845	0.3133178845	0.4643953617
$2\times 40\times 40$	0.0944759598	0.1811227560	0.1811227560	0.2730139419
$2\times 80\times 80$	0.0545441334	0.0971446917	0.0971446917	0.1478149002
$2\times 160\times 160$	0.0292004100	0.0502927505	0.0502927505	0.0768855347
$2\times 320\times 320$	0.0151111000	0.0255883308	0.0255883308	0.0392156945

Table B.5: First order EG4 scheme (mesh 1, periodic boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.4466907543	0.4532015711	0.4532015711	0.7812272128
$2\times 40\times 40$	0.2877127769	0.2177224337	0.2177224337	0.4214080662
$2\times 80\times 80$	0.1626773678	0.1104594723	0.1104594723	0.2255360638
$2\times 160\times 160$	0.0867446753	0.0565071347	0.0565071347	0.1179438478
$2\times 320\times 320$	0.044979959	0.0288417289	0.0288417289	0.0607197447

Table B.6: First order EG3 scheme (mesh 1, periodic boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.4387431179	0.4509636465	0.4509636465	0.7741007328
$2\times 40\times 40$	0.2848852861	0.2168041478	0.2168041478	0.4185304090
$2\times 80\times 80$	0.1616609391	0.1101542050	0.1101542050	0.2245042472
$2\times 160\times 160$	0.0867446753	0.0565071347	0.0565071347	0.1179438478
$2\times 640\times 640$	0.0447924645	0.0287808545	0.0287808545	0.0605230538

Table B.7: Second order EG3 scheme (mesh 1, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.0400239458	0.0585858452	0.0585858452	0.0920136879
$2\times 40\times 40$	0.0186997972	0.0199527857	0.0199527857	0.0338512885
$2\times 80\times 80$	0.0093241773	0.0070958112	0.0070958112	0.0136982246
$2\times 160\times 160$	0.0042992923	0.0030279071	0.0030279071	0.0060679780
$2\times 320\times 320$	0.0020691813	0.0014442395	0.0014442395	0.0029074330

Table B.8: Second order EG4 scheme (mesh 1, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.0330206339	0.0608528601	0.0608528601	0.0921764798
$2\times 40\times 40$	0.0172869007	0.0202920948	0.0202920948	0.0335018680
$2\times 80\times 80$	0.0091335037	0.0072391721	0.0072391721	0.0137197710
$2\times 160\times 160$	0.0043208675	0.0031088064	0.0031088064	0.0061643532
$2\times 320\times 320$	0.0021101363	0.0014863854	0.0014863854	0.0029784825

Table B.9: First order EG3 scheme with trapezoidal rule (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.1818298847	0.4014266712	0.4014266712	0.5961114423
$2\times 40\times 40$	0.1095997266	0.2375092161	0.2375092161	0.3533176412
$2\times 80\times 80$	0.0610733865	0.1297914112	0.1297914112	0.1934465803
$2\times 160\times 160$	0.0323710786	0.0679472404	0.0679472404	0.1013979372
$2\times 320\times 320$	0.0167061651	0.0347770792	0.0347770792	0.0519421451

Table B.10: Second order EG3 scheme with trapezoidal rule (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.1353951606	0.1385773797	0.1385773797	0.2381999787
$2\times 40\times 40$	0.0731344958	0.0710134659	0.0710134659	0.1242355794
$2\times 80\times 80$	0.0382643409	0.0359716066	0.0359716066	0.0636558933
$2\times 160\times 160$	0.0195192864	0.0180947541	0.0180947541	0.0321845117
$2\times 320\times 320$	0.0098657841	0.0090770790	0.0090770790	0.0161901336

Table B.11: First order EG4 scheme with trapezoidal rule (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.1805947700	0.4015049153	0.4015049153	0.5958413085
$2\times 40\times 40$	0.1093092043	0.2376065281	0.2376065281	0.3533584958
$2\times 80\times 80$	0.0610270345	0.1298358954	0.1298358954	0.1934916495
$2\times 160\times 160$	0.0323690482	0.0679634247	0.0679634247	0.1014189799
$2\times 320\times 320$	0.0167093950	0.0347828126	0.0347828126	0.0519508613

Table B.12: Second order EG4 scheme with trapezoidal rule (mesh 2, exact boundary).

