Algorithm for Solving Colocated Electromagnetic Fields Including Sources

Chris Aberle

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Aeronautics & Astronautics

University of Washington

2004

Program Authorized to Offer Degree: Aeronautics & Astronautics

University of Washington

Graduate School

This is to certify that I have examined this copy of a master's thesis by

Chris Aberle

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Uri Shumlak

_____

Scott Eberhardt

Date: _____

Signature_____

Date_____

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

Table Number Page

Chapter 1

INTRODUCTION

An electromagnetic solver, capable of simulating Maxwell Equations in two dimensions, using colocated fields with strong sources on a grid using finite volume methods, is developed. This thesis documents the equations, the algorithm, and results from simulating two benchmark problems. It is shown that using a triangular grid solves the problem of the grid effects, these effects dominate the simulations of a simple wire current source on a rectangular grid. The final solver chosen uses a weighted essentially non-oscillatory (WENO) flux scheme on a triangular grid, resulting in second order accuracy in space and time.

For the remainder of this chapter, an overview of plasma and the need for a electromagnetic solver is given, the requirements on an electromagnetic solver are given, and the general form of the solver is narrowed down to finite volume techniques.

1.1 Plasma

Plasma, "merely ionized gas" as quoted by Spock[1], commonly referred to as the "fourth state of matter", is a collection of charged ions and electrons; each species acts as a fluid and these species primarily interact on each other through coupling electromagnetic forces. Each species has a mass and charge per particle, and its collective motion can be described by a distribution function. This distribution function is seven dimensional, giving the number of particles that are at a given location with the given velocity at a specified time

$$f_\alpha = f_\alpha\left(\boldsymbol{x}, \boldsymbol{v}, t\right) \tag{1.1}$$

where

α denotes the species,

f is the density in terms of $\frac{\text{number}}{\text{m}^3 \cdot \left(\frac{\text{m}}{\text{s}}\right)^3}$,

$\boldsymbol{x}$ is a position,

$\boldsymbol{v}$ is a velocity,

t is time .

The Boltzmann equation describes how this density evolves with time and is

$$\frac{\partial f_\alpha}{\partial t} + \boldsymbol{v} \cdot \nabla f_\alpha + \frac{\boldsymbol{F}_\alpha}{m_\alpha} \cdot \nabla_{\boldsymbol{v}} \left(f_\alpha\right) = \left(\frac{\partial f_\alpha}{\partial t}\right)_{\text{collisions}} \tag{1.2}$$

where

$\boldsymbol{F}_\alpha$ is the sum of the forces acting on the fluid, with electromagnetic forces being $\boldsymbol{F}_{\alpha,\text{EM}} = q_\alpha \left(\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}\right)$,

m_α is the mass of the species particle,

$\nabla_{\boldsymbol{v}}$ means the gradient is with respect to velocity space,

q_α is the charge of a species particle,

$\boldsymbol{E}$ is the electric field (explained in the next chapter),

$\boldsymbol{B}$ is the magnetic field (explained in the next chapter) .

Directly solving the Boltzmann equation would yield the best physical accuracy, however, this is a rather intractable problem and would be computationally prohibitive to do it in a numerical fashion with reasonable accuracy. Therefore, simplifying assumptions are made to yield the plasma model, numerous models for plasmas exist and their nature depend on the phenomenon to be captured by the equations and what can be ignored. Common models are gyrokinetic, multispecies, two-fluid, and single fluid.

The electromagnetic solver of this thesis is to be incorporated into a two-fluid model, described by Loverich[2][3]; however, it is suitable for any plasma model where electromagnetic wave interaction is needed.

1.2 Electromagnetic Solver

For this thesis, a modular approach is taken which decouples the electromagnetic solver from the overall plasma solver, which assumes the presence of a plasma model that satisfies

the following properties. The plasma model gives the electromagnetic solver current and charge densities

$$\sigma = \sigma(\boldsymbol{x}, t), \tag{1.3}$$

$$\boldsymbol{J} = \boldsymbol{J}(\boldsymbol{x}, t) \tag{1.4}$$

where

σ is the charge density,

$\boldsymbol{J}$ is the current density.

In return, the electromagnetic solver gives the plasma model the electric and magnetic fields at the needed locations.

In the field of computational electromagnetics, there are a number of stock solvers; these are looked at and dismissed as not suitable for this application. Method of moments (MoM) is commonly used in antenna modeling; this is an integral method, which becomes memory and computationally intensive for dynamic systems. The finite element method (FEM) assumes a functional form for the fields (linear, quadratic, etc); this method tends to smear discontinuities, potentially blurring the shock structures that may be of interest.

The finite difference time domain (FDTD) was first popularized by Yee[4] in 1966 who introduced a compact and second order explicit centered differencing scheme whose one dimensional form is[5]

$$\frac{E_{x,k}^{n+\frac{1}{2}} - E_{x,k}^{n-\frac{1}{2}}}{\Delta t} = -\frac{1}{\epsilon_0}\frac{H_{y,k+\frac{1}{2}}^{n} - H_{y,k-\frac{1}{2}}^{n}}{\Delta z}, \tag{1.5}$$

$$\frac{H_{y,k+\frac{1}{2}}^{n+1} - H_{y,k+\frac{1}{2}}^{n}}{\Delta t} = -\frac{1}{\mu_0}\frac{H_{y,k+\frac{1}{2}}^{n+1} - H_{y,k+\frac{1}{2}}^{n}}{\Delta z} \tag{1.6}$$

where this formulation has the wave propagating along the z axis and chooses to use the electric field and the auxiliary field. Yee's scheme is a leading and well-developed electromagnetic solver, which makes this scheme's pitfalls into its main advantage in that they are knowns. A quick implementation of the FDTD showed that this algorithm was unsuitable for the plasma solver due to a consistent discrepancy between the energy injected into the domain and the energy contained within the domain; this is not acceptable in this solver as conservation of energy is critical.

The finite volume time domain (FVTD) assumes the variables are locally constant within the cell, a property that is useful in shock simulations. FVTD schemes are used in computational fluid dynamics (CFD) with the modern solvers being a variant of the Riemann solver. Applying the FVTD techniques to the Maxwell equations was done by Munz[6] with some leading work done by Shang[7]. This thesis uses the FVTD method to formulate the algorithms and the corresponding solvers. The final solver divides each cell of a rectangular grid into subcells to form a triangular grid. It is later shown that the solver based upon the triangular grid is superior to one based on the rectangular grid, for it is shown that switching to the triangular grid eliminates grid effects present in a solver based upon the rectangular grid.

Chapter 2

PHYSICAL MODEL

This chapter gives a quick overview of electric and magnetic fields then describes the physical model of the Maxwell equations, characteristics, and the physical boundary conditions.

2.1 Electric and Magnetic Fields

Like the gravitational field is a vector field that couples the forces between masses, the electric field couples the forces between charges and the magnetic field couples the forces between currents.[8]

In steady state, the electric field from a single particle is determined using Coulomb's Law at a point $\boldsymbol{r}$ from the particle.

$$\boldsymbol{E} = \frac{1}{4\pi\epsilon_0}\frac{q}{r^2}\hat{\boldsymbol{e}}_r \tag{2.1}$$

where

- $\boldsymbol{E}$ is the electric field,
- ϵ_0 is the permittivity of free space,
- q is the charge of the particle,
- r is the distance from the particle to the point of measurement, and
- $\hat{\boldsymbol{e}}_r$ is the unit vector from the particle to the point of measurement.

For many particles, the electric field is cumulative and so may be summed across all particles,

$$\boldsymbol{E} = \sum_{\alpha}^{\text{particles}} \frac{1}{4\pi\epsilon_0}\frac{q_\alpha}{r_\alpha^2}\hat{\boldsymbol{e}}_{r_\alpha} \tag{2.2}$$

where

α is the particle,

q_α is the charge of particle α,

r_α is the distance from particle α to the point of measurement, and

$\hat{\boldsymbol{e}}_{r_\alpha}$ is the unit vector from particle α to the point of measurement.

In steady state, the magnetic field from a line current is determined using the Biot-Savart law

$$\boldsymbol{B} = \frac{\mu_0}{4\pi} \int \frac{\boldsymbol{I} \times \hat{\boldsymbol{e}}_r}{r^2} \, \mathrm{d}\ell \tag{2.3}$$

where

$\boldsymbol{B}$ is the magnetic field,

μ_0 is the permeability of free space,

$\boldsymbol{I}$ is the line current,

r is the distance from the current on the line to the point of measurement,

$\hat{\boldsymbol{e}}_r$ is the unit vector from the line current to the point of measurement.

2.1.1 Constitutive Relations

Electric and magnetic fields may exist within some material (ie. air, glass, water, etc), this arises from charges and currents bound into the structure of the material. Charges and currents not bound to the material (i.e. from a battery or a lightening strike) are called free currents and charges. Electric and magnetic fields are decomposed on whether they arise from free or bound charges and currents; the exact form varies from material to material[9]

$$\boldsymbol{B} = \boldsymbol{B}\left(\boldsymbol{D}, \boldsymbol{H}\right), \tag{2.4}$$

$$\boldsymbol{E} = \boldsymbol{E}\left(\boldsymbol{D}, \boldsymbol{H}\right) \tag{2.5}$$

where

$\boldsymbol{D}$ is the electric displacement,

$\boldsymbol{H}$ is the auxiliary field.

In most materials, the relation between $\boldsymbol{B}$,$\boldsymbol{D}$,$\boldsymbol{E}$, and $\boldsymbol{H}$ is known as the constitutive relation.

The majority of materials can be classified as "linear media" meaning that they are isotropic and linear, expressed as[10]

$$\mu \boldsymbol{H} = \boldsymbol{B}, \tag{2.6}$$

$$\boldsymbol{D} = \epsilon \boldsymbol{E} \tag{2.7}$$

where

ϵ is the permittivity of the material,

ϵ_r is the relative permittivity of the material ($\epsilon = \epsilon_r \epsilon_0$),

μ is the permeability of the material,

χ_m is the magnetic susceptibility ($\mu = \mu_0 (1 + \chi_m)$),

ϵ_0 is the permittivity of free space,

μ_0 is the permeability of free space.

The speed of light propagation within a material is determined by these values

$$v^2 = \frac{1}{\mu\epsilon} = \frac{c^2}{n^2} \tag{2.8}$$

where

v is the speed of light in the material,

c is the speed of light in free space,

n is the factor of reduction which is also the index of refraction used in optics.

Depending on the material in question, χ_m and ϵ_r may be frequency dependent.

2.2 Maxwell Equations

Pioneers of the nineteenth century studied electric and magnetic fields. The efforts of Gauss, Faraday, Ampere, and others were pulled together by James Maxwell and expressed in four equations

$$\nabla \cdot \boldsymbol{E} = -\frac{\rho}{\epsilon}, \tag{2.9}$$

$$\nabla \cdot \boldsymbol{B} = 0, \tag{2.10}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\nabla \times \boldsymbol{E}, \tag{2.11}$$

$$\mu_0 \epsilon_0 \frac{\partial \boldsymbol{E}}{\partial t} = \nabla \times \boldsymbol{B} - \mu_0 \boldsymbol{J} \tag{2.12}$$

where

ρ is the charge density,

t is time,

$\boldsymbol{J}$ is the volume current density.

In a linear material, Equation (2.9) and (2.12) can be written for convenience as

$$\nabla \cdot \boldsymbol{D} = \rho_f, \tag{2.13}$$

$$\frac{\partial \boldsymbol{D}}{\partial t} = \nabla \times \boldsymbol{H} - \boldsymbol{J}_f \tag{2.14}$$

where

ρ_f is the free charge,

$\boldsymbol{J}_f$ is the free current.

Equation (2.9) is Gauss's law. Equation (2.10) states that magnetic monopoles do not exist. Equation (2.11) is Faraday's law. Equation (2.12) is Ampere's law. Collectively, Equations (2.9), (2.10), (2.11) and (2.12) are referred to as Maxwell's equations.

2.3 Characteristics

The curl of Equation (2.11) is

$$\frac{\partial}{\partial t} [\nabla \times \boldsymbol{B}] = -\nabla \times \nabla \times \boldsymbol{E}. \tag{2.15}$$

The first time derivative of Equation (2.12) is

$$\mu_0 \epsilon_0 \frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \frac{\partial}{\partial t} [\nabla \times \boldsymbol{B}] - \mu_0 \frac{\partial \boldsymbol{J}}{\partial t}. \tag{2.16}$$

The result of inserting Equation (2.15) into Equation (2.16) is

$$\mu_0 \epsilon_0 \frac{\partial^2 \boldsymbol{E}}{\partial t^2} + \nabla \times \nabla \times \boldsymbol{E} = -\mu_0 \frac{\partial \boldsymbol{J}}{\partial t}. \tag{2.17}$$

which is a hyperbolic equation in $\boldsymbol{E}$ whose characteristics for propagating are determined by

$$\boldsymbol{S} = \frac{1}{\mu_0} (\boldsymbol{E} \times \boldsymbol{B}) \tag{2.18}$$

where

$\boldsymbol{S}$ is the Poynting vector.

The interesting point is that electromagnetic waves propagate orthogonal to both the electric and magnetic fields sustaining it.

2.4 Boundaries

This section describes how electromagnetic waves interact with several typical types of boundaries and surfaces.

2.4.1 Absorbing

The absorbing boundary lets any electromagnetic wave hitting it to leave the domain while not allowing any electromagnetic waves to enter the domain. An absorbing boundary is typically used to truncate the computational domain to mimic the solution expanding into free space. In the range of visible light, an absorbing surface would be black.

