**ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

**DEPARTMENT OF MATHEMATICS**

# FINITE VOLUME METHOD FOR A LINEAR ADVECTION EQUATION IN TWO DIMENSSIONAL SPACES

**A thesis submitted in partial fulfillment of the requirements for the degree of master's of science in Mathematics (Numerical analysis)**

**BY**

**SAMUEL DEBEBE**

**September,2017**

**Adama**

ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF GRADUATE STUDIES

DEPARTEMENT OF MATHEMATICS

# FINITE VOLUME METHOD FOR A LINEAR ADVECTION EQUATION IN TWO DIMENSSIONAL SPACES

A thesis submitted in partial fulfillment of the requirements for the degree of master's of science in Mathematics (Numerical analysis)

BY

Samuel Debebe

ADVISOR: LEMMI GUTA (PH.D)

CO - ADVISOR: YADETA CHIMDESSA

September, 2017

Adama

# ACKNOWLEDGEMENT

First of all, I would like to express my deepest gratitude to almighty God for giving me patience to complete this work and It is a pleasure to acknowledge my advisor Lemmi Guta (Dr.) and my co- advisor Yadeta Chimdessa for their invaluable support and encouragement throughout this thesis work. No matter what the question I have, they always had the time to hear me. They are indeed a wonderful supervisors. Finally I acknowledge Adama University for providing financial budget needed to do this thesis work.

***ABSTRACT***

*This thesis provides a numerical solutions to a scalar advection equation which model to measure the rate of change of the concentration (density) of substances (pollutants) moving through the flow of the fluid in two dimensional space. To solve this equation using finite volume method. Constructing unstructured mesh for irregular shaped cells that needs to calculate the volume(or the area in 2D) for each of different size cells and the time steps to pass each cells make difficult to implement this scheme. Thus to overcome this difficulty the computational domain is embedded in a uniform Cartesian plane which contain most of the computational region in it. And small cells near the border are treated by the same scheme as the interior part. We apply the FV scheme which is modeled by using a wave-propagation approach in which the flux at each cell interface is built up on the basis of information propagating in the upwind direction of this interface from neighboring cells which is unconditionally stable for Courant numbers up to 1. By employing this approach we have shown the stability and convergence of the upwind finite volume scheme to solve linear advection equation through test examples.*

# Notations

PDE= partial differential equation

FVM= finite volume method

FVS= finite volume scheme

2-D = two dimensional spaces

$\|\cdot\|_p$ = Norm of a function or a vector& p denotes the order of the norm

The time level: $t_{n-1}$ = past time; $t_n$ = present time; $t_{n+1}$ = next time.

The letter k is used as a constant time step = $\Delta t = t_{n+1} - t_n$.

$t = n\,\Delta t = n\,k$ where n = 0,1, 2, . . . ., N.

$T = N\,\Delta t$ ,where N is the number of time steps.

Superscript n: $q^n$ = q at time level n.

$Q_{i,j}^n$ = the average cell center value of the q at time level.

NI&NJ is number of cells in x &y direction on the physical domain [0, 1] x [0,1].

Induces i & j are used to store the coordinates of vertices and center of the cells in x-y plane.

Subscripts: $i - 1/2$, $j$ and $i + 1/2$ , $j$ , attached with x, y, F &$G$, are used to denote the sides and the fluxes of the cell (i , j) at the left and the right interfaces of a cell respectively.

$i , j - 1/2$ are for bottom interfaces and $i , j + 1/2$ are for top interfaces of the cell

$\Delta x = h$ & $\Delta y = h$ are horizontal and vertical spatial length scales respectively.

h: denotes constant spatial length scale of a uniform Cartesian plane grid ( mesh).
(resp.) = respectively.

A stencil means a range of consecutive points in the computational domain.

# Table Content

# CHPTER ONE

# 1. INTRODUCTION

## 1.1. Background of the study

For most partial differential equations, we are not able to find their exact solutions, and sometimes we do not even know whether a unique solution exists. In addition to this if the solutions to the PDEs exist, they are functions, as opposed to standard algebraic equations whose solutions are numbers. However in sciences it is common to describe phenomena in terms of specific quantities which may take on numerical values from time to time. For example, we may describe the distribution of the pollutant in space at any point in terms of the mass, pressure, density. For these reasons, in most cases the only way to solve partial differential equations arising in concrete engineering and scientific problems is to approximate their solutions numerically. And hence numerical methods for PDEs constitute an indivisible part of modern engineering and science.

There are three classical choices for the numerical solution of PDEs; the finite difference method (FDM), the finite element method (FEM) and the finite volume method (FVM). The FDM is the oldest and is based upon the application of a local Taylor expansion to approximate the differential equations. The FDM uses a topologically square network of lines to construct the discretization of the PDE. This is a potential weakness of the method when handling complex geometries in multiple dimensions. This issue motivated the use of an integral form of the PDEs and subsequently the development of the finite volume techniques **[11].**

The starting point for a finite-volume method is a decomposition of the problem domain into a finite number of sub domains called control volumes (CVs), and related nodes where the unknown variables are to be computed. The union of all CVs should cover the whole problem domain. In general, the CVs also may overlap, but since these results in

unnecessary complications the non-overlapping case is mostly used. Since finally each CV gives one equation for computing the nodal values, their final number (i.e., after the incorporation of boundary conditions) should be equal to the number of CVs. Usually, the CVs and the nodes are defined on the basis of a numerical grid[13].

The continuous problem that result from the modeling – usually systems of differential or integral equations derived in the framework of continuum mechanics – must then be suitably approximated by a discrete problem, i.e. ,the unknown quantities to be computed have to be represented by a finite number of values. This process, which is called discretization, mainly involves two tasks: the discretization of the problem domain & the equations. The finite-volume method (FVM) is among one of the classical approaches applied for these process.

Hyperbolic equations can be decomposed in to two scalar equations. This property is a basis for the implementation of FVMs on hyperbolic equations used to model a wide variety of phenomena that involve wave motion or the advective transport of substances (advective transport refers to a substance being carried along with fluid motion). These problems are generally time-dependent so that their solutions are dependent on time as well as one or more spatial variables**[6].**

In this thesis, we consider the advection of a scalar concentration or density function q (x, y, t) whose evolution is governed by the PDE in a constant velocity field $\vec{u}$ = (u(x, y, t), v(x, y, t)) of two dimensional space in which the flow is assumed to be incompressible & wish to determine the approximate values of the concentration of the advected substance at spatial points in 2-D spaces using FVMs.

The goal of this thesis is to present numerical solutions for the advection equation by using FVM methods, starting with the most basic upwind method and adding in a sequence of simple correction terms to achieve better accuracy and stability properties.

### 1.2. Statement of the problem

As it is stated in **[6]** most advective algorithms, which have been developed by different researchers in the past, used to find the numerical solution of advection equation in multidimensional space using FVM attempt to determine a grid which conforms to boundaries of computational region. The grid is formed by dividing the computational region in to the finite number of regular polygons and irregular polygons (near the boundary of the region) in order to approximate the equation over these cells and form FV scheme using the fact that the difference of the flux of the advective quantities that propagate into and out of a cell and its neighboring cells through the sides of a cell in a fixed time level is conserved.

This mostly known approach of FVM used to solve PDEs does not consider the natural geometrical interpretation of the cells because small irregular cells which are cut off by the boundary are updated by the same formulas as regular cells away from the boundary. This leads to put restriction for small cells which in turn maximizes the computational error and time. Moreover arbitrary geometry of the physical domain of the problem can't be handled on this grid.

And hence, this thesis was done to demonstrate framework that overcome the above problem arise in solving the advection problem in irregular geometry using FVM that uses a uniform Cartesian plane mesh as the ordinary finite difference methods oftenly used by researchers, in solving the advection problem in irregular geometry and to implement the scheme using MAT-LAB software.

In this study the researcher addressed the following questions:

- What changes brought to the stability and the convergence of the scheme when the Cartesian meshes are employed to solve advection equation?
- What will happen if the number of cells increase uniform Cartesian plane mesh (as step time and step size are decrease) to the schemes?

### 1.3 The purpose of the study

Many physical phenomena of nature are described by PDEs in which two or more dependent and independent variables are assigned to describe some behavior of the system of the phenomena together with their partial derivatives. To understand the behavior of the system the derivatives of dependent variables must be solved. Analytical methods are mostly applied techniques that are used to get a solution for PDEs but not all PDEs can be solved through these techniques. Due to this reason numerical methods have been developed to solve such PDEs .And hence the purpose of this study is to provide alternative numerical solution of 2-D linear advection equation using one of the numerical methods called FVM.