N	$\|\phi(T)-\phi^n\|_{L^2}$	$\|u(T)-u^n\|_{L^2}$	$\|v(T)-v^n\|_{L^2}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^2}$
$2\times 20\times 20$	0.1327500795	0.1395089153	0.1395089153	0.2377983568
$2\times 40\times 40$	0.0719008736	0.0712070681	0.0712070681	0.1237361253
$2\times 80\times 80$	0.0378318401	0.0360063861	0.0360063861	0.0634363288
$2\times 160\times 160$	0.0193695694	0.0180972912	0.0180972912	0.0320967929
$2\times 320\times 320$	0.0098102667	0.0090744560	0.0090744560	0.0161534157

Table B.13: Second order EG4 scheme (mesh 1, exact boundary, CFL = 0.1).

N	$\|\phi(T)-\phi^n\|_{L^1}$	$\|u(T)-u^n\|_{L^1}$	$\|v(T)-v^n\|_{L^1}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^1}$	EOC
$2\times 10\times 10$	0.0764977252	0.1268415325	0.1268415325	0.3301807902	
$2\times 20\times 20$	0.0307572186	0.0323126346	0.0323126346	0.0953824879	1.79
$2\times 40\times 40$	0.0087873416	0.0067541715	0.0067541715	0.0222956847	2.1
$2\times 80\times 80$	0.0031224392	0.0019579375	0.0019579375	0.0070383143	1.67
$2\times 160\times 160$	0.0013980801	0.0008110767	0.0008110767	0.0030202336	1.22

Table B.14: Second order EG4 scheme (mesh 2, exact boundary, CFL = 0.2).

N	$\|\phi(T)-\phi^n\|_{L^1}$	$\|u(T)-u^n\|_{L^1}$	$\|v(T)-v^n\|_{L^1}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^1}$	EOC
$2\times 10\times 10$	0.0725908320	0.1259144356	0.1259144356	0.3244197033	
$2\times 20\times 20$	0.0297897995	0.0351575165	0.0351575165	0.10010483	1.70
$2\times 40\times 40$	0.0140660509	0.0100164950	0.0100164950	0.03409904	1.55
$2\times 80\times 80$	0.0057622798	0.0034913039	0.0034913039	0.01274488	1.42
$2\times 160\times 160$	0.0025215824	0.0014810930	0.0014810930	0.00548376	1.22

Table B.15: Second order EG4 scheme (mesh 2, exact boundary, CFL = 0.4).

N	$\|\phi(T)-\phi^n\|_{L^1}$	$\|u(T)-u^n\|_{L^1}$	$\|v(T)-v^n\|_{L^1}$	$\|\mathbf{U}(T)-\mathbf{U}^n\|_{L^1}$	EOC
$2\times 10\times 10$	0.1478236868	0.1483282518	0.1483282518	0.4444801905	
$2\times 20\times 20$	0.0447720624	0.0427115362	0.0427115362	0.1301951348	1.77
$2\times 40\times 40$	0.0228091374	0.0153635416	0.0153635416	0.0535362207	1.28
$2\times 80\times 80$	0.0111776148	0.0065768356	0.0065768356	0.0243312861	1.14
$2\times 160\times 160$	0.0049980034	0.0029136062	0.0029136062	0.0108252160	1.17

Bibliography

[1] R. Abgrall, "On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation", *J. of Comput. Phys.* **114**, (1994), pp. 45-58.

[2] P. Arminjon, M.-C. Viallon and A. Madarne, "A finite volume extension of the Lax-Friedrichs and Nessyahu-Tadmor schemes for conservation laws on unstructured grids", *Int. J. for Comp. Fluid Dynamics.* **9**, (1997), pp. 1-22.

[3] M. Ben-Artzi, J. Falcovitz " A second-order Godunov-type scheme for compressible fluid dynamics", *J. of Comput. Phys.* **55**, (1984), pp. 1-32.

[4] M. Breuss, "An analysis of the influence of data extrema on some first and second order central approximations of hyperbolic conservation laws", to appear in *Mathematical Modeling and Numerical Analysis* (2005).

[5] S. Canic, B. L. Keyfitz and E. H. Kim, "Mixed hyperbolic-elliptic systems in self-similar flows", *Boletim da Sociedade Brasileira de Matematica* **32**, (2002), pp. 1-23.