2.4.2 Reflecting

The reflecting boundary rebounds any electromagnetic wave back into the domain where the angle of reflection of the wave will equal the angle of incidence. This is accomplished by reversing the component of the propagation for the electromagnetic wave that is normal to the surface. In the range of visible light, a reflecting surface would be a mirror.

2.4.3 Conducting

An electrical conducting boundary represents a surface where electrons may freely flow, like in a metal such as aluminum, copper, or gold. At the surface of this boundary, the conditions imposed are

$$\frac{\partial \boldsymbol{E}}{\partial \hat{\boldsymbol{e}}_t} = 0, \tag{2.19}$$

where

$\hat{\boldsymbol{e}}_t$ is a surface tangent.

2.4.4 Periodic, Repeating

The periodic, repeating, boundary, is not a surface but instead a statement that the nature of the problem being simulated is periodic and repeats itself identically every L distance units, where L is the length of the computational domain. The conditions are set such that any electromagnetic wave going into one side comes out on the other side of the domain with the same direction and strength as it went in.

Chapter 3

NUMERICAL ALGORITHMS

This chapter covers the numerical details of the algorithms employed to solve the Maxwell Equations.

3.1 Numerical Model

The equations to be simulated are the Maxwell equations, which are in linear media

$$\frac{\partial H}{\partial t} = -v^2 \nabla \times \boldsymbol{D}, \tag{3.1}$$

$$\frac{\partial D}{\partial t} = \nabla \times \boldsymbol{H} - \boldsymbol{J}_f \tag{3.2}$$

where

$\boldsymbol{H}$	is the auxiliary field,
$\boldsymbol{D}$	is the electric displacement,
t	is time,
v	is the speed of light in the medium (which may or not be that of c in free space),
$\boldsymbol{J}_f$	is the free current.

The fundamental quantities are grouped into a set or a vector

$$\boldsymbol{Q} \equiv \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{H} \end{bmatrix} = \begin{bmatrix} D_x \\ D_y \\ D_z \\ H_x \\ H_y \\ H_z \end{bmatrix} \tag{3.3}$$

where

D_x is the x-component of the electric displacement,

D_y is the y-component of the electric displacement,

D_z is the z-component of the electric displacement,

H_x is the x-component of the auxiliary field,

H_y is the y-component of the auxiliary field,

H_z is the z-component of the auxiliary field.

The group form of the maxwell equations is

$$\frac{\partial}{\partial t}\begin{bmatrix} D_x \\ D_y \\ D_z \\ H_x \\ H_y \\ H_z \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} 0 \\ H_z \\ -H_y \\ 0 \\ -v^2 D_z \\ v^2 D_y \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} -H_z \\ 0 \\ H_x \\ v^2 D_z \\ 0 \\ -v^2 D_x \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} H_y \\ -H_x \\ 0 \\ -v^2 D_y \\ D_x \\ 0 \end{bmatrix} = \begin{bmatrix} -\sigma D_x - J_x \\ -\sigma D_y - J_y \\ -\sigma D_z - J_z \\ -\sigma H_x \\ -\sigma H_y \\ -\sigma H_z \end{bmatrix} \tag{3.4}$$

where

σ is the coefficient of artificial damping (explained later).

Define

$$\boldsymbol{F}_x \equiv \begin{bmatrix} 0 \\ H_z \\ -H_y \\ 0 \\ -v^2 D_z \\ v^2 D_y \end{bmatrix}, \quad \boldsymbol{F}_y \equiv \begin{bmatrix} -H_z \\ 0 \\ H_x \\ v^2 D_z \\ 0 \\ -v^2 D_x \end{bmatrix}, \quad \boldsymbol{F}_z \equiv \begin{bmatrix} H_y \\ -H_x \\ 0 \\ -v^2 D_y \\ D_x \\ 0 \end{bmatrix}, \quad \boldsymbol{\mathcal{J}} \equiv \begin{bmatrix} J_x \\ J_y \\ J_z \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.5}$$

which allows for Maxwell's equations to be expressed as a conservation law

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} + \frac{\partial \boldsymbol{F}_z}{\partial z} = -\boldsymbol{\mathcal{J}} - \sigma \boldsymbol{Q} \tag{3.6}$$

which can be further reduced using

$$\mathsf{F} \equiv \boldsymbol{F}_x \hat{\boldsymbol{e}}_x + \boldsymbol{F}_y \hat{\boldsymbol{e}}_y + \boldsymbol{F}_z \hat{\boldsymbol{e}}_z \tag{3.7}$$

to

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \nabla \cdot \mathsf{F} = -\boldsymbol{\mathcal{J}} - \sigma \boldsymbol{Q}. \tag{3.8}$$

3.1.1 Characteristics, the Waves

The characteristics of the equations along the x-axis are computed by determining the eigenvalues and eigenvectors of the flux Jacobian A. The flux Jacobian along the x-axis is

$$A \equiv \frac{\partial \boldsymbol{F}_x}{\partial \boldsymbol{Q}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -v^2 & 0 & 0 & 0 \\ 0 & v^2 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.9}$$

where

A is the flux Jacobian $\frac{\partial \boldsymbol{F}_x}{\partial \boldsymbol{Q}}$.

Each eigenvalue/eigenvector corresponds to a wave in the system, the six speeds are $+v$, $+v$, $-v$, $-v$, 0, and 0 which means that there are four waves along the x-axis. The vectors representing the $+v$ characteristics are

$$\boldsymbol{\lambda}_1^+ \equiv \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -v \end{bmatrix}^T, \tag{3.10}$$

$$\boldsymbol{\lambda}_2^+ \equiv \begin{bmatrix} 0 & 0 & 1 & 0 & v & 0 \end{bmatrix}^T \tag{3.11}$$

where

$\boldsymbol{\lambda}_1^+$,$\boldsymbol{\lambda}_2^+$ are the symbols assigned to represent the $+v$ characteristics.

The vectors representing the $-v$ characteristics are

$$\boldsymbol{\lambda}_1^- \equiv \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & v \end{bmatrix}^T, \tag{3.12}$$

$$\boldsymbol{\lambda}_2^- \equiv \begin{bmatrix} 0 & 0 & 1 & 0 & -v & 0 \end{bmatrix}^T. \tag{3.13}$$

The vectors representing the 0 characteristics are

$$\boldsymbol{\lambda}_1^0 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{3.14}$$

$$\boldsymbol{\lambda}_2^0 \equiv \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T. \tag{3.15}$$

The eigenvalues and eigenvectors can be expressed in matrix form; the order is assigned to be λ_1^-, λ_2^-, λ_1^+, λ_2^+, λ_1^0, and λ_2^0. The eigenvectors are consolidated as column vectors into

$$\lambda \equiv \left\{ \boldsymbol{\lambda_1^-} | \boldsymbol{\lambda_2^-} | \boldsymbol{\lambda_1^+} | \boldsymbol{\lambda_2^+} | \boldsymbol{\lambda_1^0} | \boldsymbol{\lambda_2^0} \right\} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -v & 0 & v & 0 & 0 \\ v & 0 & -v & 0 & 0 & 0 \end{bmatrix} \tag{3.16}$$

while the eigenvalues are consolidate into a diagonal matrix

$$\Lambda \equiv \begin{bmatrix} \lambda_1^- & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2^- & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_1^+ & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2^+ & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_1^0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_1^0 \end{bmatrix} = \begin{bmatrix} -v & 0 & 0 & 0 & 0 & 0 \\ 0 & -v & 0 & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 0 & 0 \\ 0 & 0 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.17}$$

with the directionally sensitive matrices only keep the eigenvalues of interest leading to the notation

$$\Lambda^+ \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 0 & 0 \\ 0 & 0 & 0 & v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.18}$$

$$\Lambda^- \equiv \begin{bmatrix} -v & 0 & 0 & 0 & 0 & 0 \\ 0 & -v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.19}$$

The matrix form allows for the following relations

$$A = \lambda \Lambda \lambda^{-1}, \tag{3.20}$$

$$A^+ = \lambda \Lambda^+ \lambda^{-1}, \tag{3.21}$$

$$A^- = \lambda \Lambda^- \lambda^{-1} \tag{3.22}$$

where

A^+ represents the positive flowing flux Jacobian, and

A^- represents the negative flowing flux Jacobian.

3.1.2 Splitting

The equations are split along hyperbolic fluxes and the source terms of the equations. For a finite time step, the equations are solved independently,

$$\frac{\partial Q}{\partial t} + \nabla \cdot F = 0, \tag{3.23}$$

$$\frac{\partial Q}{\partial t} = -\boldsymbol{\mathcal{J}} - \sigma \boldsymbol{Q}. \tag{3.24}$$

For each step in time, Equation (3.23) is solved first, then Equation (3.24) is solved.

3.2 Finite Volume Formulation and Grids

The finite volume formulation is a degeneracy of the finite element method and is commonly used in modern CFD codes. This formulation is applied to the Maxwell Equations starting with the balance law of Equation 3.8,

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \nabla \cdot F = \boldsymbol{\mathcal{J}} - \sigma \boldsymbol{Q}. \tag{3.25}$$

Finite volume schemes assume $\boldsymbol{Q}$ is constant across some region $\mathcal{R}$ and integrate across the region which gives

$$\iiint_{\mathcal{R}} \frac{\partial \boldsymbol{Q}}{\partial t} \, \mathrm{d}\mathcal{R} + \iiint_{\mathcal{R}} \nabla \cdot F \, \mathrm{d}\mathcal{R} = \iiint_{\mathcal{R}} \boldsymbol{\mathcal{J}} \, \mathrm{d}\mathcal{R}, \tag{3.26}$$

$$V \frac{\partial \boldsymbol{Q}}{\partial t} + \oiint_{d\mathcal{R}} F \cdot \mathrm{d}\mathcal{R} = V \boldsymbol{\mathcal{J}} \tag{3.27}$$

where

V is the volume of region $\mathcal{R}$.

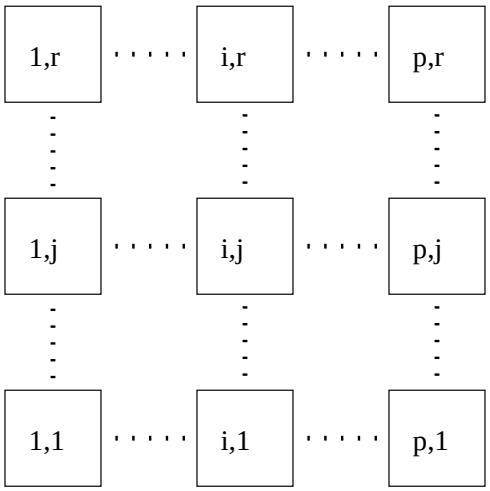


Figure 3.1: This shows the logical indices for the Cartesian two-dimensional grid.

The region $\mathcal{R}$ is normally a polyhedron with a fixed number of sides, this is simplified to

$$V\frac{\partial \boldsymbol{Q}}{\partial t} + \sum_{\beta} S_\beta \boldsymbol{F} \cdot \boldsymbol{n}_\beta = V\boldsymbol{\mathcal{J}} \tag{3.28}$$

where

β indicates the side,

S_β is the area of the side,

$\boldsymbol{n}_\beta$ is the unit normal to the side.

Using this formulation, the flux jacobian can be determine for any other direction by a simple coordinate transformation ont he matrix.

3.2.1 Grids

The solver uses a two-dimensional Cartesian grid to represent the computational domain, where i, j are used as coordinates as shown in Figure 3.1. Each i, j point in the grid represents a grid cell, which gives each cell four neighbor cells. The surface joining two cells is the cell interface, with the interface between cells i, j and $i-1, j$ being denoted as $I_{i,j}$ and the interface between cells i, j and $i, j-1$ being denoted as $J_{i,j}$.

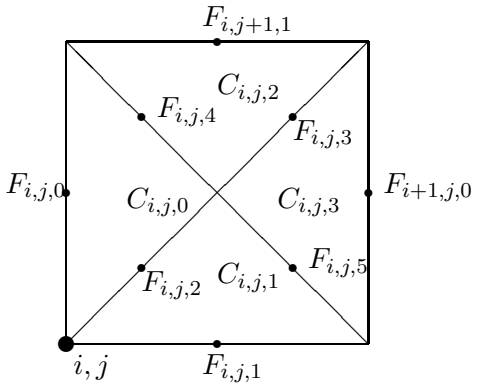


Figure 3.2: This shows how the i, j cell is divided into subcells and the subinterfaces between the subcells. C denote the subcells, there are four of these per grid cell. F denote the subinterfaces, there are six of these per grid cell. The dots in the middle of the subinterfaces denote the center of each subinterface.

Subcells

The solver makes use of two different grid configurations, the rectangular grid and the triangular grid. The rectangular grid refers to the Cartesian grid as just described. The triangular grid divides each cell of the Cartesian grid into four subcells which makes for six subinterfaces per cell; these subcells and subinterfaces are indexed by a third integer k which range from 0 to 3 for subcells and from 0 to 5 for subinterfaces which is shown in Figure 3.2.