### 1.4. Objective of the study

#### 1.4.1 The general objective of the study

The main aim of this thesis is to find the numerical solution of linear advection equations in two dimensional spaces using FVM.

#### 1.4.2. The specific objectives of the study

The Specific objectives of this study are to:

- Show how the domain of solution (computational region)& the partial differential equation are descritsized spatially to obtain an approximate numerical solution to the linear advection equation in two dimensional spaces.
- Examine consistency and convergence of the approximation of the linear advection equation in 2-D spaces.
- Examine the Convergence and Stability of the method to the solution of a linear advection equation 2-D spaces.
- Show how the finite volume scheme, that has been constructed to find numerical solution for linear advection equation in 2-D space, is implemented on MAT-LAB

### 1.5. Significance of the study

The main purpose of this thesis is to provide numerical solution to linear advection equations in two dimensional spaces. Any interested researcher may use this document as reference for further on two and three dimensional hyperbolic equations. This study may serve as a motivation to the beginners to realize the application of numerical methods in computational physics and initiate them for further research in applied mathematics.

### 1.6. Scope of the study

The research study is mainly delimited on a single first order linear advection equation that describe the evolution of the advection of a scalar concentration or density function in constant velocity field of incompressible flow in two dimensional spaces. By doing so the main focus is on comparison of the numerical and analytical solution of a linear advection equation in two dimensional spaces. A uniform Cartesian grid is applied to construct control volumes (cells) on the rectangular computational region.

### 1.7. Limitation of the study

The limitation encountered in this study was lack of adequate information due to shortage of literature materials. It was also too difficult for the researcher to meet his advisors physically in order to exchange constructive ideas, suggestions, comments and advices on each and every major steps of this work due to the distant where the researcher and the advisors live. Time limitation has constrained the researcher as he was over loaded. The other limitations of our study is shortage of sufficient budget.

# CHAPTER TW0

## 2. REVIEW OF LITERATURE

Partial differential equation (PDE) is one in which there are two or more independent variables and the derivatives that occur in it. PDE describe many types of physical phenomena in engineering and science, ranging from transient heat equation through vibration of strings and plates to fluid flows and the behavior of electric and magnetic fields **[1].**

PDEs are classified in to elliptic, parabolic & hyperbolic equations. Not all PDEs fall in to one of these classes by any means, but most of important equations that arise in practice do. These classes of equations model different sorts of phenomena, display different behavior and require different techniques for their solution.

Finite volume method is one of numerical methods for the approximation of the solution of PDE's. In this method a given PDE is first represented by a semi integral form which is latter spatially descritized over the domain of computational region and form an algebraic scheme to approximate the numerical solution**[13]**.

The finite volume approach is one of the premier mathematical tools employed to solve partial differential equations. It is a means of obtaining numerical solutions to PDEs. FVMs are closely related to finite difference methods and they can often be interpreted directly as a finite difference .However ,FVS are derived on the basis of the integral form of the conservation law**[3].**

These methods are particularly appropriate for solving systems of conservation laws, in which the integral of the solution over each grid cell is modified only due to fluxes through the edges of the grid cell. Taylor series expansion is used to define these fluxes in order to obtain the results, accuracy better than first-order and limiters are also introduced to obtain high-resolution non-oscillatory results**[6].**

FVM have several good behaviors: they can handle general meshes, they have a compact numerical stencil, they conserve the quantity and they have stability properties under some hypothesis (CFL condition)[**2**].

As it has been reported in **[14]** the numerical result obtained by using FV method converges due to the cancelation in time of error. The author consider this result as a variation around the Lax theorem which states that stability and consistency are sufficient condition for a linear scheme to be convergent, however, there exist numeral methods, such as finite volume method in 2-D space on triangular methods.

In addition, as it has shown in **[10]** a linear advection equation with the given initial data that satisfy Lipchitz divergence free space condition FVS are not consistent in the finite difference classical sense and Lax's theorem cannot be used to prove it but if the meshes are uniformly regular and satisfy the courant- Friedrich- levy condition (CFL), then the scheme can achieve second order convergence rate in the maximun norm. The result has been proved by author and Volelle.

# CHAPTER THREE

# 3. FINITE VOLUME METHODS

## 3.1. Linear advection equations

In two space dimensions, a homogeneous first-order system of partial differential equations in space $X$ and $t$ has the form : $q_t(x,y) + A\, q_x(x,y) + B\, q_y(x,y) = 0$ In the simplest constant-coefficient equation. The function q: $\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^m$ is a vector with m components representing the unknown functions (pressure, velocity, etc.) we wish to determine its value at a given point (x, y) and time t where A& B are constant mxm real matrices. In order for this problem to be a linear hyperbolic, the matrices must have real Eigen values & a corresponding set of m linearly independent Eigen vectors**[8].**

The constant-coefficient scalar problem, in which m = 1 and the matrices A and B reduce to scalar values, is hyperbolic provided the given scalars A and B are real and constant. This problem can model either advective transport or wave motion depending on the context.

Advection is transport of a substance due to the motion of the fluid. For example consider a contaminant being advected downstream with some fluid flowing through two-dimensional pipe at constant velocity field $\vec{u} = (u, v)$.Then the concentration or density q(x, y, t) of this contaminant present in this fluid (in very small quantities that do not affect the fluid dynamics) measured in units of mass per unit volume, e.g., grams per cubic meter satisfies an equation of the form:

$$q_t(x,y,t) + uq_x(x,y,t) + vq_y(x,y,t) = 0 \qquad (3.1.1)$$

This equation has solutions of the form:

$$q(x, y, t) = \tilde{q}(x - ut, y - vt) \qquad \text{for any function } \tilde{q}(\xi). \qquad (3.1.2)$$

In this context the equation (3.1.1) is generally called the advection equation.

In this thesis work, we consider the advection of a scalar concentration or density function

q (x, y, t) in a constant velocity field $\vec{u}$ = (u(x, y, t), v(x, y, t)) of two dimensional space in which the flow is assumed to be incompressible ($\nabla \cdot \vec{u} = 0$) and We want to determine the concentration of the advected substance q whose evolution is governed by the scalar linear advection equation (3.1.1):

To find q = q (x, y, t), we solve the equation over the time interval $t \in [0, T]$ with constant horizontal and vertical velocity components: u(x, y, t) = u & v(x, y ,t) = v (resp.) and the given

initial condition for our tracer which we can think of as pollutant dumped in the ocean is a circular Gaussian hill distribution with a decay length scale of $\ell$,

$$q(x, y) = e^{-\frac{(x-x_c)^2-(y-y_c)^2}{\ell^2}} \qquad (3.1.3)$$

and the profile is located at the center $(x_c, y_c)$ of the mesh.

### 3.2. Finite volume methods

FVMs are based on volume approach, rather than a point wise approximation to the solution, the numerical solution consists of approximations to the cell averages of the solution over grid cells to properly compute discontinuous solutions we need to use finite volume methods, which are based on the integral form instead of the differential form. The most common standard methods for the solution of a linear advection equation that use the difference approximations schemes for the approximations of derivatives terms appears in the Taylor series expansion of the equation are:

- $1^{st}$-order-Upwind Method(FOU).
- $2^{nd}$ – order Lax–Wendroff Method
- Beam–Warming.

### 3.3. Derivation of conservation laws for the advection equation

We derive the conservation law by considering an arbitrary spatial domain Ω over which q is assumed to be conserved, so that the integral of q over Ω varies only due to flux across the boundary of Ω . This boundary is denoted by $\partial$ Ω.We thus have

$$\frac{d}{dt}\iint_{\Omega} q(x,\ y,\ t)dx\,dy = \text{net flux across } \partial\,\Omega \tag{3.1.4}$$

The net flux is determined by integrating the flux of q normal to $\partial\,\Omega$ around $\partial\,\Omega$.

Let f (q) is the flux of *q* in the *x*-direction (per unit length in *y*, per unit time). This means that the total flux through an interval from $(x_o\ ,\ y_o)$ to $(x_o, y_o + \Delta y)$ over a time $\Delta t$ is nearly $\Delta t\ \Delta y\ f\ (q(x_o,\ y_o))$, for $\Delta$ t and $\Delta$y sufficiently small.