[6] S.C. Chang, "The method of space time conservation element and solution element -A new approach for solving the Navier Stokes and Euler equations", *J. of Comput. Phys.* **119**, (1995), pp. 295.

[7] S.C. Chang and W.M. To, "A new numerical framework for solving conservation laws-the method of space-time conservation element and solution element", NASA TM 104495, August 1991.

[8] S.C. Chang, X.Y. Wang and C.Y. Chow, "New developments in the method of space-time conservation element and solution element: Applications to two-dimensional time-marching problems", NASA TM 106758, December 1994.

[9] S.C. Chang, S.T. Yu, A. Himansu, X.Y. Wang, C.Y. Chow and C.Y. Loh, "The method of space-time conservation element and solution element: A new paradigm for numerical solution of conservation laws", *Computational Fluid Dynamics Review 1*, **206**, (1998), Edited by M. Hafez and K. Oshima, World Scientific Publisher.

[10] S.C. Chang, X.Y. Wang and C.Y. Chow, "The space-time conservation element and solution element method: A new high resolution and genuinely multidimensional paradigm for solving conservation laws", *J. of Comput. phys.* **156**, (1999), pp. 89.

[11] P.N. Childs, "The characteristic Galerkin method for hyperbolic conservation laws", Ph.D thesis, Oxford university, 1988.

[12] R. Courant and K.V. Friedrichs, "Supersonic flow and shock waves", Wiley-Intersciences, New York, (1948).

[13] R. Courant and D. Hilbert, "Methods of mathematical physics", Interscience publishers, 1962.

[14] V. Dolejsi and J. Felcman, "Investigation of accuracy for higher order finite volume schemes", *Journal of Engineering Mechanics* **5**, (1998), pp. 327-337.

[15] W. Dreyer, M. Kunik, K. Sabelfeld, N. Simonov and K. Wilmański, "Iterative procedure for multidimensional Euler equations", *Monte Carlo Methods and Appl.* **4**, (1998), pp. 253-271.

[16] W. Dreyer and S. Qamar, "Kinetic flux-vector splitting schemes for the hyperbolic heat conduction", *J. of Comput. Phys.* **198**, (2004), pp. 403-423.

[17] L. J. Durlofsky, B. Enquist, and S. Osher, "Triangle based adaptive stencils for the solution of hyperbolic conservation laws", *J. of Comput. Phys.* **98**, (1992), pp. 64-73.

[18] K.O. Friedrichs and P.D. Lax, "Systems of conservation equations with a convex extension", *Proc. Acad. Sci. USA* **68**, (1971), pp. 16-86.

[19] E. Godlewski and P.-A. Raviart, "Numerical approximation of hyperbolic systems of conservation laws", Springer-Verlag, New York, (1996).

[20] E. Godlewski, P.A. Raviart, "Hyperbolic systems of conservation laws", Mathematical Appl., Ellipses, Paris, (1991).

[21] S.K. Godunov, "A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics", *Mat. Sb.* **47**, (1959), pp. 357-393.

[22] C. Hirsch, "Numerical computations of internal and external flows. volume 2: Computational methods for inviscid flows", John Wiley & Sons, New York (1992).

[23] C. Hu, "Weighted essentially non-oscillatory schemes on triangular meshes", *J. of Comput. Phys.* **150**, (1999), pp. 97-127.

[24] G. Jiang and C.-W. Shu, "Efficient implementation of weighted ENO schemes", *J. of comput. Phys.* **126**, (1996), pp. 202-228.

[25] G.-S. Jiang , E. Tadmor, "Non-oscillatory central schemes for multidimensional hyperbolic conservation laws", *SIAM J. Sci. Comput.* **19**, (1998), pp. 1892-1917.

[26] R. Khanfr, G. Chanteur and J.P. Croisille, "Resolution of the system of equations of ideal magnetohydordynamics by a finite volume kinetic-type method", *Computer physics Communications* **98**, (1996), pp. 301-316.