Coordinates on the Interface

A local coordinate system is imposed on every interface and subinterface; this is done because the solver makes no assumption to the physical layout of the grid in three dimensional x, y, z space. Local coordinates use a normal-tangent-tangent n, t_1, t_2 system where $\hat{\mathbf{e}}_n$ is

normal (across) the interface, $\hat{\boldsymbol{e}}_{t_1}$ and $\hat{\boldsymbol{e}}_{t_2}$ are in the plane of the interface; this forms an orthogonal coordinate system. The conversion matrix between the coordinate systems is defined as

$$\boldsymbol{\Phi} \equiv \begin{bmatrix} n_x & T_{1_x} & T_{2_x} & 0 & 0 & 0 \\ n_y & T_{1_y} & T_{2_y} & 0 & 0 & 0 \\ n_z & T_{1_z} & T_{2_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & n_x & T_{1_x} & T_{2_x} \\ 0 & 0 & 0 & n_y & T_{1_y} & T_{2_y} \\ 0 & 0 & 0 & n_z & T_{1_z} & T_{2_z} \end{bmatrix}. \tag{3.29}$$

When Φ is used for a interface between the regular cells, the coordinates are denoted as $\Phi_{I_{i,j}}$. When Φ is used for a subinterface, the coordinates are denoted as $\Phi_{i,j,k}$.

Ghost Cells

Ghost cells are added to implement the boundary conditions (described later) onto the solution. These ghost cells are added by tacking on several rows of exterior cells to each side of the computation domain.

Memory Layout

The memory layout is the relation between the computational domain and how that data is laid out in the computer memory. A mapping function is defined

$$N = \mho_{i,j,k}(i,j,k) \tag{3.30}$$

where

i,j,k	is the three dimensional integer coordinates,
N	is the index of the memory location with respect to the start of the array, where $N \in [0, \text{Size})$.

Which is expanded to

$$N = \begin{bmatrix} \mho_i & \mho_j & \mho_k & \mho_{\text{base}} \end{bmatrix} \begin{bmatrix} i \\ j \\ k \\ 1 \end{bmatrix} \tag{3.31}$$

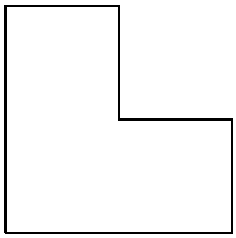


Figure 3.3: This is an example of a non-rectangular domain that would benefit from domain decomposition by splitting the domain into three blocks.

or

$$N = i\mho_i + j\mho_j + k\mho_k + \mho_{\text{base}} \tag{3.32}$$

where the $\mho$ coefficients are calculated appropriately.

Domain Decomposition and the Block

Domain decomposition is a technique to break the computational domain into smaller chunks, called blocks, to facilitate non-rectangular domains, assorted boundary conditions, and/or parallel processing. This technique can be applied to domains that, while not rectangles, can be decomposed into rectangles such as the example shown in Figure 3.3 which can be readily decomposed into three blocks. This technique would be applied in situations where more flexibility is needed in specifying the boundary conditions such as in the example shown in Figure 3.4; the boundary conditions are specified on a face-by-face or a side-by-side basis, therefore, the multiple conditions in the Figure would force the domain be decomposed into at least two blocks. This technique is also applied when the computational domain is to be split to allow for parallel processing on multiple processors or even multiple machines, the domain is divided into blocks and then the each processor or ma-

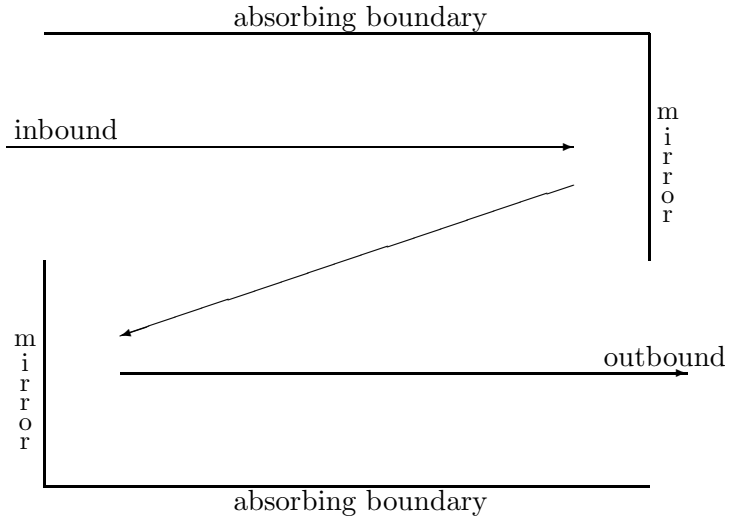


Figure 3.4: This is an example of the boundary conditions forcing the domain decomposition into blocks so that boundary conditions may be properly applied since boundary conditions are specified on a face-by-face basis for each block.

chine is assigned its blocks. When multiple blocks are used, a neighbor boundary condition is imposed between adjacent blocks to ensure continuity between the blocks.

3.2.2 Hyperbolic and Source Terms

This solver splits the governing balance law into two parts, the hyperbolic terms given by Equation (3.23) and the source terms given by Equation (3.24). The solution for the next time step advance is computed by first solving for the hyperbolic advance, then using those results to solve for the source contribution.

In practice, the solver described by this document is a group of algorithms contained within the code of a single program. That program used an input file to configure the grid and to select which algorithms to use for specific tasks within the solver.

3.3 Hyperbolic Terms

The hyperbolic terms are solved with two steps. First, the fluxes across a cell interface are calculated for each and every interface or subinterface if using subcells; this step is the flux calculation. Second, the fluxes are then used to advance the solution in Equation (3.23); this step is the time advance.

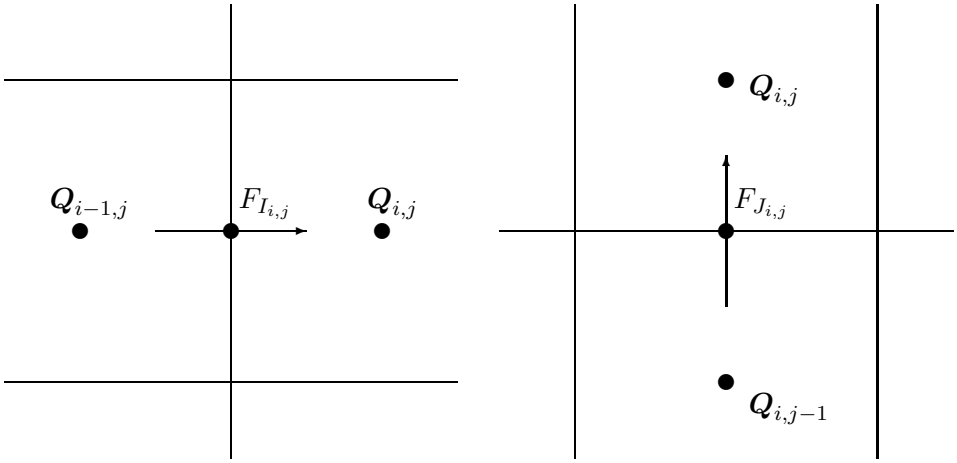
3.3.1 Upwind Flux

The calculation of fluxes across a cell interface is a non trivial matter since the conserved variable $\boldsymbol{Q}$ is normally discontinuous across the interface, where the flux across a cell interface is defined to be

$$\boldsymbol{F}_\alpha \equiv \mathsf{F} \cdot \boldsymbol{n}_\alpha \tag{3.33}$$

which is computed from $\boldsymbol{Q}$ using a flux scheme.

During the flux calculation, the focus is upon the interface, not the center of the cells; therefore, an assumed flow is assigned for each interface and subinterface. The assumed flow on a rectangular grid is shown in Figure 3.5. The assumed flow on a triangular grid is shown in Figure 3.6. The footprint is the range of cells used to interpolate the flux to a particular interface.



(a) Flux direction at interface $I_{i,j}$ (b) Flux direction at interface $J_{i,j}$.

Figure 3.5: Assumed flux direction at the interfaces for a rectangular grid.

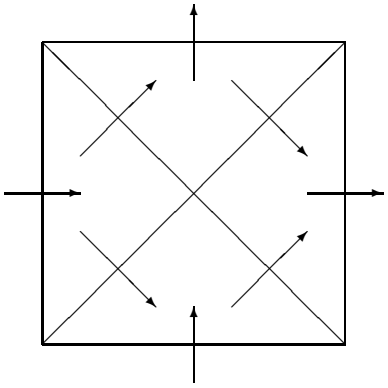


Figure 3.6: Assumed flux direction at the subinterfaces for a triangular grid.

An upwind flux scheme is a scheme that respects the characteristic directions of the hyperbolic equations by using an eigensystem decomposition, where the eigensystem is derived from the flux Jacobian. For each of the following schemes, the same technique is employed. First, the conserved quantities of each cell within the interface footprint is expressed into the local coordinate system of the interface.

$$\boldsymbol{q}_\gamma = \boldsymbol{\Phi}_\beta \boldsymbol{Q}_\gamma \tag{3.34}$$

where

$\boldsymbol{q}$ is the conserved quantities in local coordinates,

β denotes the interface or sub-interface across which the flux is to be determined,

γ denotes a cell or sub-cell in the

Then, the flux at the center of each cell or sub-cell is computed

$$\boldsymbol{f}_\gamma = f_{\text{normal}}\left(\boldsymbol{q}_\gamma\right) = \mathit{A}\boldsymbol{q}_\gamma \tag{3.35}$$

where

$\boldsymbol{f}_\gamma$ denotes the flux as measured at the center of cell or subcell γ,

f_{normal} is the flux function defined from the original physics equation that computes flux from known conserved quantities,

A is the flux Jacobian $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{Q}}$ computed from the flux function.

Each cell flux is then decomposed into eigenwaves, based on the eigensystem, and the Riemann invariants along each characteristic are calculated; these eigenwaves are separated and classified as positive and negative based upon the sign of the corresponding eigenvalue, therefore

$$w_{\psi,\beta} = \boldsymbol{\lambda}_\psi \cdot \boldsymbol{f}_\beta \tag{3.36}$$

where

ψ represents a particular eigenwave,

$\boldsymbol{\lambda}_\psi$ represents the eigenvector of the eigenwave,

$w_{\psi,\beta}$ is the Riemann invariant as computed lying along the eigenwave.

The Riemann invariants along the eigenwaves at the interface or subinterface are determined by interpolating the values as measured at the center of the cells or subcells to the center point on the cell interface or subinterface; each of the flux schemes only differ in this interpolation in both technique and accuracy and it is this difference that will lead to the different sized footprints. The interpolated Riemann invariants are then converted into a flux at the interface or subinterface by reversing the decomposition such that

$$\boldsymbol{F}_\alpha = \Phi_\alpha^T \sum_\gamma w_\gamma \boldsymbol{\lambda}_\gamma \tag{3.37}$$

where

$\boldsymbol{F}_\alpha$ is the flux across the interface or subinterface with the quantities expressed in global coordinates,

w_γ is the interpolated Riemann invariant at the interface or subinterface for the particular eigenwave γ.

The following schemes each compute the flux across the interface or subinterface, however, they only differ in the interpolation, therefore, the discussion will be limited to interpolation and any further simplifications.

First Order Upwind Flux

The first order upwind flux interpolates by assuming that the immediately adjacent cells or subcells have a constant Riemann Invariant, each one taken from the cell or subcell directly upwind of the interface or subinterface. For this scheme, all of the operations are linear and can therefore be simplified into fewer operations. Define

$$A^+ \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}v & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2}v & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2}v^2 & 0 & \frac{1}{2}v & 0 \\ 0 & \frac{1}{2}v^2 & 0 & 0 & 0 & \frac{1}{2}v \end{bmatrix}, \quad A^- \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2}v & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & -\frac{1}{2}v & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2}v^2 & 0 & -\frac{1}{2}v & 0 \\ 0 & \frac{1}{2}v^2 & 0 & 0 & 0 & -\frac{1}{2}v \end{bmatrix} \tag{3.38}$$

where

$$A^+ + A^- = \frac{\partial \boldsymbol{F}_x}{\partial \boldsymbol{Q}}. \tag{3.39}$$

When multiplied against a vector $\boldsymbol{Q}$, A^+ gives the flux flowing along characteristics in the positive x direction, whereas A^- gives the flux flowing along characteristics in the negative x direction. This is then applied on each interface in the following pattern

$$\boldsymbol{F}_\alpha = \left(\Phi_\alpha A^+ \Phi_\alpha^T\right) \boldsymbol{Q}_\alpha^+ + \left(\Phi_\alpha A^- \Phi_\alpha^T\right) \boldsymbol{Q}_\alpha^- \tag{3.40}$$

where

- $\boldsymbol{F}$ is the flux across the interface in the direction of the assigned flow,
- Φ_α is the coordinate transform of the interface,
- $\boldsymbol{Q}_\alpha^+$ is the conserved variables from the cell designated as the "+" contributor for the interface α,
- $\boldsymbol{Q}_\alpha^+$ is the conserved variables from the cell designated as the "-" contributor for the interface α.