Similarly, let *g*(*q*) be the flux in the *y*-direction, then the total flux through an interval from $(x_o\ ,\ y_o)$ to $(x_o + \Delta x\ , y_o)$ is $\Delta t\ \Delta x\ g\ (q(x_o\ ,\ y_o))$

Hence $\vec{f}(q) = (\ f\ (q),\ g(q))$ represent the flux vector.

Finally, let $\vec{n}\ (s) = (n^x(s),\ n^y(s))$ be the outward-pointing unit normal vector to $\partial\,\Omega$ at a point (x(s), y(s)) on $\partial\,\Omega$, where s is the arc length of $\partial\,\Omega$.

Then from vector calculus the flux at $\vec{x}\ (s) = (x(s),\ y(s))$ in the direction $\vec{n}\ (s)$ is expressed as

$$\vec{n}\ (s)\ \cdot\ \vec{f}(q(x(s),y(s),t)) = n^x(s)\ f(q)\ +\ n^y(s)\ g(q) \tag{3.1.5}$$

and (3.1..5) 3 becomes

$$\frac{d}{dt}\iint_{\Omega} \mathbf{q(x,y,t)\ dx\,dy} = -\int_{\partial\,\Omega} \vec{n}(s)\cdot\vec{f}(q)ds \tag{3.1.6}$$

If q is smooth then we can use the divergence theorem to rewrite the above as

$$\frac{d}{dt}\iint_{\Omega} \mathbf{q(x,y,t)\ dx\,dy} = -\iint_{\Omega} \vec{\nabla}\cdot\vec{f}(q)\ dx\,dy \tag{3.1.7}$$

The divergence of $\vec{f}$ is $\vec{\nabla}\ \cdot\ \vec{f}(q) = f(q)_x\ +\ g(q)_y$ ,where $\vec{\nabla} = \frac{\partial}{\partial_x}\,\hat{\imath} + \frac{\partial}{\partial_y}\hat{\jmath}$ is the differential operator. This leads to

$$\iint_{\Omega} \left[\mathbf{q_t}\ +\ \vec{\nabla}\cdot\vec{f}(q)\right]\mathbf{dx\,dy} = \mathbf{0} \tag{3.1.8}$$

since (3.1.9) must hold over any arbitrary region $\Omega$, the integrand must be zero, and thus we obtain the 2 D conservation law in differential form:

$$q_t\ +\ f(q)_x\ +\ g(q)_y = 0 \tag{3.1.9}$$

The differential equation (3.1.9) can be derived from the integral equation (3.1.4) by simple operation if q and f (q) are sufficiently smooth.

As the author Leveque in **[5]** has suggested in practice many interesting solutions are not smooth, but contain discontinuities such as shock waves. At a discontinuity in q, the partial differential equation (3.1.9) does not hold in the classical sense in which derivatives are approximated by finite differences and this indicate that the integral conservation law (3.1.4) is the more fundamental equation which does continue to hold.

### 3.3.1. Flux function

Suppose a fluid is flowing with a known velocity $\vec{u} = (u(x, y, t), v(x, y, t))$ in the plane, and let the scalar q(x, y, t) represent the concentration of a tracer, measured in units of mass per unit area in the plane. Then the flux functions are:

$$f = u(x, y, t)\, q(x, y, t),$$
$$g = v(x, y, t)\, q(x, y, t) \qquad (3.2.0)$$

so that $f(q) = \vec{u}\, q$ and we obtain the conservation law

$$q_t + (uq)_x + (vq)_y = 0 \qquad (3.2.1)$$

In this derivation f and g depend explicitly on (x, y, t) as well as on the value of q. But to be specific if $\vec{u} = (u, v)$ = constant in space & time so the fluid is simply translating at constant speed in a fixed direction, then (3.2.1) reduces to the constant coefficient variable equation (3.1.1)

In this case the flux at any point and time can be determined directly from the value of the conserved quantity at that point, and does not depend at all on the location of the point in space–time. Thus, once the flux function $f(q)$ is specified. We have an equation for $q$ to solve.

The exact or the analytical solution of (3.1.1) is then $q(x, y, t) = q^0(x - u\,t, y - v\,t)$ where $q^0$ is determined by the initial and boundary conditions.

### 3.3.2. Taylor Series

The Taylor series expansion of the function q(x) about the point $x = x_0$ is given by the formula:

$$q(x) = \sum_0^\infty \frac{q^{(n)}(x_0)}{n!}(x - x_0)$$

Where $q^{(n)}(x_0) = \frac{d^n q}{dx^n}$ $at\ x = x_0$ and $q^{(0)}(x_0) = q(x_0)$

If we let $x = x_0 + h$ then $x - x_0 = h$, and the series can be written as

$$q(x + h) = \sum_0^\infty \frac{q(x_0+h)}{n!} h^n \qquad (3.2.2)$$

$$= q(x_0) + \frac{u'(x_0)}{1!} h + \frac{u''(x_0)}{2!} h^2 + O(h^3) \qquad (3.3.3)$$

Where the expression O ($h^3$) represents the remaining terms of the series and indicates that the leading term is of order $h^3$. Because h is a small quantity, we can write $0 < h < 1$, and $h > h^2 > h^3 > h^4 > \ldots..$ Therefore, the remaining of the series represented by O ($h^3$) provides the order of the error incurred in neglecting this part of the series expansion when calculating $q(x_0 + h)$

3.3.3. The Taylor series expansion of the exact solution

The Taylor series expansion of the exact solution at a point after a single time step for the equation (3.1.1) is

$$q(x, y, t_{n+1}) = q(x, y, t_n) + \Delta t\, q_t(x, y, t_n) + \frac{1}{2}(\Delta t)^2 q_{tt}(x, y, t_n) + \ldots.$$

$$= q(x, y, t_n) - u\,\Delta t\, q_x - v\Delta t\, q_y$$

$$+ \frac{1}{2}(\Delta t)^2 \left[u^2 q_{xx} + vuq_{xy} + uvq_{yx} + v^2 q_{yy}\right] + \ldots \qquad (3.2.4)$$

This will be useful in identifying terms arising in finite volume approximations to the advection equation.

**3.4. Finite volume method and approaches to discretization**

We use a uniform grid with equal spacing $h=\Delta x = \Delta y$ in both directions. Let $C_{ij}$ be the (i, j) grid cell $[x_{i-1/2}, x_{i+1/2}] \times [y_{i-1/2}, y_{i+1/2}]$ and The value $Q_{ij}^n$ represents a cell average over the (i, j) grid cell at time $t_n$:

$$Q_{ij}^n \approx \frac{1}{\Delta x\, \Delta y} \iint_{C_{ij}} q(x, y, t_n)\, dx\, dy = \frac{1}{h^2} \iint_{C_{ij}} q(x, y, t_n)\, dx\, dy \qquad (3.2.5)$$

we use the integral form of the equation to determine how this cell average varies with time, and develop finite volume methods based on numerical approximations to the fluxes at each cell edge.

If q(x, y, t) is a smooth function, then the integral in (3.2.4) agrees with the value of q at the midpoint of the interval to $\mathcal{O}(\Delta x^2)$. By working with cell averages, however, it is easier to use important properties of the conservation law in deriving numerical methods.

In particular, we can insure that the numerical method is conservative in a way that takes on the appearance of the true solution, and this is extremely important in accurately calculating shock waves. This is because $\sum_{i,j=1}^{N} Q_{ij}^{n} h^2$ approximates the integral of q over the entire cell, and if we use a method that is in conservation form, then this discrete sum will change only due to fluxes at the boundaries. The total mass within the computational domain will be preserved, or at least will vary correctly provided the boundary conditions are properly imposed.