[27] T. Kröger, S. Noelle, S. Zimmermann, "On the connection between some Riemann-Solver free approaches to the approximation of multidimensional systems of hyperbolic conservation laws", *ETH Zentrum, Seminar fur Angewandte Mathematik, CH-8092 Zuerich.*

[28] D. Kroñer, "Numerical schemes for conservation laws", Wiley & Tuebner, (1997).

[29] A. Kurganov and P.D. Levy, "A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations", *SIAM. J. Sci. Comput.,* **22**, (2000), pp. 1461-1488.

[30] J. O. Langseth and R. J. LeVeque, "A wave propagation method for three-dimensional hyperbolic conservation laws", *J. Comput. Phys.* **165**, (2000), pp. 126-166.

[31] B. van Leer, "Flux vector splitting for the Euler equations", ICASE Report No. 82-30, 1982.

[32] X.D. Liu, S. Osher and T. Chan, "Weighted essentially non-oscillatory schemes", *J. of comput. Phy.* **115**, (1994), pp. 200-212.

[33] R. J. LeVeque. "Wave propagation algorithms for multidimensional hyperbolic systems", *J. of Comput. Phys.* **131**, (1997), pp. 327-353.

[34] R.J. LeVeque, "Numerical methods for conservation laws", Second Edition, Birkhäuser, (1992).

[35] K.-A. Lie, S. Noelle, "An improved quadrature rule for the flux computation in staggered central difference schemes in multi-dimensions", *J. Sci. Comput.* **18**, (2003) , pp. 69-81.

[36] J. Li, T. Zhang and S. Yang, "The two-dimensional Riemann problems in gas dynamics", Harlow: Longman, (1998).

[37] M. Lukáčová-Medvidova, J. Saibertova, G. Warnecke and Y. Zahaykah, "On evolution Galerkin methods for the Maxwell and the linearized Euler equations", *Appl. Math.* **49**, (2004), pp. 415-439.

[38] M. Lukáčová-Medvidova, G. Warnecke and Y. Zahaykah, "Third order finite volume evolution Galerkin (FVEG) methods for two-dimensional wave equation system", *J. Numer. Math* **11**, (2003), pp. 235-251.

[39] M. Lukáčová-Medvidova, G.Warnecke and Y.Zahaykah, "On the boundary conditions for EG-methods applied to the two-dimensional wave equation system", *ZAMM* **84**, (2004), pp. 237-251.

[40] M. Lukáčová-Medvidova, G. Warnecke and Y.Zahaykah, "On the stability of the evolution Galerkin schemes applied to a two-dimensional wave equation System", submitted to *Numer. Math.*, (2004).

[41] M. Lukáčová, K.W. Morton and G. Warnecke, "Evolution Galerkin methods for hyperbolic systems in two space dimensions", *Math. Comput.* **69**, (2000), pp. 1355-1384.

[42] M. Lukáčová, K.W. Morton and G. Warnecke, "Evolution Galerkin methods for multidimensional hyperbolic systems", *Proceedings of ECCOMAS 2000, Barcelona, 1-14*, (2000).

[43] M. Lukáčová ,J. Saibartová and G. Warnecke, "Finite volume evolution Galerkin methods for nonlinear hyperbolic systems", *J. of Comput. Phys.* **183**, (2002), pp. 533-562.

[44] M. Lukáčová, Warnecke and Y. Zahaykah, "Third order finite volume evolution Galerkin (FVEG) methods for two-dimensional wave equation system", *Proceedings of ENNUMAMATH'99, World Scientific Publishing Company, Singapore*, (1999).

[45] M. Lukáčová, W. Morton and G. Warnecke, "Finite volume evolution Galerkin methods for Euler equations of gas dynamic", *Int. J. Num. Methods in Fluids* **40**, (2002), pp. 425-434.

[46] M. Lukáčová and Z. Vlk, "Well balanced finite volume evolution Galerkin methods for the shallow water equations with source terms", *Int. J. Numer. Meth. Fluids*, accepted (2004).

[47] K.W. Morton, "On the analysis of finite volume methods for evolutionary problems", *SIAM J. Numer. Anal.* **35**, (1998), pp. 2195-2222.

[48] H. Nessyahu, E. Tadmor, "Non-oscillatory central differencing fo hyperbolic conservation Laws",*SIAM J. Comput. Phys.* **87**, (1990), pp. 408-448.