The pattern is then applied to both types of interfaces for the rectangular grid

$$\boldsymbol{F}_{I_{i,j}} = \left(\Phi_{I_{i,j}} A^+ \Phi_{I_{i,j}}^T\right) \boldsymbol{Q}_{i-1,j} + \left(\Phi_{I_{i,j}} A^- \Phi_{I_{i,j}}^T\right) \boldsymbol{Q}_{i,j}, \tag{3.41}$$

$$\boldsymbol{F}_{J_{i,j}} = \left(\Phi_{J_{i,j}} A^+ \Phi_{J_{i,j}}^T\right) \boldsymbol{Q}_{i,j-1} + \left(\Phi_{J_{i,j}} A^- \Phi_{J_{i,j}}^T\right) \boldsymbol{Q}_{i,j}. \tag{3.42}$$

The pattern is then applied to all the subinterfaces for the triangular grid

$$\boldsymbol{F}_{i,j,0} = \left(\Phi_{i,j,0} A^+ \Phi_{i,j,0}^T\right) \boldsymbol{Q}_{i-1,j,3} + \left(\Phi_{i,j,0} A^- \Phi_{i,j,0}^T\right) \boldsymbol{Q}_{i,j,0}, \tag{3.43}$$

$$\boldsymbol{F}_{i,j,1} = \left(\Phi_{i,j,1} A^+ \Phi_{i,j,1}^T\right) \boldsymbol{Q}_{i,j-1,2} + \left(\Phi_{i,j,1} A^- \Phi_{i,j,1}^T\right) \boldsymbol{Q}_{i,j,1}, \tag{3.44}$$

$$\boldsymbol{F}_{i,j,2} = \left(\Phi_{i,j,2} A^+ \Phi_{i,j,2}^T\right) \boldsymbol{Q}_{i,j,0} + \left(\Phi_{i,j,2} A^- \Phi_{i,j,2}^T\right) \boldsymbol{Q}_{i,j,1}, \tag{3.45}$$

$$\boldsymbol{F}_{i,j,3} = \left(\Phi_{i,j,3} A^+ \Phi_{i,j,3}^T\right) \boldsymbol{Q}_{i,j,2} + \left(\Phi_{i,j,3} A^- \Phi_{i,j,3}^T\right) \boldsymbol{Q}_{i,j,3}, \tag{3.46}$$

$$\boldsymbol{F}_{i,j,4} = \left(\Phi_{i,j,4} A^+ \Phi_{i,j,4}^T\right) \boldsymbol{Q}_{i,j,0} + \left(\Phi_{i,j,4} A^- \Phi_{i,j,4}^T\right) \boldsymbol{Q}_{i,j,2}, \tag{3.47}$$

$$\boldsymbol{F}_{i,j,5} = \left(\Phi_{i,j,5} A^+ \Phi_{i,j,5}^T\right) \boldsymbol{Q}_{i,j,1} + \left(\Phi_{i,j,5} A^- \Phi_{i,j,5}^T\right) \boldsymbol{Q}_{i,j,3}. \tag{3.48}$$

WENO in One Dimension

This section describes the WENO interpolation in one-dimension.

WENO schemes are based on ENO (essentially non-oscillatory) schemes. The key idea of ENO schemes is to use the smoothest stencil among several candidates to approximate the flux at the cell interface to a high order accuracy and at the same time avoid spurious oscillations near shocks and steep gradients.[11]

For r-th order 1D-ENO, r stencils are used, each spanning r cells include cell i for the cell interface located at $x_{i+\frac{1}{2}}$. Denote each stencil with

$$S_k = \{x_{i+k-r+1}, x_{i+k-r+2}, \cdots, x_{i+k}\}$$

that approximates the Riemann invariant at the cell interface by

$$w^k_{i+\frac{1}{2}} \approx \sum_l a^k_l w_l \tag{3.49}$$

where

a^k_l	is the coefficient,
l	indexes through each term of the interpolating polynomial of the given stencil,
k	indicates the stencil being used.

ENO uses the stencil with the smoothest data in w which is determined by calculating the roughness

$$\mathcal{R}^k = \sum_{l=1}^{r-1} \left[(\Delta x)^l \, \frac{\partial^l}{\partial x^l} \, w \Big|_k \right]^2 \tag{3.50}$$

where

$\mathcal{R}^k$	is the roughness coefficient for stencil k,
$\frac{\partial^l}{\partial x^l} \, w$	is the l-th derivative of the Riemann invariant with respect to x as measured at the cell interface using the points of stencil k.

Since each stencil is r-th order accurate, ENO schemes are uniformly high order accurate; however, WENO can improve the accuracy in regions of smooth w by using a convex combination of all the candidate stencils. This weight is computed

$$\omega^k = \frac{\eta^k}{\sum \eta} \tag{3.51}$$

where

ω^k is the weight used for stencil k,

η^k is the tentative weight for stencil k

where

$$\eta^k = \frac{\mathcal{C}^k}{(\epsilon + \mathcal{R}^k)^p} \tag{3.52}$$

where

$\mathcal{C}^k$ is the optimal weight in a completely smooth solution,

p is the power, which is chosen to be two for this thesis, but can be higher,

ϵ is a positive but small real number used to avoid a zero denominator ($\epsilon = 10^{-6}$ is used in this thesis).

The optimal weight $\mathcal{C}^k$ is chosen such that on a smooth solution, the resulting interpolation is $2r-1$ order accurate as all $2r-1$ cells would be weighted in as if a single stencil $2r-1$ in size were used. However, the weights must be non-negative for stability–Shu[12] discusses a way to deal with negative weights. For the purposes of this thesis, if negative weights occur, then $\mathcal{C}^k = \frac{1}{r}$ is assigned.

For the case of $r = 3$, there are three stencils, so the coefficients are expanded to give

$$\begin{aligned}
w^1_{i+\frac{1}{2}} &= \frac{1}{3}w_{i-2} - \frac{7}{6}w_{i-1} + \frac{11}{6}w_i, \\
w^2_{i+\frac{1}{2}} &= -\frac{1}{6}w_{i-1} + \frac{5}{6}w_i + \frac{1}{3}w_{i+1}, \\
w^3_{i+\frac{1}{2}} &= \frac{1}{3}w_i + \frac{5}{6}w_{i+1} - \frac{1}{6}w_{i+2}, \\
\mathcal{R}^1 &= \frac{13}{12}\left(w_{i-2} - 2w_{i-1} + w_i\right)^2 + \frac{1}{4}\left(w_{i-2} - 4w_{i-1} + 3w_i\right)^2, \\
\mathcal{R}^2 &= \frac{13}{12}\left(w_{i-1} - 2w_i + w_{i+1}\right)^2 + \frac{1}{4}\left(w_{i-1} - w_{i+1}\right)^2, \\
\mathcal{R}^3 &= \frac{13}{12}\left(w_i - 2w_{i+1} + w_{i+2}\right)^2 + \frac{1}{4}\left(3w_i - 4f_{i+1} + w_{i+2}\right)^2, \\
\mathcal{C}^1 &= \frac{1}{10}, \\
\mathcal{C}^2 &= \frac{6}{10}, \\
\mathcal{C}^3 &= \frac{3}{10},
\end{aligned}$$

where the other values are determined using the formulas as given.

WENO on the Triangular Mesh

This section extends the WENO interpolation of the previous section onto the two-dimensional grid with subcells. Bookkeeping of all of the appropriate coefficients and weights is error prone. In future codes, the bookkeeping will be automated, with all coefficients computed as a part of the solver initialization. For this thesis, the bookkeeping was done by hand, with the coefficients given in the tables and hand coded into the solver. The scheme developed in this section assumes that the grid is uniform and regular, and that $\Delta x = \Delta y$.

This scheme uses the subcells in clusters. The biggest cluster is all of the subcells used to compute the subinterface of interest; this cluster is the *footprint*. The footprint may be qualified to include some of the subcells; the positive footprint are those subcells used to compute the positive going information at the subinterface of interest, while the negative footprint are those subcells used to compute the negative going information. The smallest cluster is the stencil, which are the subcells used to produce a single interpolation.

Bookkeeping for the WENO scheme is intricate; the positive footprint for the subinterface 0 will be analyzed in detail. After that, the details for the remaing footprints will simply be given to permit independent reconstruction of the solver.

For second order accuracy on the positive footprint, shown in full in Figure 3.7, up to three subcells are needed, one of which is to be p_c. For three subcells, nine different groupings can be had, these are shown in Figure 3.8 as the nine different stencils used to interpolate to the subinterface. For each stencil, the interpolating coefficients are determined; this results in each stencil being able to calculate w, $\Delta x \frac{\partial w}{\partial x}$, and $\Delta y \frac{\partial w}{\partial y}$ at the subinterface based upon the values of w at the centroids of the subcells

$$w^k = \sum_s a_s^k w_s^k, \tag{3.53}$$

$$\Delta x \frac{\partial w}{\partial x}^k = \sum_s b_s^k w_s^k, \tag{3.54}$$

$$\Delta y \frac{\partial w}{\partial y}^k = \sum_s d_s^k w_s^k, \tag{3.55}$$

$$\tag{3.56}$$

where

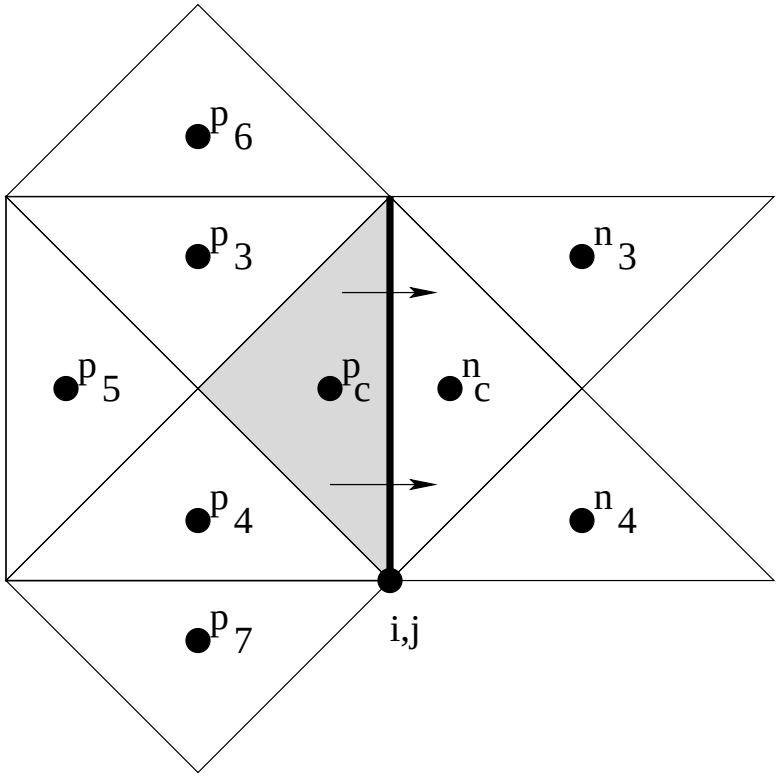
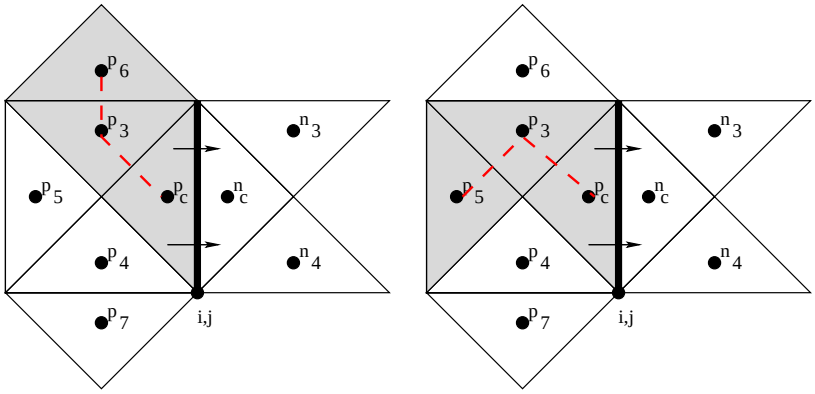
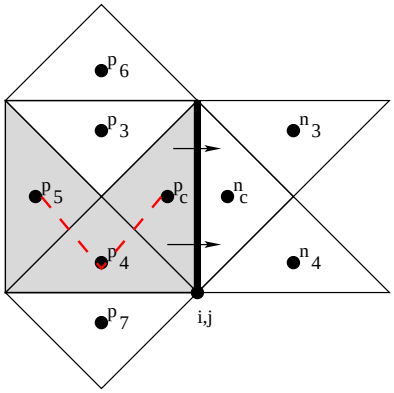


Figure 3.7: This shows the positive footprint used for the interpolating to subinterface 0 for the triangular WENO scheme. The heavy dark vertical line in the middle shows the subinterface of interest. The dark spots show the centroid of each cell accompanied by a designation that is relevant to this footprint; with p and n being used to exploit symmetry to implement the interpolation of the negative footprint. The shaded subcell represents the subcell immediately upwind of the subinterface of interest. This diagram shows two more subcells than in Figure 3.11a, this is because p_6 and p_7 will later be deemed moot for the interpolation, a result already incorporated into the later tables and diagrams.