To solve the equation (3.1.1) so, the integral form of the conservation law (3.1.4) gives

$$\frac{d}{dt}\iint_{\Omega} q\ (x,\ y,\ t_{n+1})\, dx\, dy = -\int_{\partial\Omega} \vec{n}(s)\ \cdot\ \vec{f}(q) ds \qquad (3.2.6)$$

Then the idea is to use a function defined by these cell values as initial data $\tilde{q}^{\,n}(x\ ,\ y\ ,\ t_n)$ & Solving over time step $\Delta t = k$ with this data gives a function $\tilde{q}^{\,n}\ (x\ ,\ y\ ,\ t_{n+1})$ which is then averaged over each cell to obtain

$$Q_{ij}^{n+1} = \frac{1}{h^2}\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{i-1/2}}^{y_{i+1/2}} \tilde{q}^{\,n}(x,\ y,\ t_{n+1})\, dx\, dy \qquad (3.2.7)$$

Integrating the conservation law (3.2.7) over $[x_{i-1/2},\ x_{i+1/2},]$ x $[y_{i-1/2},\ y_{i+1/2}]$ x $[t_n\ ,\ t_{n+1}]$ and dividing both sides with the area of the cell $h^2 = \Delta x\, \Delta y$ lead to a fully discrete conservative finite volume method in flux-differencing form:

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{k}{h}\left[F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n} + G_{j+1/2}^{n} - G_{i,j-1/2}^{n}\right] \qquad (3.2.8)$$

Where

$$F_{i-1/2,j}^{n} \approx \frac{1}{kh}\int_{t_n}^{t_{n+1}}\int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i-1/2}, y, t))\, dy\, dt,$$

$$G_{i,j-1/2}^{n} \approx \frac{1}{kh}\int_{t_n}^{t_{n+1}}\int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x, y_{j-1/2}, t))\, dx\, dt \qquad (3.2.9)$$

The numerical fluxes $F^n$ and $G^n$ at each edge are typically computed from the data $Q^n$ at the initial time. We can obtain approximations to these interface fluxes by using the Taylor expansion & which can be rewritten as

$$q(x, y, t_n + k) = q - k(uq - \frac{k}{2}u^2 q_x - \frac{k}{2}uvq_y)_x$$

$$- k(vq - \frac{k}{2}v^2 q_y - \frac{k}{2}vuq_x)_y + \ldots . \quad (3.3.0)$$

This suggests that we need

$$F_{i-1/2,j} \approx u\, q(x_{i-1/2}, y_j, t_n) - \frac{k}{2}u^2 q_x(x_{i-1/2}, y_j, t_n) - \frac{k}{2}uvq_y(x_{i-1/2}, y_j, t_n),$$

$$G_{i,j-1/2} \approx v\, q(x_i, y_{j-1/2}, t_n) - \frac{k}{2}v^2 q_y(x_i, y_{i-1/2}, t_n) - \frac{k}{2}vuq_x(x_{i, y_{i-1/2,}} t_n) \quad (3.3.1)$$

It can be shown that for the equation (3.1.1) these expressions agree with the integrals to $O((\Delta t)^2)$.

### 3.5. Periodic boundary conditions & ghost cell

Most numerical methods used for updating the cell average $Q^n_{i,j}$ consider only neighboring cell values $Q^n_{i-1,j}$ , $Q^n_{i+1,j}$ , $Q^n_{i,j-1}$, and $Q^n_{i,j+1}$ and perhaps further values away as needed in order to compute the fluxes at the cell interfaces. So to set this condition, we should follow one of the following two approaches as suggested by the author Randal Leveque in **[8].**

One is to use special formulas for use near the boundaries, which will depend both on the type of boundary conditions and the numerical method we use. And the other is to extend the computational domain to include a few additional cells on either end, called ghost cells, whose values are set at the beginning of each time step depending on the boundary conditions and the interior solution.

To solve the two-dimensional advection problem on bounded physical domain $[a, b] \times [c, d]$. Periodic functions are appropriate to impose boundary conditions for any numerical method for example

$q(a, t) = g_0(t)$ , where $g_0(t)$ is a given function (3.3.2)

we can not specify a boundary condition at x = b. We may need to specify a boundary condition at x = b for the numerical method, however, if the stencil for computing $Q_i^{n+1}$ involves points to the right of $x_i$

### 3.5.1. Outflow boundaries

First we consider the outflow boundary at $x = b$. If we use a one-sided method such as upwind or Beam–Warming, then we do not need any numerical boundary condition. If we implement the method using ghost cells, then we can assign arbitrary values to the ghost cells on the right with no effect on the interior solution, since these values will never be used. If we use a three-point method such as Lax–Wendroff that does use values to the right, then some numerical boundary conditions must be specified.

To achieve stable and accurate method one is to use a fully upwind method at the rightmost point, together with the Lax–Wendroff method at all other points. This works quite well but this switching create difficult in later in stability analysis. Thus rather than explicitly switching to a different formula at the right boundary, we can achieve the same effect by setting ghost cell values by extrapolation from the interior solution. If the ghost-cell value $Q_{N+1}^n$ is set based on $Q_N^n$ , $Q_{N-1}^n$ ,.,.,.then the new value $Q_N^{n+1}$ will effectively be computed on the basis of values to the left alone, even if the formula depends on $Q_{N+1}^n$ and hence this reduces to some sort of upwind method. The simplest approach is to use a zero-order extrapolation (extrapolation by a constant function). We simply set

$$Q_{N+1}^n = Q_N^n \qquad \text{at the start of each time step} \tag{3.3.3}$$

It is recommended by [**5**] that zero-order extrapolation, decoupled with the methods we are using that perform characteristic decomposition in the solution to each Riemann problem, is often a very effective way to accomplish this process. Similarly we follow the same procedure for other Riemann problem in the y sweep to specify the required numerical condition at the upper outflow boundary.

### 3.5.2. Inflow boundary conditions

At inflow boundary x = a, where we specify the boundary condition. For the advection equation we can easily compute sensible values using our knowledge of the exact solution .We set:

$$Q_o^n = \frac{1}{\Delta x} \int_{a-\Delta x}^{a} q(x, t_n)\ dx \tag{3.3.4}$$

Here, the true solution is not defined for x < a, but we can extend the solution past the boundary using the knowledge of characteristics, setting

$$q(x, t_n) = q(a, t_n + \frac{a-x}{u}) \tag{3.3.5}$$

Similarly, using the same procedure we can compute value at the other inflow boundary y =c .

### 3.7. Constriction of grid (meshes) for FVMs.

The finite volume method subdivides the space domain into grid cells (meshes) &approximates the integral form for each cell. In order to do this, the notion of numerical flux needs to be introduced. The grid is chosen to be cell-centered.

**Definition 3.7.1.** (Cell centered grid). A cell centered grid of a spatial domain D ⊂ R x R consists of set of control volumes V = {$V_i$ ⊂ D : i = 1,...,I} and a set of storage locations X = {$\underline{x}_i$ ∈ D : i = 1,...,I} such that D = $\cup^{I}_{i=1} V_i$ and $\underline{x}_i$ is at the center of mass of Vi. G = (V,X).

Here the grid cells are assumed to be uniform Δx=Δy= h unit length rectangles $[x_{i-1/2,j}, x_{i+1/2,j}]$ × $[y_{j-1/2,j}, y_{j+1/2,j}]$ in two dimensions. The computational grid is extended by one or two rows of cells along each side and the values of q are assigned to each of these cells at the beginning of each time step.

**Remark 3.7.1.**

a) If no limiter is used then only one additional row of cells needs to be introduced. The flux value$F_{j+1/2}$ depends only on the values $q_{im}$for i= 0, 1 and m = j - 1, j, j + 1 and is determined in the usual way.

b) If limiters are used then the jump $q_{1,j}$ - $q_{0,j}$ may be limited by comparing it with $q_{0,j}$ - $q_{-1,j}$ if u > $o$and so a second row of exterior cells is needed near an inflow boundary. The method is then applied over the interior of the expanded grid so that fluxes at interfaces corresponding to the original boundary are determined.

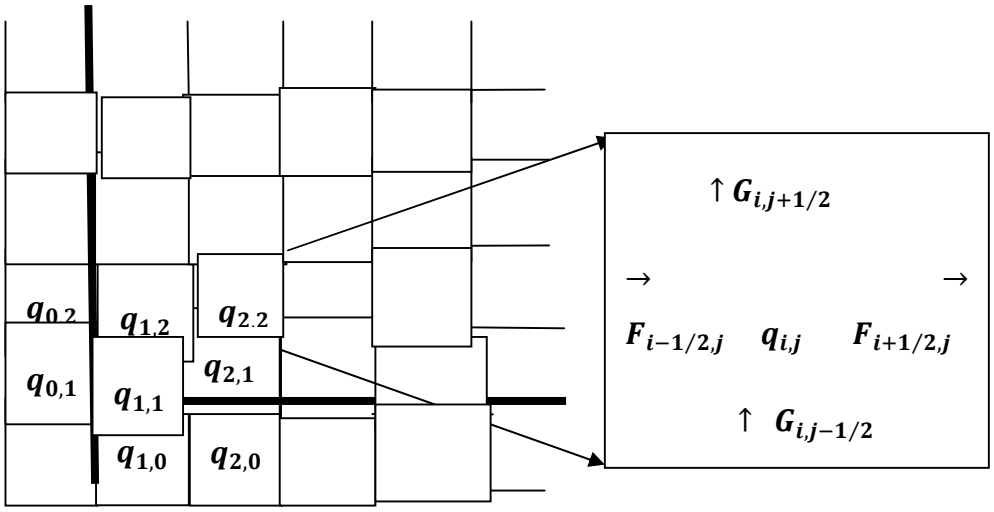


*Fig.3.1. a) Uniform Cartesian finite volume grid with cells. The solid lines represent boundary of the physical region & $q_{i,j}$ represents the average cell value at the center. b) The arrows denote the direction of fluxes in to a cell through its 4 edges in the x & y direction*.