[49] S. Noelle. "The MOT-ICE: a new high resolution wave-propagation algorithm for multidimensional systems of conservative laws based on Fey's method of transport", *J. of Comput. Phys.* **164**, (2000), pp. 283-334.

[50] C.F. Ollivier-Gooch, "Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-square reconstruction", *J. of Comput. Phys.* **133**, (1997), pp. 6-17.

[51] S. Ostkamp, "Multidimensional characteristic Galerkin schemes and evolution operators for hyperbolic systems", *Dissertation, Hannover*, 1995.

[52] S. Ostkamp. "Multidimensional characteristic Galerkin schemes and evolution operators for hyperbolic systems", *Math. Meth. Appl. Sci* **20**, (1997), pp. 1111-1125.

[53] S. Qamar, "Kinetic schemes for the relativistic hydrodynamics", *Dissertation, Magdeburg*, 2004.

[54] S. Qamar and G. Warnecke, "A high order kinetic flux-spitting method for the special relativistic magnetohydrodynamics", to appear in *J. of Comput. Phys.*, (2004).

[55] S. Qamar and G. Warnecke, "Simulation of multicomponent flows using high order central schemes", *ANUMP* **50**, (2004), pp. 183-201.

[56] J. Quirk. "A contribution to the great Riemann solver debate", *Int. J. Numer. Meth. in Fluid Dyn.* **18**, (1994), pp. 555-574.

[57] P. Roe, "Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics", *J. of Comput. Phys.* **63**, (1986), pp. 458-476.

[58] C. W. Schulz-Rinne, J. P. Collins and H. M. Glaz, "Numerical solution of the Riemann problem for two-dimensional gas dynamics", *SIAM J. Sci. Comput.* **14**, (1993), pp. 1394-1414.

[59] J. Smoller, "Shock waves and reaction-diffusion equations", Springer-Verlag, New York-Heidelburg-Berlin, (1983).

[60] G. A. Sod, "A survey of several finite difference methods for systems of nonlinear hyperbolic conservation Laws", *J. of Comput. Phys.* **27**, (1978), pp. 1-31.

[61] T. Sonar, "Mehrdimensionale ENO-verfahren", Teubner, Stuttgart, (1997).

[62] T. Sonar, "On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations:polynomial recovery, accuracy and stencil selection", *Comput. Methods Appl. Mech. Engrg.* **140**, (1997), pp. 157-181.

[63] T. Thomas, "Numerical partial differential equations, finite difference methods", Springer-Verlag, (1999).

[64] E.F. Toro, "Riemann solvers and numerical method for fluid dynamics", Second Edition, Springer-Verlag, New York, (1995).

[65] X.Y. Wang, C.Y. Chow and S.C. Chang, "Application of the space-time conservation element and solution element method to shock-tube problem", NASA TM 106806, December 1994.

[66] X.Y. Wang, C.Y. Chow and S.C. Chang, "Application of the space-time conservation element and solution element method to two-dimensional advection-diffusion problems", NASA TM 106946, June 1995.

[67] G. Warnecke, "Analytiche methoden in der theorie der erhaltungsgleichungen", B. G. Teubner, Stuttgart, (1999).

[68] G. B. Whitham, "Linear and nonlinear waves", John Wiley & Sons, New York-Chichester-Brisbane-Toronto-Singapore, (1974).

[69] P. Woodward and P. Colella, "The numerical simulation of two-dimensional fluid flow with strong shocks", *J. of Comput. Phys.* **54**, (1988), pp. 115-173.

[70] K. Xu, "A slope-update scheme for compressible flow simulation", *J. Comput. Phys.* **178**, (2002), pp. 252-259.

[71] Y. Zahaykah, "Evolution Galerkin schemes and discrete boundary conditions for multidimensional first order systems", *Dissertation, Magdeburg*, 2002.

[72] P. Zhang, J. Li and T. Zhang, "On the two-dimensional Riemann problems for pressure-gradient equations of the Euler system", *Discrete Contin. Dynamical Systems*, **4**, (1998), pp. 609-634.

[73] Z-C Zhang, S.T. Yu and S.C. Chang, "A space-time conservation element and solution element method for solving the two-dimensional unsteady Euler equations using quadrilateral and hexahedral meshes", *J. Comput. Phys.*, **175**, (2002) pp. 168-199.