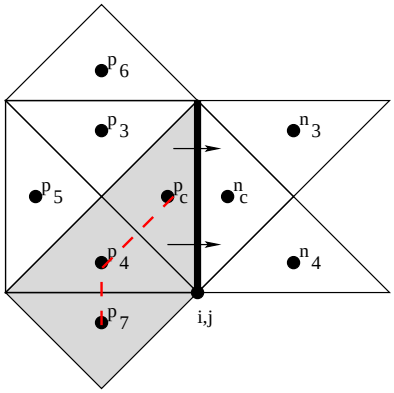


Stencil 1

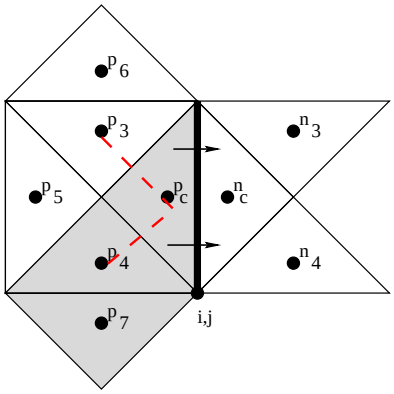
Stencil 2



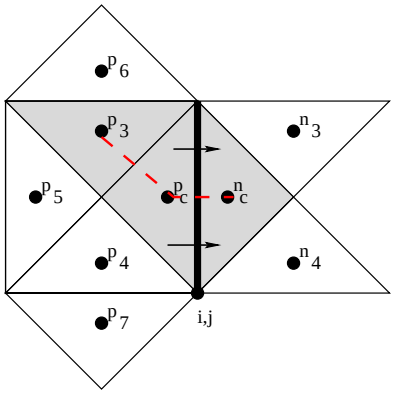
Stencil 3



Stencil 4

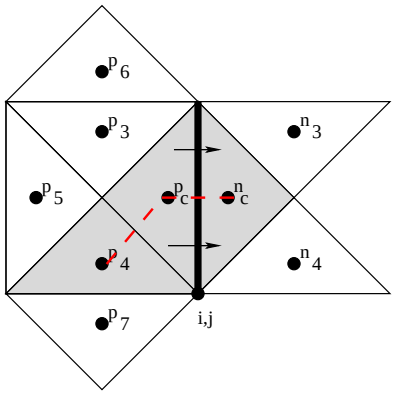


Stencil 5

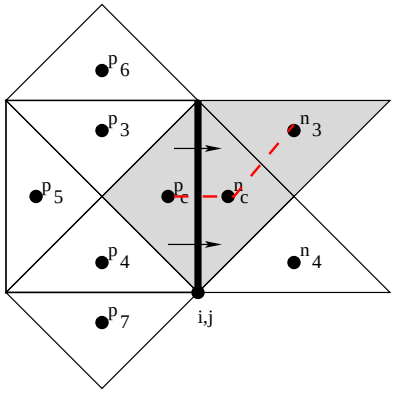


Stencil 6

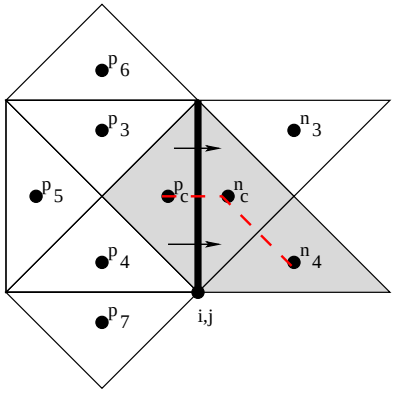
Figure 3.8: These figures show stencils 1-6 for the positve footprint of subinterface 0. Each stencil is denoted by the shaded subcells and the dotted line connecting all three subcells.



Stencil 7



Stencil 8



Stencil 9

Figure 3.9: This is a continuation of Figure 3.8, showing stencils 7-9.

w^k is the estimate provided by this stencil,

$\Delta x \frac{\partial w}{\partial x}^k$ is the estimate to the slope of w at the subinterface based upon this stencil,

$\Delta y \frac{\partial w}{\partial y}^k$ is the estimate to the slope of w at the subinterface based upon this stencil,

k is the stencil k,

a_s^k is interpolating coefficient for stencil k and subcell s,

b_s^k is interpolating coefficient for stencil k and subcell s to compute the slope term,

d_s^k is interpolating coefficient for stencil k and subcell s to compute the slope term,

s is the subcell which will iterate to include each subcell in the stencil.

A second interpolation is performed using *all* of the cells in the positive footprint, which would provide a higher order interpolation of w at the subinterface such that

$$w^* = \sum_s a_s^* w_s \tag{3.57}$$

where

w^* is the higher order interpolation based upon all of the subcells,

a_s^* is the coefficient for the s subcell,

s is the subcell and will range across all of the subcells in the positive footprint.

With the higher order interpolation, a third set of coefficients are computed, which, yields the following property

$$a_s^* = \sum^k \mathcal{C}_k a_s^k \tag{3.58}$$

where

k denotes a stencil,

s denotes a subcell,

a_s^k is the interpolating coefficient from the stencil k if subcell s lies in that stencil, otherwise this is zero,

$\mathcal{C}_k$ is better known as the optimal weight for the stencil.

Solving Equation 3.58 requires care, as the matrix of a_s^k is normally rank deficient allowing for some flexibility in the solution. For the positive footprint for subinterface 0, three stencils have optimal weights of zero; stencils 1, 4, and 5; which means these stencils can be dropped from the calculations – a side effect being that subcells p_6 and p_7 do not figure into the calculations and are therefore dropped from the footprint.

The negative footprint exploits the symmetry around subinterface 0 by switching the n-subcells and the p-subcells with each other; additional subcells are defined for n_5 to n_7 where p_5 to p_7 were.

For the other five subinterfaces, a similar analysis is needed. However, by exploiting symmetry again, the needed number of type of footprints is reduced to two, the on-axis or Axial footprint, and the diagonal footprint. Figure 3.11 shows the two footprints used by the subinterfaces, each one being rotated and reflected until the direction of the assumed flow lines up with that of the subinterface. Subinterface 0 uses the axial footprint as is. Subinterface 1 uses the axial footprint after rotating it by 90° clockwise. Subinterface 2 uses the diagonal footprint after rotating it by 45° clockwise. Subinterface 3 uses the diagonal footprint after rotating it by 45° counterclockwise then reflecting it across the constant $i + \frac{1}{2}$ line. Subinterface 4 uses the diagonal footprint after rotating it by 45° clockwise then reflecting it across the constant $i + \frac{1}{2}$ line Subinterface 5 uses the diagonal footprint after rotating it by 45° counterclockwise. For a given subinterface, Table 3.1 indicates how the subcells are fitted to the footprint used, after the footprint has been rotated and reflected.

Table 3.2 lists the assignments to stencils, the coefficients, and optimal weights used for the On-Axis stencil. As noted earlier, three stencils had optimal weights of zero; this permits stencils 1, 4, and 5 to be ignored which renders four subcells as not needed.

Table 3.3 lists the assignment to stencils, the coefficients, nd optimal weights used for the diagonal stencil. Two stencils, 6 and 7, are omitted from this table since their optimal

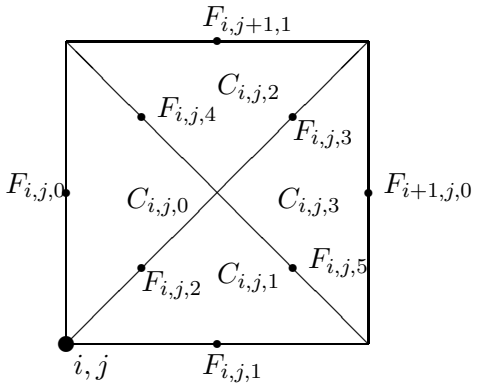
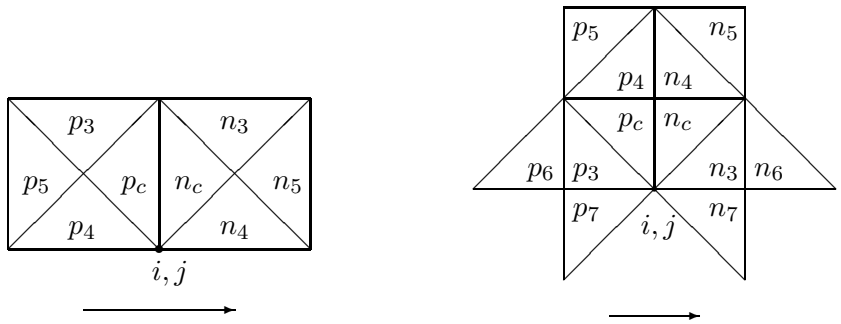


Figure 3.10: This shows how the i, j cell is divided into subcells and the subinterfaces between the subcells. C denote the subcells, there are four of these per grid cell. F denote the subinterfaces, there are six of these per grid cell. The dots in the middle of the subinterfaces denote the center of each subinterface.
This figure is a repeat of Figure 3.2.



(a) On-Axis (b) Diagonal

Figure 3.11: Each subinterface is influenced by a group of surrounding subcells. The pattern of these subcells is the footprint, which contains every subcell needed to calculate the flux at a particular subinterface of the i, j grid cell shown in Figure 3.10. The thick line, between subcells p_c and n_c, represents the subinterface of interest.
This figure shows the footprints used for the triangular WENO flux scheme to calculate the flux at each subinterface. (a) is the On-Axis footprint. (b) is the Diagonal footprint.
Symmetries are used to make these footprints fit to each of the subinterfaces; each footprint is transformed so that an arrow from p_c to n_c lines up with the arrow shown in Figure 3.6. Subinterface 0 uses the On-Axis footprint as shown. Subinterface 1 uses the On-Axis footprint rotated 90°. Subinterface 2 uses the Diagonal footprint rotated 45°. Subinterfaces 3, 4, and 5 use the Diagonal footprint with appropriate rotation and reflection.
Subcells are labeled for convenience in applying the WENO interpolating function; p and n depend on which side of the sub-interface the sub-cell is on, the number is a designation, the c subscript indicates that the sub-cell is adjacent to the sub-interface.

Table 3.1: This table lists the assignments of subcells to the appropriate position into the footprint; these assignment account for the needed rotation to bring the footprint positive flux flow in parallel to the x-axis. Subface 0 and 1 map onto the Axial footprint, which has four blanks in the table since they do not make use of those cells in the WENO interpolation. Subface 2 thru 5 map onto the Diagonal footprint.
Further explanation is in the text.

	subface 0	subface 1	subface 2	subface 3	subface 4	subface 5,
p_c	$i-1,j,3$	$i,j-1,2$	$i,j,0$	$i,j,2$	$i,j,0$	$i,j,1$
p_3	$i-1,j,2$	$i,j-1,0$	$i-1,j,3$	$i,j+1,1$	$i-1,j,3$	$i,j-1,2$
p_4	$i-1,j,1$	$i,j-1,3$	$i,j,2$	$i,j,0$	$i,j,1$	$i,j,0$
p_5	$i-1,j,0$	$i,j-1,1$	$i,j+1,1$	$i-1,j,3$	$i,j-1,2$	$i-1,j,3$
p_6			$i-1,j,2$	$i,j+1,0$	$i-1,j,1$	$i,j-1,0$
p_7			$i-1,j,1$	$i,j+1,3$	$i-1,j,2$	$i,j-1,3$
n_c	$i,j,0$	$i,j,1$	$i,j,1$	$i,j,3$	$i,j,2$	$i,j,3$
n_3	$i,j,2$	$i,j,3$	$i,j-1,2$	$i+1,j,0$	$i,j+1,1$	$i+1,j,0$
n_4	$i,j,1$	$i,j,0$	$i,j,3$	$i,j,1$	$i,j,3$	$i,j,2$
n_5	$i,j,3$	$i,j,2$	$i+1,j,0$	$i,j-1,1$	$i+1,j,0$	$i,j+1,1$
n_6			$i,j-1,3$	$i+1,j,1$	$i,j+1,3$	$i+1,j,2$
n_7			$i,j-1,0$	$i+1,j,2$	$i,j+1,0$	$i+1,j,1$

Table 3.2: This table lists all the coefficients needed for using the Axial footprint. Stencils 1, 4, and 5 are omitted since their optimal weights are zero; therefore, these stencils are eliminated. The second set of columns selects the cells used for the stencil, which will always implicitly include p_c, therefore, stencil 2 uses cell p_c, p_3, and p_5. The third set of columns indicate the coefficients used for the actual interpolation to the sub-interface. The fourth and fifth set of columns indicate the coefficients needed to interpolate the derivatives to the sub-interface based upon the values at the cell centers.
Further explanation is in the text.

stencil	c_2	c_3	ρ_1	ρ_2	ρ_3	r_1	r_2	r_3	q_1	q_2	q_3	ω
2	p_3	p_5	$\frac{5}{4}$	0	$\frac{1}{4}$	$\frac{3}{2}$	0	$-\frac{3}{2}$	$-\frac{3}{2}$	3	$-\frac{3}{2}$	$\frac{1}{12}$
3	p_4	p_5	$\frac{5}{4}$	0	$\frac{1}{4}$	$\frac{3}{2}$	0	$-\frac{3}{2}$	$\frac{3}{2}$	-3	$\frac{3}{2}$	$\frac{1}{12}$
6	n_c	p_3	$\frac{1}{2}$	$\frac{1}{2}$	0	-3	3	0	-6	3	3	$\frac{5}{24}$
7	n_c	p_4	$\frac{1}{2}$	$\frac{1}{2}$	0	-3	3	0	6	-3	-3	$\frac{5}{24}$
8	n_c	n_3	$\frac{1}{2}$	$\frac{1}{2}$	0	-3	3	0	3	-6	3	$\frac{5}{24}$
9	n_c	n_4	$\frac{1}{2}$	$\frac{1}{2}$	0	-3	3	0	-3	6	-3	$\frac{5}{24}$

Table 3.3: This table lists all the coefficients needed for using the Diagonal footprint. Stencils 6 and 7 are omitted since their optimal weight are zero; therefore, these stencils are eliminated. The second set of columns selects the cells used for the stencil, which will always implicitly include p_c, therefore, stencil 2 uses cell p_c, p_4, and n_4. The third set of columns indicate the coefficients used for the actual interpolation to the sub-interface. The fourth and fifth set of columns indicate the coefficients needed to interpolate the derivatives to the sub-interface based upon the values at the cell centers.
Further explanation is in the text.