**3.8. High-resolution FVM methods for linear advection.**

A general class of finite volume methods for conservation laws can be derived by the following sequence of steps:

1. From the given cell averages $Q_{ij}^{n}$ at time $t_n$ construct a function $\tilde{q}^{n}(x, y, t_n)$.

2. Solve the advection equation exactly with this data over a time step of length k, giving

$$\tilde{q}^{n}(x, y, t_{n+1}) = \tilde{q}(x - u\,k, y - u\,k)$$

3. Average this shifted function over the grid cells to obtain the new cell averages:

$$Q_{ij}^{n+1} = \frac{1}{h^2} \iint_{C_{ij}} \tilde{q}^{n}(x, y, t_{n+1})\, dx\, dy$$

This whole process is then repeated in the next time step. These steps are referred to as the reconstruct–evolve–average (REA) algorithm.

**Definition3.8.1.** A method is called monotonicity -preserving if $Q_i^{n} \geq Q_i^{n+1}$ for all i implies that $Q_i^{n+1} \geq Q_i^{n}$ for all i. Any TVD method is monotonicity-preserving

**Theorem 3.1.(Goodman & Leveque)**. Except in certain trivial cases, any method that is in two space dimensions is at most first-order accurate.

To derive a method that is based on the reconstruct–evolve–average approach the reconstruction must be performed in such a way that

$$\text{TV}\,(\tilde{q}^{\,n}\,(\cdot\,,\,t_n\,)) \leq \text{TV}\,(Q^n\,) \tag{3.4.6}$$

The reason is that the evolving and averaging steps cannot possibly increase the total variation. For the advection equation (3.1.1), since $\tilde{q}^{\,n}$ simply advects without changing shape at the evolve step we have,

$$\text{TV}\,(\tilde{q}^{\,n}\,(\cdot\,,\,t_{n+1}\,)) = \text{TV}(\tilde{q}^{\,n}\,(\cdot\,,\,t_n\,)) \tag{3.4.7}$$

& the averaging step gives,

$$\text{TV}\,(Q^{n+1}\,) \leq \text{TV}\,(\tilde{q}^{\,n}\,(\cdot\,,\,t_{n+1}\,)) \tag{3.4.8}$$

Combining (3.4.6) (3.4.7) and (3.4.8) then shows that the method is TVD

The particular method applied to solve PDE depends on the form of the function $\tilde{q}^{\,n}$ chosen in step 1.

As it is stated on [5] the general approach of (REA) Algorithm was originally proposed by Godunov as a method for solving the non linear Euler equations of gas dynamics. Application in that context depend on the fact that, even for this nonlinear system, the Riemann problem with piecewise constant initial data can be solved and the solution consists of a finite set of waves traveling at constant speeds,

In Godunov's original approach this reconstruction is a simple piecewise constant function, and this leads most naturally to Riemann problems, but gives only a first-order accurate method. To obtain better accuracy one might consider using a better reconstruction, for example a piecewise linear function that is allowed to have a nonzero slope in the (i , j) grid cell. This idea forms the basis for the high-resolution method.

The upwind method for the advection equation (3.1.1) can be derived as a special case of the REA algorithm, which can also be applied to systems of equations. In order to implement this procedure, we must be able to solve the equation in step 2 of the algorism. Because we are starting with piecewise constant data, this can be done using the theory of Riemann problems.

### 3.8.1. The Donor-Cell Upwind Method for Advection

This method is the simplest first-order upwind, which takes the general form:

$$Q_{i,j}^{n+1} = Q_{i,j} - \frac{\Delta t}{\Delta x}\left[u^{+}\left(Q_{i,j} - Q_{i-1,j}\right) + u^{-}\left(Q_{i+1,j} - Q_{i,j}\right)\right]$$

$$- \frac{\Delta t}{\Delta y}\left[v^{+}\left(Q_{i,j} - Q_{i,j-1}\right) + v^{-}\left(Q_{i,j+1} - Q_{i,j}\right)\right] \qquad (3.4.9)$$

Which uses an upwind approximation to the derivatives $q_x$ and $q_y$ in the O($\Delta t$ ) terms of the Taylor series expansion. The fluctuations simply given by:

$$\mathcal{A}^{\pm}\,\Delta Q_{i-1/2\,,\mathrm{j}} = u^{\pm}\left(Q_{i,j} - Q_{i-1,j}\right)$$

$$\mathcal{B}^{\pm}\,\Delta Q_{i\,,\mathrm{j}-1/2} = v^{\pm}\left(Q_{i,j} - Q_{i,j-1}\right)$$

(3.5.0)

Which are a special case of the first-order Godunov update to the cell value $Q_{i,j}$ resulting from the Riemann problems at the edge ($i - 1/2, j$)&( $i\,,\mathrm{j} - 1/2$). Then the fluxes for the method DCU (3.4.9) are :

$$F_{i-1/2\,,j} = u^{+}\left(Q_{i-1,j} + u^{-}Q_{i,j}\right)$$

$$\mathrm{G_{i\,,j-1/2}} = \mathrm{v^{+}\left(Q_{i,j-1} + v^{-}Q_{i,j}\right)} \qquad (3.5.1)$$

Where $u^{\pm}$ and $v^{\pm}$ are the positive or negative part of the flow velocity.

The fluxes are agree with the Godunov fluxes. In this case each Riemann solution consists of a single wave carrying the jump in Q between the neighboring two grid cells, propagating at speed u horizontally or at speed v vertically depending on the orientation of the two cells. Each flux in (3.4.7) approximates the amount of q flowing normal to the edge, assuming that the only contribution to this flux is from the adjacent cell on the upwind side (the donor cell). This is indicated schematically in Figure 3.2 (a).

The updated value $\mathrm{Q_{ij}^{n+1}}$ depends on only the three values $\mathrm{Q_{ij}}$ , $\mathrm{Q_{i-1\,j}}$ and $\mathrm{Q_{i\,j-1}}$ This is not correct, since the flow is really at an angle to the grid, as indicated in Figure 3.2(b) and the value $\mathrm{Q_{i-1,j-1}}$ should also affect $\mathrm{Q_{ij}^{n+1}}$.

The CFL condition suggests that this may cause stability problems, and indeed this method does not have the best possible stability properties that this method is stable only for small Δt enough that:

$$\left|\frac{u\Delta t}{\Delta x}\right| + \left|\frac{v\Delta t}{\Delta y}\right| \leq 1 \qquad (3.5.2)$$

An improved upwind method is developed by taking account of the flow direction more fully and has the stability bound:

$$\text{Maximum } \left(\left|\frac{u\Delta t}{\Delta x}\right|, \left|\frac{v\Delta t}{\Delta y}\right|\right) \leq 1 \qquad (3.5.3)$$

This is better than (3.5.2) whenever u and v are both nonzero, i.e., when flow is at an angle

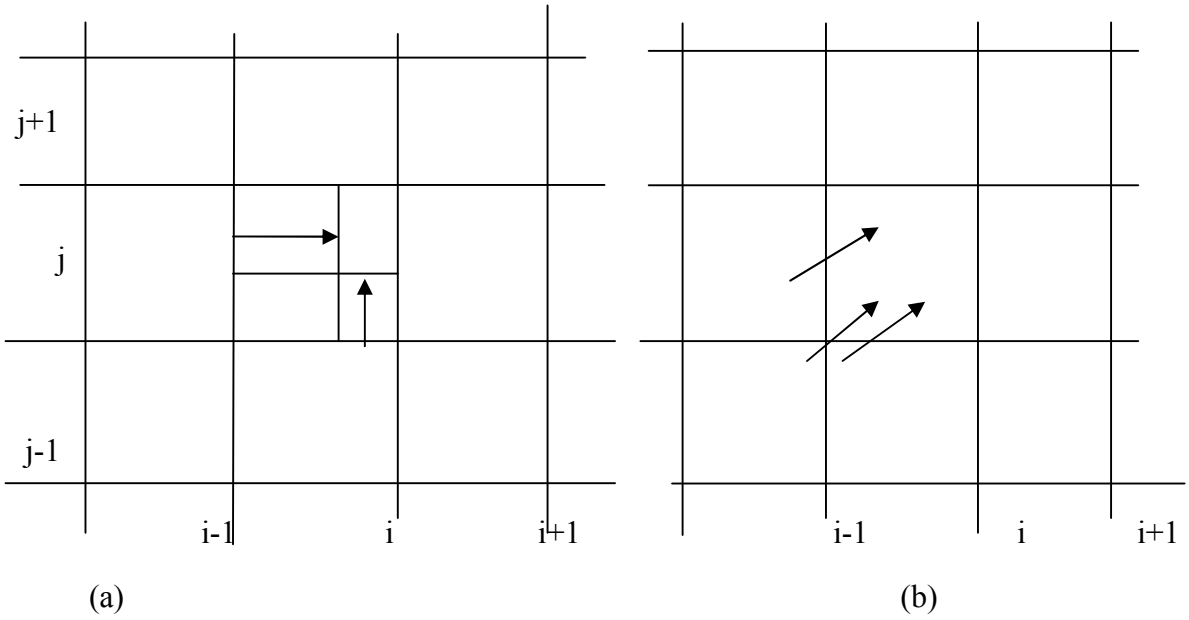


**Fig.3.2**. (a) Waves moving normal to the cell interfaces give the updates for the DCU method. (b) The true velocity (u, v) is at an angle to the grid, and information from cell (i − 1, j − 1) should also affect the new value in cell (i, j).This corner coupling is missing in the DCU method.

### 3.8.2 The Corner-Transport Upwind Method for Advection

For the advection equation, a better first-order accurate upwind method can be derived by taking the reconstruct–evolve–average approach:

View the cell averages at time $t_n$as defining a piecewise constant function $\tilde{q}^n(x, y, t_n)$ with constant $Q_{ij}^n$ value in cell $C_{ij}$.

Evolve the advection equation exactly with this data over time $\Delta t$.Average the resulting solution $\tilde{q}^{\,n}(x, y, t_{n+1})$ back onto the grid.

For the constant-coefficient advection equation this is easily done, since

$$\tilde{q}^{\,n}(x, y, t_{n+1}) = \tilde{q}^{\,n}(x - u\Delta t, y - v\Delta t)$$

The exact solution is the same piecewise constant function, simply shifted by $(u\Delta t, v\Delta t)$ So we find that

$$Q_{ij}^{n+1} = \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \tilde{q}^{\,n}(x - u\Delta t, y - v\Delta t, t_n)\, dx\, dy$$

$$= \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \tilde{q}^{\,n}(x, y, t_n)\, dx\, dy \qquad (3.5.4)$$

The new cell average $Q_{ij}^{n+1}$ is given by the cell average of $\tilde{q}^{\,n}(x, y, t_n)$ over the shaded region shown in Figure 3.3.Since $\tilde{q}^{\,n}(x, y, t_n)$ is constant in each grid cell, this reduces to a simple convex combination of four cell values.

$$Q_{ij}^{n+1} = \frac{1}{\Delta x \Delta y} [(\Delta x - u\Delta t)(\Delta y - v\Delta t)\, Q_{ij}^{n} + (\Delta x - u\Delta t)(v\Delta t)\, Q_{i,j-1}^{n} +$$

$$(\Delta y - v\Delta t)(u\Delta t)\, Q_{i-1,j}^{n} + (u\Delta t)(v\Delta t)\, Q_{i-1,j-1}^{n}] \qquad (3.5.5)$$

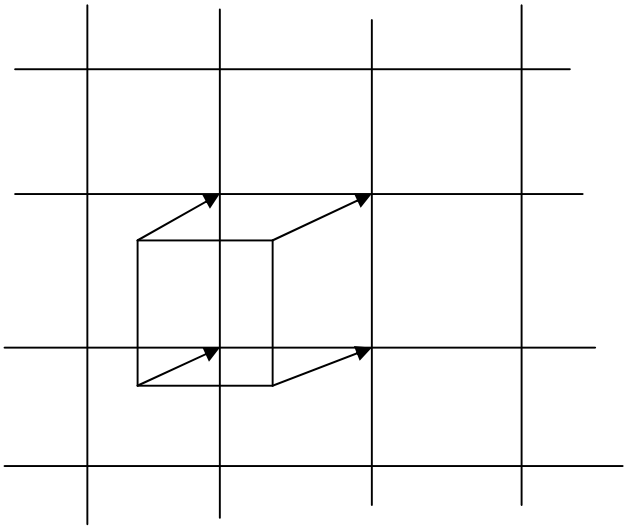

***Fig.3.3****. The corner-transport upwind method is obtained by shifting the piecewise constant data by distance ( u t , v t) and averaging back on the grid. Alternatively, the new value* $Q_{ij}^{n+1}$ *is determined by averaging the piecewise constant function over the shaded region shown in the figure*.

This can be rearranged to yield:

$$Q_{ij}^{n+1} = Q_{i,j} - \frac{u\Delta t}{\Delta x}\left(Q_{i,j} - Q_{i-1,j}\right) - \frac{v\Delta t}{\Delta y}\left(Q_{i,j} - Q_{i,j-1}\right)$$
$$+\frac{1}{2}(\Delta t)^2\left\{\frac{u}{\Delta x}\left[\frac{v}{\Delta y}\left(Q_{i,j} - Q_{i,j-1}\right) - \frac{v}{\Delta y}\left(Q_{i-1,j} - Q_{i-1,j-1}\right)\right] + \right.$$
$$\left.\frac{v}{\Delta y}\left[\frac{u}{\Delta x}\left(Q_{i,j} - Q_{i-1,j}\right) - \frac{u}{\Delta x}\left(Q_{i,j-1} - Q_{i-1,j-1}\right)\right]\right\} \qquad (3.5.6)$$

The top line of this expression corresponds to the donor-cell upwind method. The additional terms can be arranged in several different ways. They have been displayed here in a manner that relates directly to the Taylor series expansion We see that the final term in (3.5.6) models the cross-derivative terms u v $q_{yx}$ + v u $q_{xy}$in the O ($(\Delta t)^2$) term of that expansion.

The method (3.5.6) is often called corner-transport upwind (CTU) method following Colella as stated in [8] since it includes the proper transport across the corner from cell $C_{i-1,j-1}$ to $C_{i\,j}$. It is still only first-order accurate, for two reasons: It is missing an approximation to $q_{xx}$ and $q_{yy}$ terms and the approximations to $uq_x$ and $vq_y$ terms are only first order one-sided approximations.

Both of these deficiencies can be addressed by introducing slopes in the x- and y-directions separately & consequently a high-resolution version of this algorithm is easy to construct. [6]

### 3.9. Convergence, Consistency and Stability

Whenever we use a numerical method to solve a differential equation, we should be concerned about the accuracy and convergence properties of the method. In practice we must apply the method on some particular discrete grid with a finite number of points (volumes), and we wish to ensure that the numerical solution obtained is a sufficiently good approximation to the true solution. For real problems we generally do not have the true solution to compare against, and we must rely on some combination of the following techniques to gain confidence in our numerical results:

Validation on test problems, the method should be tested on simpler problems for which the true solution is known, or on problems for which a highly accurate comparison solution can be computed by other means.

Theoretical analysis of convergence and accuracy, ideally one would like to prove that the method being used converges to the correct solution as the grid is refined, and also obtain reasonable error estimates for the numerical error that will be observed on any particular finite grid.

#### 3.9.1 Convergence

In order to talk about accuracy or convergence, we need to quantify the error. In two space dimension we have an approximation $Q_{ij}^{n}$ in each grid cell when using a finite volume method. For comparison we will let $q_{ij}^{n}$ represent the exact value we are hoping to approximate well. For a finite difference method we would probably choose the point-wise value:

$$q_{ij}^{n} = q(x_i, y_j, t_n) \tag{3.5.7}$$

while for a finite volume method we might instead want to compare $Q_{ij}^{n}$ with

$$q_{ij}^{n} = \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} q(x, y, t_n)\, dx\, dy \tag{3.5.8}$$

**Remark 3.9.1** If the function q(x y, t) is sufficiently smooth, then the point wise value (3.5.7) evaluated at the cell center agrees with cell average (3.5.8) to $\mathcal{O}(\Delta x^2)$ .Comparison with the point wise value is often simpler and can be used for finite volume methods.