stencil	c_2	c_3	ρ_1	ρ_2	ρ_3	r_1	r_2	r_3	q_1	q_2	q_3	ω
1	p_4	p_5	$\frac{3}{4}$	$\frac{5}{4}$	-1	$-\frac{3}{2}$	$\frac{9}{2}$	-3	$-\frac{3}{2}$	$\frac{3}{2}$	0	$-\frac{95}{96}$
2	p_4	n_4	$\frac{5}{4}$	$-\frac{3}{4}$	$\frac{1}{2}$	0	$-\frac{3}{2}$	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{3}{2}$	0	$\frac{385}{288}$
3	p_3	p_6	$\frac{1}{2}$	$\frac{5}{4}$	$-\frac{3}{4}$	0	$\frac{3}{2}$	$-\frac{3}{2}$	3	$-\frac{0}{2}$	$\frac{3}{2}$	$-\frac{1}{4}$
4	p_3	p_7	2	$-\frac{7}{4}$	$\frac{3}{4}$	3	$-\frac{9}{2}$	$\frac{3}{2}$	0	$\frac{3}{2}$	$-\frac{3}{2}$	$\frac{1}{2}$
5	p_3	p_4	$\frac{11}{4}$	-1	$-\frac{3}{4}$	$\frac{9}{2}$	-3	$-\frac{3}{2}$	$-\frac{3}{2}$	0	$\frac{3}{2}$	$\frac{7}{4}$
8	n_c	n_3	$\frac{3}{4}$	$-\frac{1}{4}$	$\frac{1}{2}$	$-\frac{3}{2}$	$\frac{3}{2}$	0	$-\frac{3}{2}$	$\frac{9}{2}$	-3	$\frac{29}{48}$
9	n_c	n_4	$\frac{1}{2}$	$\frac{3}{4}$	$-\frac{1}{4}$	$-\frac{3}{2}$	$\frac{3}{2}$	0	0	$-\frac{3}{2}$	$\frac{3}{2}$	$-\frac{281}{144}$

weights are zero. As shown, some of the optimal weights are negative, which is unstable according to Shu[12]. Therefore, for the diagonal footprint, the optimal weights are ignored, replaced with singular values of 1.0.

With these coefficients, the triangular WENO follows the same steps as the one dimensional WENO.

3.3.2 Time Advance

The hyperbolic solver advances in time in accordance to

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \sum_{\alpha} S_{\alpha} F \cdot \boldsymbol{n}_{\alpha} = 0. \tag{3.59}$$

For each time scheme, the summation is needed and is therefore assigned its own operator

$$\mathcal{L}(\boldsymbol{Q}) \equiv -\sum_{\alpha} S_{\alpha} F \cdot \boldsymbol{n}_{\alpha}. \tag{3.60}$$

On a rectangular grid,

$$\mathcal{L}(\boldsymbol{Q})_{i,j} \equiv \frac{1}{V_{i,j}} \left(S_{I_{i,j}} \boldsymbol{F}_{I_{i,j}} - S_{I_{i+1,j}} \boldsymbol{F}_{I_{i+1,j}} + S_{J_{i,j}} \boldsymbol{F}_{J_{i,j}} - S_{J_{i,j+1}} \boldsymbol{F}_{J_{i,j+1}} \right). \tag{3.61}$$

On a triangular grid,

$$\mathcal{L}(\boldsymbol{Q})_{i,j,0} = \frac{1}{V_{i,j,0}} \left(S_{i,j,0} \boldsymbol{F}_{i,j,0} - S_{i,j,2} \boldsymbol{F}_{i,j,2} - S_{i,j,4} \boldsymbol{F}_{i,j,4} \right), \tag{3.62}$$

$$\mathcal{L}(\boldsymbol{Q})_{i,j,1} = \frac{1}{V_{i,j,1}} \left(S_{i,j,1} \boldsymbol{F}_{i,j,1} + S_{i,j,2} \boldsymbol{F}_{i,j,2} - S_{i,j,5} \boldsymbol{F}_{i,j,5} \right), \tag{3.63}$$

$$\mathcal{L}(\boldsymbol{Q})_{i,j,2} = \frac{1}{V_{i,j,2}} \left(S_{i,j,4} \boldsymbol{F}_{i,j,4} - S_{i,j+1,1} \boldsymbol{F}_{i,j+1,1} - S_{i,j,3} \boldsymbol{F}_{i,j,3} \right), \tag{3.64}$$

$$\mathcal{L}(\boldsymbol{Q})_{i,j,3} = \frac{1}{V_{i,j,3}} \left(S_{i,j,3} \boldsymbol{F}_{i,j,3} + S_{i,j,5} \boldsymbol{F}_{i,j,5} - S_{i+1,j,0} \boldsymbol{F}_{i+1,j,0} \right). \tag{3.65}$$

First Order Euler Time

For first order Euler time, the derivative in Equation (3.59) is expanded to first order

$$\frac{\boldsymbol{Q}^{n+1} - \boldsymbol{Q}^{n}}{\Delta t} \approx \mathcal{L}(\boldsymbol{Q}^{n}) \tag{3.66}$$

so the update formula is

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^{n} + \Delta t \mathcal{L}(\boldsymbol{Q}^{n}). \tag{3.67}$$

Third Order Runge Kutta

The third-order total variation diminishing (TVD) Runge-Kutta as applied by Shu[11] and in this solver is

$$\begin{aligned} \boldsymbol{Q}^{(1)} &= \boldsymbol{Q}^n + \Delta t\, \mathcal{L}\left(\boldsymbol{Q}^n\right), \\ \boldsymbol{Q}^{(2)} &= \frac{3}{4}\boldsymbol{Q}^n + \frac{1}{4}\boldsymbol{Q}^{(1)} + \frac{1}{4}\Delta t\, \mathcal{L}\left(\boldsymbol{Q}^{(1)}\right), \\ \boldsymbol{Q}^{n+1} &= \frac{1}{3}\boldsymbol{Q}^n + \frac{2}{3}\boldsymbol{Q}^{(2)} + \frac{2}{3}\Delta t\, \mathcal{L}\left(\boldsymbol{Q}^{(2)}\right). \end{aligned} \tag{3.68}$$

This scheme is TVD because the fractions are positive. Third order accuracy is shown by evaluating $\boldsymbol{Q}^{n+1}$ in terms of $\boldsymbol{Q}^n$, by letting $\mathcal{L} = \frac{\partial \boldsymbol{Q}}{\partial t}$.

$$\begin{aligned} \boldsymbol{Q}^{(1)} &= \boldsymbol{Q}^n + \Delta t \frac{\partial \boldsymbol{Q}}{\partial t}, \\ \boldsymbol{Q}^{(2)} &= \frac{3}{4}\boldsymbol{Q}^n + \frac{1}{4}\left(\boldsymbol{Q}^n + \Delta t \frac{\partial \boldsymbol{Q}}{\partial t}\right) + \frac{1}{4}\Delta t \left.\frac{\partial \boldsymbol{Q}}{\partial t}\right|_{\boldsymbol{Q}^n + \Delta t \frac{\partial \boldsymbol{Q}}{\partial t}}, \\ &= \boldsymbol{Q}^n + \frac{1}{4}\Delta t\left[\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{Q}}{\partial t} + \Delta t\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^2}{2}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^3}{6}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \frac{(\Delta t)^4}{24}\frac{\partial^5 \boldsymbol{Q}}{\partial t^5}\right], \\ &= \boldsymbol{Q}^n + \frac{1}{4}\Delta t\left[2\frac{\partial \boldsymbol{Q}}{\partial t} + \Delta t\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^2}{2}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^3}{6}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \frac{(\Delta t)^4}{24}\frac{\partial^5 \boldsymbol{Q}}{\partial t^5}\right], \\ \boldsymbol{Q}^{n+1} &= \frac{1}{3}\boldsymbol{Q}^n + \frac{2}{3}\boldsymbol{Q}^{(2)} + \frac{2}{3}\Delta t\, \mathcal{L}\left(\boldsymbol{Q}^{(2)}\right), \\ &= \frac{1}{3}\boldsymbol{Q}^n + \frac{2}{3}\boldsymbol{Q}^n + \frac{2}{3}\cdot\frac{1}{4}\Delta t\left[2\frac{\partial \boldsymbol{Q}}{\partial t} + \Delta t\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^2}{2}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^3}{6}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \frac{(\Delta t)^4}{24}\frac{\partial^5 \boldsymbol{Q}}{\partial t^5}\right] \\ &\quad + \frac{2}{3}\Delta t \left.\frac{\partial \boldsymbol{Q}}{\partial t}\right|_{\boldsymbol{Q}^{(2)}}, \\ &= \frac{1}{3}\boldsymbol{Q}^n + \frac{2}{3}\boldsymbol{Q}^n + \frac{1}{6}\Delta t\left[2\frac{\partial \boldsymbol{Q}}{\partial t} + \Delta t\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^2}{2}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^3}{6}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \frac{(\Delta t)^4}{24}\frac{\partial^5 \boldsymbol{Q}}{\partial t^5}\right] \\ &\quad + \frac{2}{3}\Delta t\left[\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{1}{4}\left(\Delta t\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^2}{2}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^3}{6}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \frac{(\Delta t)^4}{24}\frac{\partial^5 \boldsymbol{Q}}{\partial t^5}\right)\right], \\ &= \boldsymbol{Q}^n + \Delta t\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^3}{6}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^4}{18}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \mathcal{O}\left((\Delta t)^5\right). \end{aligned}$$

Compare this result with a Taylors series expansion about $\boldsymbol{Q}^n$ which is

$$\boldsymbol{Q}^{n+1}_{\text{Taylor}} = \boldsymbol{Q}^n + \Delta t\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 \boldsymbol{Q}}{\partial t^2} + \frac{(\Delta t)^3}{6}\frac{\partial^3 \boldsymbol{Q}}{\partial t^3} + \frac{(\Delta t)^4}{24}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \mathcal{O}\left((\Delta t)^5\right). \tag{3.69}$$

Therefore, the Runge Kutta method agrees with the Taylors series to the third time derivative with the error term being

$$\boldsymbol{Q}^{n+1}_{\text{Error}} = \boldsymbol{Q}^{n+1}_{\text{Taylor}} - \boldsymbol{Q}^{n+1} = -\frac{(\Delta t)^4}{72}\frac{\partial^4 \boldsymbol{Q}}{\partial t^4} + \mathcal{O}\left((\Delta t)^5\right). \tag{3.70}$$

Which means this time integration scheme is third order accurate.

3.4 Source Terms

Source terms are decomposed into the two contributing terms, then the composite time advance formulas will be given.

3.4.1 Currents

The current is assumed to be independent of $\boldsymbol{Q}$, therefore, the governing equation is

$$\frac{\partial \boldsymbol{Q}}{\partial t} = -\boldsymbol{\mathcal{J}} \tag{3.71}$$

and the trapezoidal rule is applied by averaging $\boldsymbol{\mathcal{J}}$ at n and $n+1$. So, if the current were the sole source term, the time advance would be

$$\boldsymbol{Q}^{n+1} = \boldsymbol{Q}^n - \frac{1}{2}\Delta t\left(\boldsymbol{\mathcal{J}}^{n+1} + \boldsymbol{\mathcal{J}}^n\right). \tag{3.72}$$

3.4.2 Decay

The decay is used to add artificial damping to the solution, usually at the edges of the computational domain for absorbing boundary conditions; this is described later. The damping obeys the relation

$$\frac{\partial \boldsymbol{Q}}{\partial t} = -\sigma \boldsymbol{Q} \tag{3.73}$$

where

σ is the decay coefficient, normally zero within the domain, and

$\sigma > 0$ where damping is desired.

An analytic solution is used for this time advance,

$$\boldsymbol{Q}^{n+1} = e^{-\sigma \Delta t}\boldsymbol{Q}^n. \tag{3.74}$$

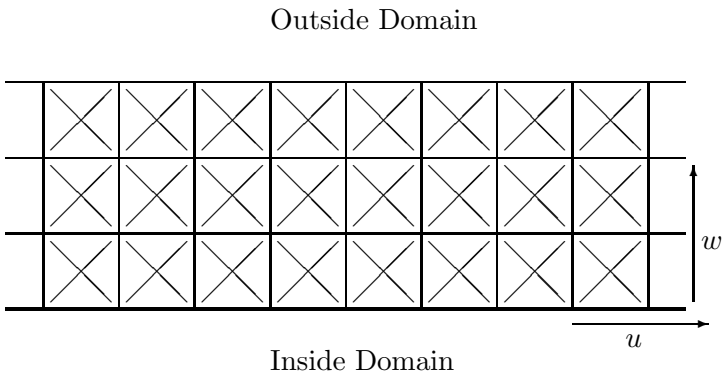


Figure 3.12: This figure depicts the three layers of ghost cells for a boundary, whose coordinates for the boundary are u, w to be mapped into i, J.

3.4.3 Time Advance

The time advance for both the current and decay sources is

$$\boldsymbol{Q}^{n+1} = e^{-\sigma \Delta t} \boldsymbol{Q}^{n} - \frac{1}{2} \Delta t \left(\boldsymbol{\mathcal{J}}^{n+1} + \boldsymbol{\mathcal{J}}^{n} \right). \tag{3.75}$$

3.5 Boundary Conditions

Boundary conditions are implemented by adding a layer of ghost cells to the exterior of the domain and setting the values within them such that they have physical meaning for each boundary type. This section describes how each boundary condition is implemented.