To discuss convergence we must first pick some finite time T over which we wish to compute. We expect errors generally to grow with time, and so it would be unreasonable to expect that any finite grid would be capable of yielding good solutions at arbitrarily large times.

**Remark 3.9.2** As we refine the grid, the number of time steps to reach time T will grow like T/Δt and go to infinity (in the limit that must be considered in convergence theory), and so even in this case we must deal with an unbounded number of time steps. We will use N to indicate the time level corresponding to time T = NΔt and we wish to obtain bounds on this grid function as the grid is refined.

For hyperbolic problems it is reasonable to assume that the ratio Δt/Δx is fixed. Then we can speak of letting $\Delta t \to 0$ to refine the grid, and speak of convergence with order s if the errors vanish like $\mathcal{O}(\Delta t^s)$ or as $\mathcal{O}(\Delta x^s)$.

If the solution q(x, y, t) is smooth, then it may be reasonable to expect point wise convergence. For problems with discontinuous solutions there will be some smearing at one or more grid points in the neighborhood of the discontinuity. In this case we cannot expect convergence in the max norm, no matter how good the method is. In such cases we don't care about point wise convergence, however. Convergence in the 1-norm is more relevant physically, and we may still hope to obtain this.

**Remark: 3.9.2. c)** The rate of convergence observed can depend on what norm is being used, and so it is important to choose an appropriate norm that yields an easier analysis, e.g., the 1-norm for conservation laws and the 2-norm for linear equations.

Study of the error introduced in a single time step, showing that the method is consistent with the differential equation and introduces a small error in any one step show that the method is stable, so that these local errors do not grow beyond the expectation and hence a bound on the global error can be obtained in terms of these local errors.

**Remark 3.9.3** If we can get a bound on the local error in an appropriate sense, then stability can be used to convert this into a bound on the global error that can be used to prove convergence.

Moreover we can determine the rate of convergence and perhaps even obtain reasonable error bounds.

### 3.9.2. Consistency

**Definition 3.9.5.Contractive Operators** The numerical solution operator N (·) is called contractive in some norm‖·‖.

$$\text{If} \quad \| N(P) - N(Q) \| \leq \| P - Q \| \tag{3.6.6}$$

for any two grid functions P and Q. If the method is contractive, then it is stable in this norm and we can obtain a bound on the global error from very simply using $P = q^n + E^n$ and $Q = q^n$:

$$\begin{aligned} \| E^{n+1} \| &\leq \| N(q^n + E^n) - N(q^n + E^n) \| + \Delta t \, \| \tau^n \| \\ &\leq \| E^n \| + \Delta t \, \| \tau^n \| \end{aligned} \tag{3.6.9}$$

Applying this recursively gives

$$\| E^N \| \leq \| E^0 \| + \Delta t \sum_{n=1}^{N-1} \| \tau^n \|$$

Suppose the local truncation error is bounded by

$$\| \tau \| \equiv \max_{0 \leq n \leq N} \| \tau^n \|$$

Then we have

$$\| E^N \| \leq \| E^0 \| + N \Delta t \, \| \tau \| \leq \| E^0 \| + T \, \| \tau \| \quad \text{for } (N \Delta t = T) \tag{3.6.8}$$

Where the term $\| E^0 \|$ measures the error in the initial data on the grid

**Remark 3.9.5.**In order to be solve the correct initial-value problem, it is required that $\| E^0 \| \to 0$ as $\Delta t \to 0$.and If the method is consistent, then also $\| \tau \| \to 0$ as $\Delta t \to 0$ and we have proved convergence. Moreover, if $\| \tau \| = (\Delta t)^s$, then the global error will have this same behavior as $\Delta t \to 0$ (if the initial data is sufficiently accurate), and the method has global order of accuracy s.

### 3.9.3. Stability

A somewhat weaker requirement on the operator N is sufficient for stability. Rather than the contractive property it is sufficient to have,

$$\| N(P) - N(Q) \| \leq (1+ \alpha\Delta t) \| P - Q \| \tag{3.6.9}$$

for all P and Q, where $\alpha$ is some constant independent of $\Delta t$ as $\Delta t \to 0$.

Then the above proof still goes through with a slight modification. We now have

$$\| E^{n+1} \| \leq (1+\alpha\Delta t) \| E^n \| + \Delta t \|\tau\|, \text{ and so}$$

$$\|E^N\| \leq (1 + \alpha\Delta t)^N \|E^0\| + \Delta t \sum_{n=1}^{N-1} (1 + \alpha\Delta t)^{N-1-n} \|\tau\|$$

$$\leq e^{\alpha T}(\|E^0\| + T\|\tau\|) \quad , \quad \text{for } (T= N\Delta t) \tag{3.7.0}$$

**Remark 3.9.6.** The error may grow exponentially in time, but this growth is bounded independently of the time step $\Delta t$. For fixed T we have a bound that goes to zero with $\Delta t$. This depends on the fact that any growth in error resulting from the operator N is at most order $\mathcal{O}(\Delta t)$ in one time step.

# CHAPTER FOUR

# 4. RESULT AND DISCUSSION

## 4.1. Results and Discussions

Consider the linear scalar advection equation: $q_t(x,y,t) + uq_x(x,y,t) + v\,q_y(x,y,t) = 0$ in the incompressible physical domain of $[0,1] \times [0,1]$ within a uniform velocity field $\vec{u}$= (u ,v)=(1,2) subject to initial data : Gaussian function $q(x,y,0) = e^{-0.01d^2}$ where

d =$\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$ is the distance between the point $(x_i, y_j)$ of cell center and the point $(x_c, y_c)$ of the center of computational mesh for length scale $\ell$= 10

The cell average $Q^0_{i,j}$ of cell (i, j)

$Q^0_{i,j}=\frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} q(x,y,0)\, dx\, dy$ have been obtained from the given initial function q(x,y,0) by using the second order accuracy approximation which take the cell averages to be the value of q at the center of the cell.

We have begun with this approximated cell averages $Q^n_{i,j}$ at time level n & solve the problem, for the sake of simplicity Dirichlet boundary conditions are used and the computation is done up to time T=1.And

At the outflow boundaries x=1&y=1, at the start of each time step the ghost cells values in the $(N+1)^{th}$ column and row is set by using ,

$$Q^n_{N+1} = Q^n_N$$

We have taken a uniform grid with equal spacing h in both directions of the plane and solve the problem for different values of h, that are chosen by considering the condition of stability bounds stated in (), to see how the algorism behaves as the grid is refined.

Let C (i, j) be the (i, j) grid cell then to appropriate the numerical flux $F^n_{i\pm1/2,j}$ the cell (i,j) at the interface between cells $C_{i-1,j}$ and $C_{i,j}$.

Using FOU(DCT) finite volume the scheme that consider one ghost cell at the end of computational domain before each time steps. These fluxes are calculated by the formula that use the cell center values of the neighboring cells in up wind directions as follows.

$$F^{n}_{i-1/2,j} = uQ_{i-1,j} \text{ , if } u>0$$

$$G_{i,j-1/2} = vQ_{i,j-1} \text{ , if } v>0$$

When the main function file stated in the appendix are executed on mat lab for different numbers of the cells on the computational domain of [0,1]x[0,1] ,the following results were displayed as follows:

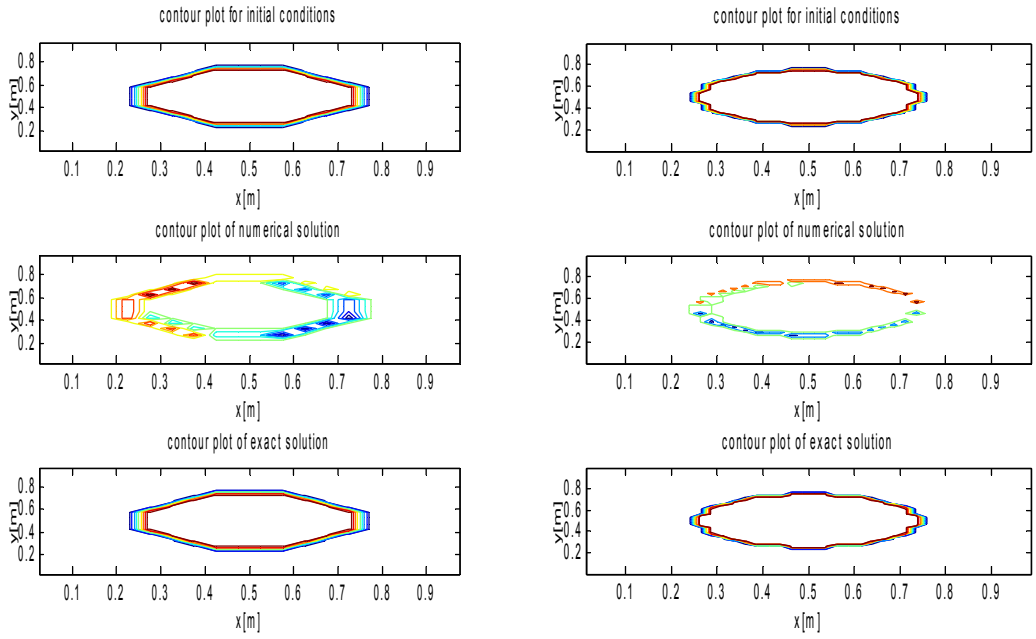


***Figure 4.2****.The contour plot of the initial condition ,the numerical and the exact solution of the test problem computed on two different grids. The plots on the left dawn from top to bottom and the plots on the right dawn from top to bottom shows the results of the initial ,the numerical and the exact solution of the equation (resp.). All plots are shown at time t and* $t + dt$ *.From left to right the grid sizes are 20x20 and 40x40.*

### 4.2.Conclutions and Recommendations

From the above results ,we have observed that the errors of the scheme employed to solve the equation for the given smooth initial data grows with time. Because as the grid refine(the number of cells decrease),the number of time steps to reach the runtime T=1 grow by fixed ratio ($T/\Delta t=0.4$) and go to infinity. We have shown this by using the time level ( N=101,201.. ) corresponding to time $T = N\Delta t$.

As for the stability of the numerical scheme, we have observed that, the presence of small cell near the boundary cell doesn't place any unreasonable restriction on the size of the time steps because time steps $\Delta t=0.4dx=0.4dy$ (for courant number 0.8) have been taken to maintain stability condition as stated by the remark (3.5.2). I finally recommended that beside refining the mesh & using only one ghost cells at the boundary. If the equation is solved by employing more than one ghost cell the better result will be obtained and I recommend this to any interested researchers in the field

## CHAPTER FIVE

## 5. REFERENCE MATERIALS


**[1].M.Y A**ttenborough, *Mathematics for Electrical engineering and computing*, An imprint of Elsevier Linacre house, Jordan Hill, Oxford xx28Dp200 wheeler Road Burlington MA 01803, first published © 2003.

**[2].M.G.C**randall and **A.M**ajda. The method of fractional steps for conservation laws. Numerische Mathematik, 34:285–314, ©1980.

**[3].D.M.C**auson,**C.G.M**ingham and L.Qian, *Introductory Finite volume methods for PDEs*, Ventus publishing Aps,ISBN 978-87-7681-882-1,© 2011.

**[4].A.S G**ilat, *Matlab Introduction with Applications* , Printed in the United States of America 10 9 8 7 6 5 4 3 2 1 $4^{th}$ Edition, November, 2010.

**[5].R.J.L**EVEQUE, *Finite Volume Methods for Hyperbolic Problems,* Cambridge university press © 2004.

**[6].R.J.L**eveque, High-resolution conservative algorithms for advection in incompressible flow .**SIAMJ**. Numerical. Anal., Vol.33: 627-665,1996.

**[7]**.**R.J.L**eveque, High resolution finite volume methods on arbitrary grids via wave propagation, J. Computational. Phys., 78 (1988), pp. 36-63

**[8]**.**R.J.L**eveque, *Hyperbolic conservation laws and numerical methods*, Von Karman Institute for Fluid Dynamics Lecture Series, Von Karmen Institute for Fluid Dynamics, Rhode-St. Genese, Belgium, 1990, pp. 1-137.

**[9].R.J.Leveque**, *Simplified multi-dimensional flux limiter methods, in Numerical Methods for Fluid Dynamics* Vol. 4, M. J. Brines and K. W. Morton, eds., Oxford University Press, 1993, pp. 175-190.

**[10].B.T M**erlet, $L^{\infty}$ −error estimation for a finite volume approximation of linear advection equation, **AMS** subject classifications, 35L6565M15.

**[11].J.N.** PEIRO' and **S. S**herwin, *Handbook of Materials Modeling*, Volume I: Methods and Models, 1–32.©2005 Springer, Printed in Netherlands.

**[12].R.B.R**OOD, Numerical advection algorithms and their role in atmospheric transport and chemistry models, Rev. Geophys.,25 (1987), pp. 71 100.

**[13].M.C.S**chafer Computational Engineering – Introduction to Numerical Methods,30, 64287 Darmstadt Germany, ISBN -10/ 3-540 30685 - 4Springer Berlin Heidelberg New York.

**[14].C.W**angshun, Numerical methods for hyperbolic conservation laws (AM257), 2006

## APPENDIX.

```
%solve the 2D advection equation,dq/dt +div(f)=0 using FOU FVscheme this equation coreosspondes to %the integral form:double integral(dq/dt dA) lineintegral(f.nds)=0 in this case flux density: f=vq% this functiion uses

clc;clf;clear all;

runtime=1.0;%target runtime[s]

t=0.0;%time[s]

tlevel=0;%time level

vx=1;vy=2;%vx&vy are velocity components in the x&y direction

v=[vx vy];% v is the velocity vector<vx,vy>

% get the coordinates of the 2D computational mesh cellcenters &solid(excluding ghost cells)by calling sub function fresdadmesh

[x,y,x_cen,y_cen,solid]=preparemesh;

%extract the size of the computational domain from the mesh data

[m,n]=size(x);

NI=m-1;% number of cells in x direction

NJ=n-1;%number of cells in y direction

%compute and store cells area in NIxNJ array using sub function -cellarea

A=cellarea40;

%compute and store side vectors in array(NI+2)x(NJ+2)x4x2 that contain ghost cells using sub function-
% sidevectors

S=sidevectors40;

%put the initial cell center value of q in an NIxNJ array using sub function-initialq to put the initial % condition

q=initialq40;
```

```
%store ploting for initial condition& copies this values to a fixed array to use it for comparision laters
disp(q)
% add extra cells around arrays to store any ghost cell values this is done by calling sub function-insertghost
q=insertghost(q);%call sub function-insertghost to append extra ghost cells
%around the initial data array now q is (NI+2)x(NJ+2)and the first
%computational cell has index(2,2)since matlab induce start at 1 due to only
%one ghost cell is is needed for this scheme
qnext=zeros(size(q));%(NI+2)x(NJ+2) array for q values at the next  time leve
A=insertghost(A);%A is now (NI+2)x(NJ+2) array
%start the time loop which run untile runtime
while(t < runtime)
   %implement bc using ghost cells q is(NI+2)x(NJ+2) & cell is
   %indexed i=2 to (NI+1),j=2 to (NJ+1)
dt=calcdt40;% find stabe time step for each itration
t=t+dt;%update time
tlevel=tlevel+1;
for j=2:NJ+1
   for i=2:NI+1
      H=interflux40(v,q,i,j);%get inteface fluxes for (i,j)cell
      H1=[H(1,1),H(1,2)];%interface flux vector for s1
      H2=[H(2,1),H(2,2)];%interface flux vector for s2
      H3=[H(3,1),H(3,2)];%interface flux vector for s3
      H4=[H(4,1),H(4,2)];%interface flux vector for s4
```

```
        %
        s1=[S(i,j,1,1),S(i,j,1,2)];%side vector for s1
        s2=[S(i,j,2,1),S(i,j,2,2)];%side vector for s2
        s3=[S(i,j,3,1),S(i,j,3,2)];%side vector for s3
        s4=[S(i,j,4,1),S(i,j,4,2)];%side vector for s4
% FVS
% compute the total flux out of cell(i,j)
totalfluxout=dot(H1,s1)+dot(H2,s2)+dot(H3,s3)+dot(H4,s4);
qnext(i,j)=q(i,j)-dt/A(i,j)*totalfluxout;
    end% for i loop
end% for j loop
q(2:NI+1,2:NJ+1)=qnext(2:NI+1,2:NJ+1);%update q values
end
disp(q)
qexact40=initialq40(x_cen,y_cen,t,v);%exact solution at time t
q=q(2:NI+1,2:NJ+1);%extract finalcomputational values
drawmesh40(x,y,solid)%draw mesh if fplotmesh is commented out
fundisp(q)%print results as a matrix for for a small mesh(if needed)
plotresults40(x_cen,y_cen,initialq40,q,qexact40);%plot results
disp('fineshed the program,see figure')
end
```