For each boundary, two memory mapping functions are defined in a u, v, w space. (In 2D, this is reduced to u, w.) One of the mapping functions maps outside the domain, into the boundary, with a u, v, w coordinate system as shown in Figure 3.12. The second mapping function maps inside the domain, where appropriate, as defined by the particular boundary condition, such that, relevant cell data can be addressed using the u, v, w coordinate system. The adoption of the u, v, w simplifies the implementation of most of the boundary conditions. The first mapping function is denoted $\mho_{\text{ext}}$. The second mapping function is denoted $\mho_{\text{int}}$.

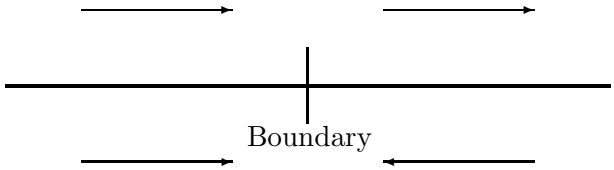


Figure 3.13: This shows the effective manipulation by the reflecting boundary condition. The top two arrows show the initial wave, while the bottom shows the waves after reflection.

3.5.1 Reflecting Boundary Condition

Reflecting boundary conditions mirror any outgoing wave component into an incoming wave component as illustrated in Figure 3.13. The inside mapping function is mapped to the interior edge of the given boundary, such that, cell u, v, w of $\mho_{\text{ext}}$ is the same distance from the boundary as cell u, v, w of $\mho_{\text{int}}$.

3.5.2 Absorbing Boundary Condition

Absorbing boundary conditions have been an area of active research[13], with common approaches being field extrapolation, impedance conditions, surface integrals, absorbing material, and one-way wave equations. *Field extrapolation* is where the fields are extrapolated from some point in a cylindrical or spherical fashion (two or three dimensions); this approach can be reasonably successful but it forces the user to pick an origin for expansion which might not be suitable for some problems. *Impedance conditions* is where the electromagnetic waves are assumed to be normally propagating plane waves therefore the fields in the boundary are set such that no incoming waves are present; however, this only works well when the outgoing waves impinge the boundary at near-normal angles, otherwise, the tangential components of the waves remain and will result in noticeable errors. *Surface Integrals* is where equivalence principles are used in rigorous surface integrals to create virtual currents that are then used to extrapolate the fields; while this can be highly effective, the

global nature of the calculations can be computationally intractable. *Absorbing material* is where layers of cells are added which have damping terms designed to decay the fields by the time the waves reach the boundary; this has the drawback that extra cells must be added which increase the computational burden. *One-way wave equations* is where the solver is modified to accommodate equations that only permit the waves to propagate outward; this adds complexity to the solver.

In this thesis, two of the above methods are employed, impedance conditions and the absorbing material. For the boundary, the $\mho_{\text{int}}$ is set to the immediate interior of the domain, then the fields in the interior are decomposed into the characteristic waves based on the eigenvectors of the Flux Jacobian; outbound waves are left.

3.5.3 Neighbor Boundary Condition

The neighbor boundary condition, which will be applied automatically between two neighboring blocks unless the end user specifies otherwise, is used to implement continuity between the two adjacent faces of two adjacent blocks. To implement, the $\mho_{\text{int}}$ is set to the immediate interior of the neighboring block, the data is then copied over into the boundary. If the two blocks are on different processors, a message passing interface is used.

3.5.4 Periodic Boundary Condition

Specifying periodic boundary conditions is a statement to the repeated nature of the problem being solved. The periodic boundary condition is implemented by setting $\mho_{\text{int}}$ to the interior on the other side of the computational domain, then cycling through u, v, w copying data from $\mho_{\text{int}}$ into the $\mho_{\text{ext}}$ of the boundary.

3.5.5 Copy Boundary Condition

The copy boundary condition is useful for simulating a floating boundary condition, it propagates the outermost cells of the computational domain out into the ghost cells of the boundary. $\mho_{\text{int}}$ is set to the interior of the computational domain.

Chapter 4

TESTING THE SOLVER

The solvers are implemented in C/C++ code; these simulations are run on a AthlonXP running Linux with 512MB of memory.

Two simulations are chosen to test the capabilities needed of the solver, to handle wave propagation and to handle currents; these simulations are the plane wave to test wave propagation and the current to the test the ability to handle a current source.

Three combinations of solver algorithms are tested. *Rectangular Upwind* (RU) is a solver using a rectangular grid, the first order upwind flux scheme, and the first order Euler time advance for the hyperbolic terms. *Triangular Upwind* (TU) is a solver using a triangular grid, the first order upwind flux scheme as applied to the sub-interfaces, and the first order Euler time advance for the hyperbolic terms. *Triangular WENO* (TW) is a solver using a triangular grid, the WENO flux scheme, and the third order TVD Runge Kutta time advance for the hyperbolic terms.

For all simulations, $\mu = \epsilon = v = 1$.

This chapter presents the results, while the following chapter discusses and explains the results

4.1 Plane Wave Propagation

The plane wave propagation simulation is used to show that the solvers can resolve a simple plane wave and it is used to measure the accuracy of the solvers.

A plane wave of the form

$$\xi(x,t) = \sin(2\pi x + 2\pi t) \tag{4.1}$$

is applied to one of the negative going characteristic waves for

$$D_z(x,t) \;=\; \xi(x,t), \tag{4.2}$$

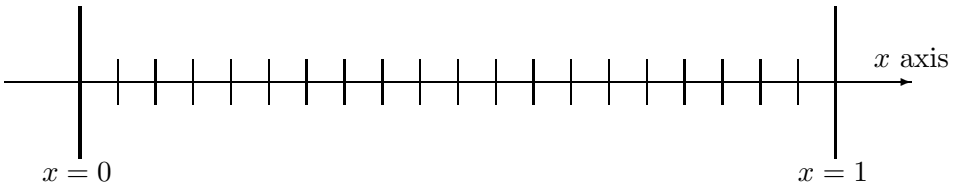


Figure 4.1: This is the computational domain used for the plane wave propagation simulation, which is a periodic domain. Width of the domain is one with periodic boundary conditions on both sides of the domain.

$$H_y(x,t) = -\xi(x,t) \tag{4.3}$$

where these equations are both the initial conditions when $t = 0$ and the analytic solution.

The plane wave is set as the initial solution on a periodic domain (shown in Figure 4.1) and the solver is ran for one light transit time. Except where otherwise noted, the grid is 20 cells in the x-direction and 3 cells in the y-direction. Under these conditions, the wave should propagate, and at one time transit time, the initial solution is the analytic solution. The time step used is a CFL of 0.2 for all three solvers.

The propagation of the wave with the rectangular upwind solver is shown in Figure 4.2 and demonstrates that the wave does propagate properly to the left. As can be seen, significant dissipation in the rectangular upwind solver is evident in the drastic decrease in wave amplitude over the single light transit time.

The propagation of the wave for each of the solvers can be compared in Figure 4.3 where the solution is plotted after one light transit, the solutions should be the initial sine wave $\sin 2\pi x$. As can be seen, the triangular WENO performed the best, however, a bit of super-compression can be seen near the peak. Second best is the triangular upwind which shows itself to be better than the rectangular upwind; however, the triangular upwind solver does have greater fidelity in the usage of four sub-cells per grid cell, therefore, the increase of accuracy over the rectangular upwind solver is not unexpected. In all three cases, the solutions converge to the analytic solution as the grid is refined.

The order of accuracy for each of the solvers is determined by refining the grid, repeating

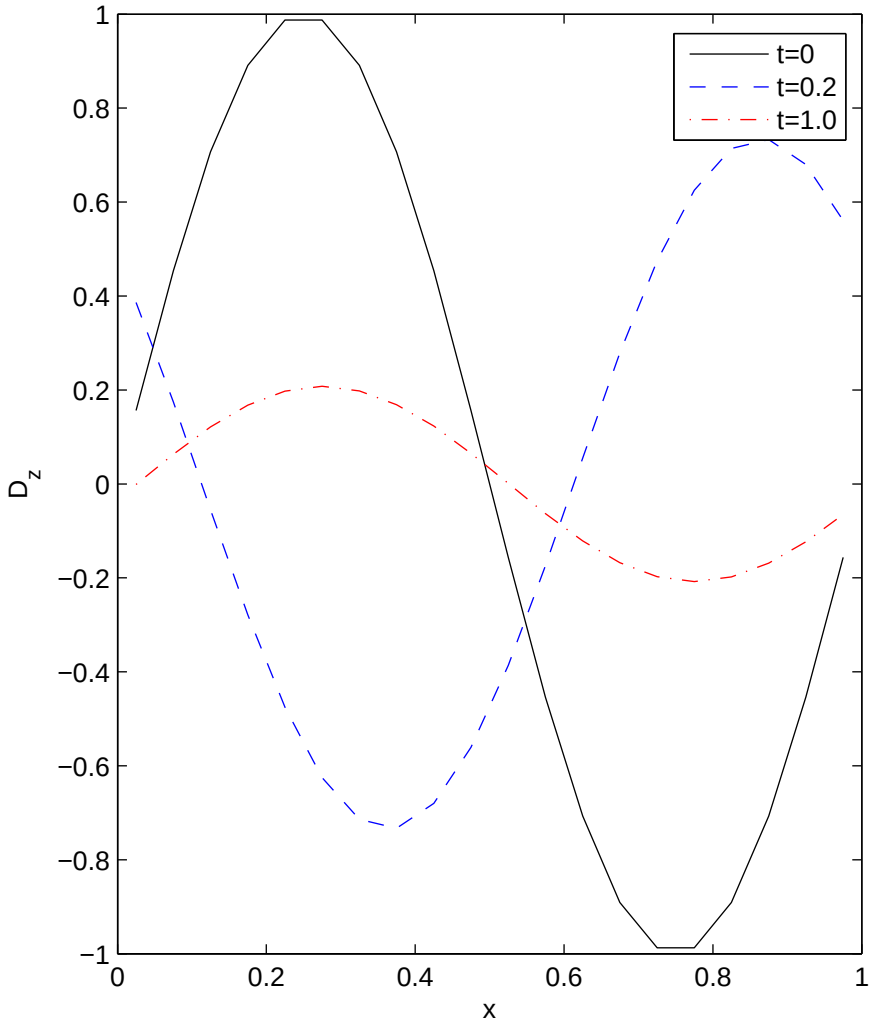


Figure 4.2: This shows the plane wave propagating with the rectangular upwind at two different times; all three lines show that the wave is propagating to the left, which is correct.

Table 4.1: Accuracy of the solvers with a least squares fit, this is the r in $\mathcal{O}(h^r)$. Asymptotic r is computed using $n \geq 750$.

Solver	Overall	Asymptote
Rectangular Upwind	0.7805	0.9867
Triangular Upwind	0.9444	0.9974
Triangular WENO	2.3850	2.0819

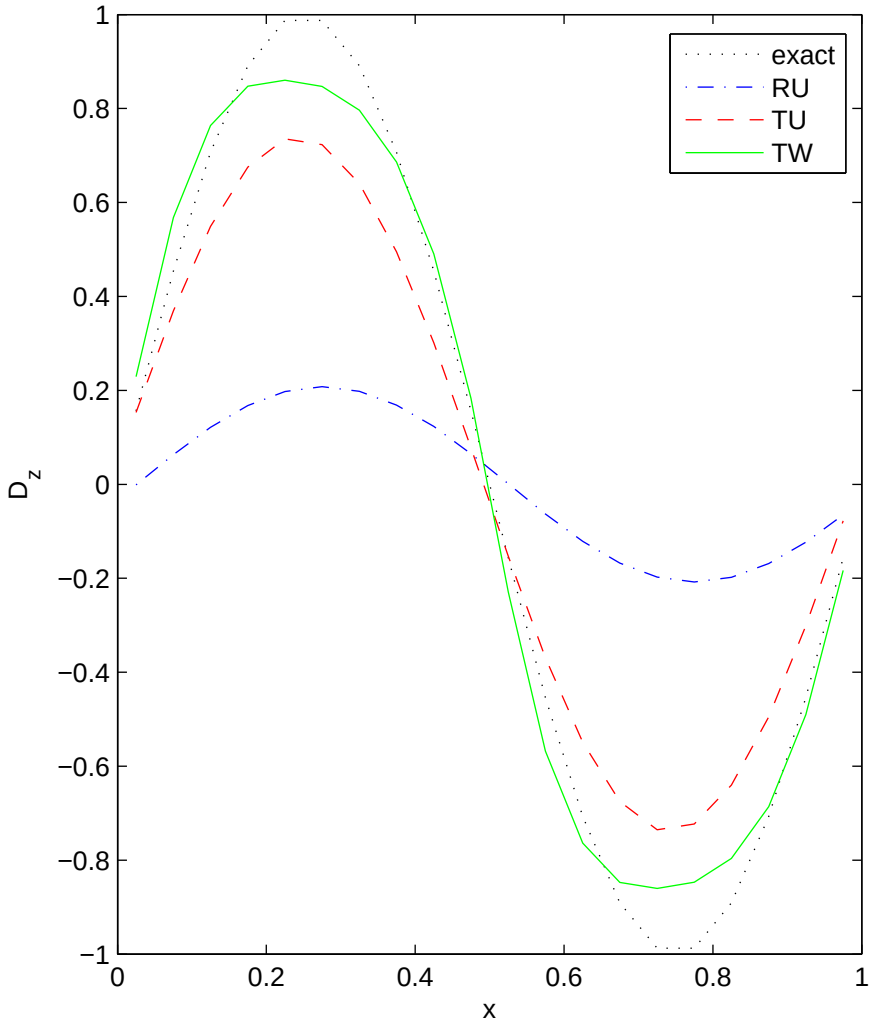


Figure 4.3: This shows the plane wave solution for each of the solvers after one light transit; the analytic solution is shown as the dotted sine wave, with the solutions from the solvers shown by the other lines. From this graph, the triangular WENO solver shows a solution that is the closest to the analytical solution, whereas the triangular upwind and the rectangular upwind show significant decay.

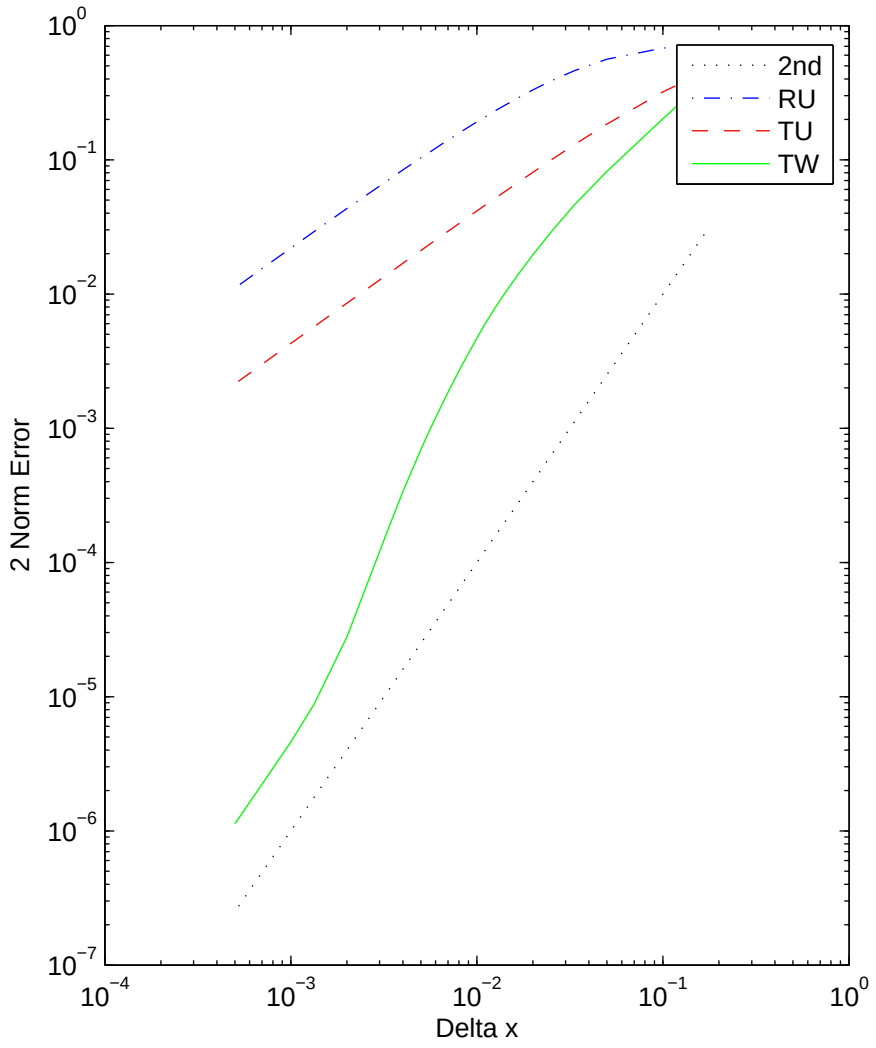


Figure 4.4: This shows the accuracy of solver as the grid is refined. The error is defined as the difference between the solver solution and the analytic solution at one light transit; the 2 norm of this error is plotted on the vertical axis while the Δx of the grid spacing is plotted on the horizontal axis. From the lines, the rectangular upwind solver is the poorest while the triangular WENO solver is the greatest in terms of accuracy.

Table 4.2: Speed Comparison for the solvers to run for 2000x3 cell grid (x4 for triangular upwind and WENO). Listed is the time to completion and the norm-2 error in the solution.

Solver	Time	Error	
		Absolute	Relative
Rectangular Upwind	0:06:15	$1.108 \cdot 10^{-2}$	9,824
Triangular Upwind	0:17:56	$2.140 \cdot 10^{-3}$	1,898
Triangular WENO	4:07:35	$1.128 \cdot 10^{-6}$	1

the simulation, and measuring the error. The error measure ϵ is calculated using the p-norm measure where $p = 1, 2, \infty$, as in

$$\epsilon_{p=1} = \frac{1}{N} \sum_{k}^{\text{cells}} \left| D_{zk} - \sin 2\pi x_k \right|, \tag{4.4}$$

$$\epsilon_{p=2} = \sqrt{\frac{\sum_{k}^{\text{cells}} \left(D_{zk} - \sin 2\pi x_k \right)^2}{N}}, \tag{4.5}$$

$$\epsilon_{p=\infty} = \text{Max}_k \left| D_{zk} - \sin 2\pi x_k \right| \tag{4.6}$$

where

p is the order of the norm,

k is the index to indicate the cell for the rectangular upwind, or the sub-cell for the triangular upwind and WENO solvers,

N is the number of cells for the rectangular upwind, or the number of sub-cells for the triangular upwind and WENO solvers.

The grid is refined in increments from 20 cells in the x direction to 2,000 cells in the x-direction. The measured 2-norm errors for all three solvers is shown in the log-log plot of Figure 4.4; the 1-norm and ∞-norm are similar, therefore, they are not shown. As shown, the triangular WENO provides the least errors for a given resolution.

The 2-norm errors are fitted to the form

$$\epsilon = A \left(\Delta x \right)^r \tag{4.7}$$

where

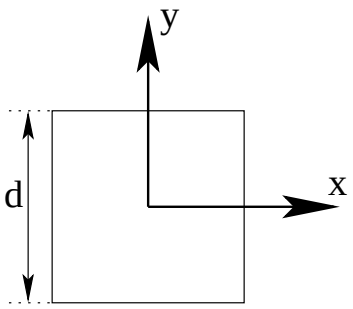


Figure 4.5: This is a cross section of the wire used for the wire simulation; this cross section is square due to the fact that each grid cell is square.

r is the order of accuracy.

This form is fitted to each of the error curves for each of the solvers, the accuracy is listed in Table 4.1. Overall accuracy is determined using the entire dataset from all of the grid refinements. Asymptotic accuracy is determined using the error data from the five finest grid refinements (2,000, 1,500, 1,000, 750, and 500 grid cells).

Speed of each solver is measured by listing the time the main for loop in the code starts the time marching and the time the code stops the time marching to the nearest second. This comparison is done using a grid spanning 2,000 cells in the x-direction. The timings are listed in Table 4.2. While the triangular WENO solver takes the longest to compute, it has the highest accuracy. The rectangular upwind solver takes a mere 6 minutes to run the simulation; however, to reach the same error figures as the triangular WENO solver, a grid of nearly twenty million cells would be required, this would take 11.5 centuries on a single 2.0 GHz AthlonXP.

4.2 Wire Current

The wire current simulation tests the ability to handle a source term in the domain. For each simulation, the computational domain spans $x, y \in (-1, 1)$, the grid is 40 cells in each direction unless otherwise specified. The wire current is assigned a value of $J_z = -100$ on a square wire that measures 0.1 on each side such that the total current is $I_z = 1$; a cross

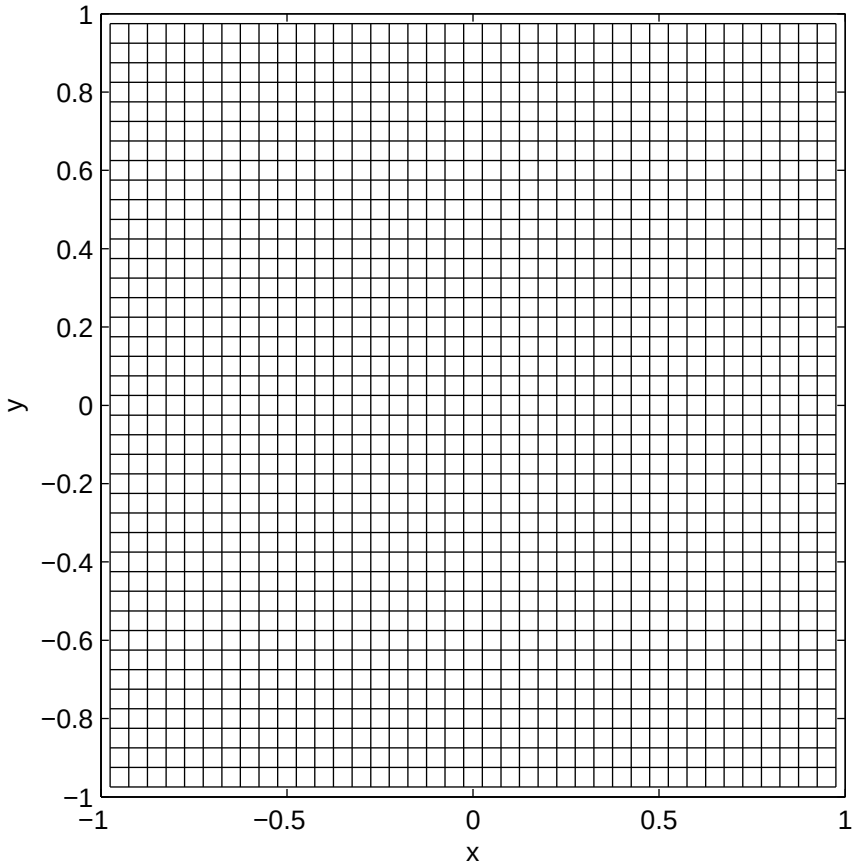


Figure 4.6: This shows the 40x40 computational grid normally used for the simulations. The triangular grid cuts each cell into four sub-cells as described in the previous chapter.

section of this wire is shown in Figure 4.5 which is square since that is the respective shape of each grid cell. The damping source terms is applied outside of a circular region starting at 0.71 distance units from the origin, therefore,

$$\sigma\left(r=\sqrt{x^2+y^2}\right)=\begin{cases}0 & r\leq 0.71\\ 10\,(r-0.71) & r\geq 0.71\end{cases}$$

which is used in conjunction with absorbing boundary conditions. The normal 40x40 grid is shown in Figure 4.6; the other 40x40 grid is shown in Figure 4.7 where the interior cells are rotated by up to 45° in the xy plane to illustrate grid effects.

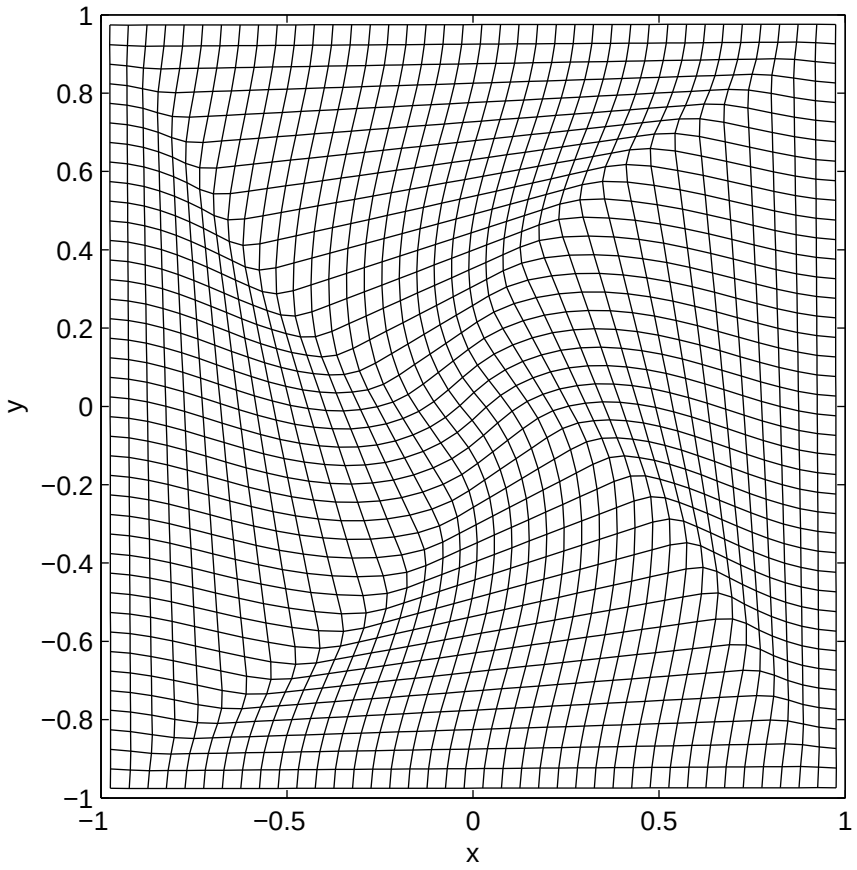


Figure 4.7: This shows the 40x40 computational grid with a 45° applied to the middle of the grid for the simulations that test for grid dependent effects.

4.2.1 Analytic Solution

The analytic solution to the magnetic field of a square wire in an infinite domain is derived and shown in Appendix A; this is also shown in Figure 4.8. As shown there, the solution to the square wire rapidly converges to the solution of the round wire within several wire widths away from the wire. The exact solution shown on the 2D line plots is that of the square wire.

4.2.2 Rectangular Upwind

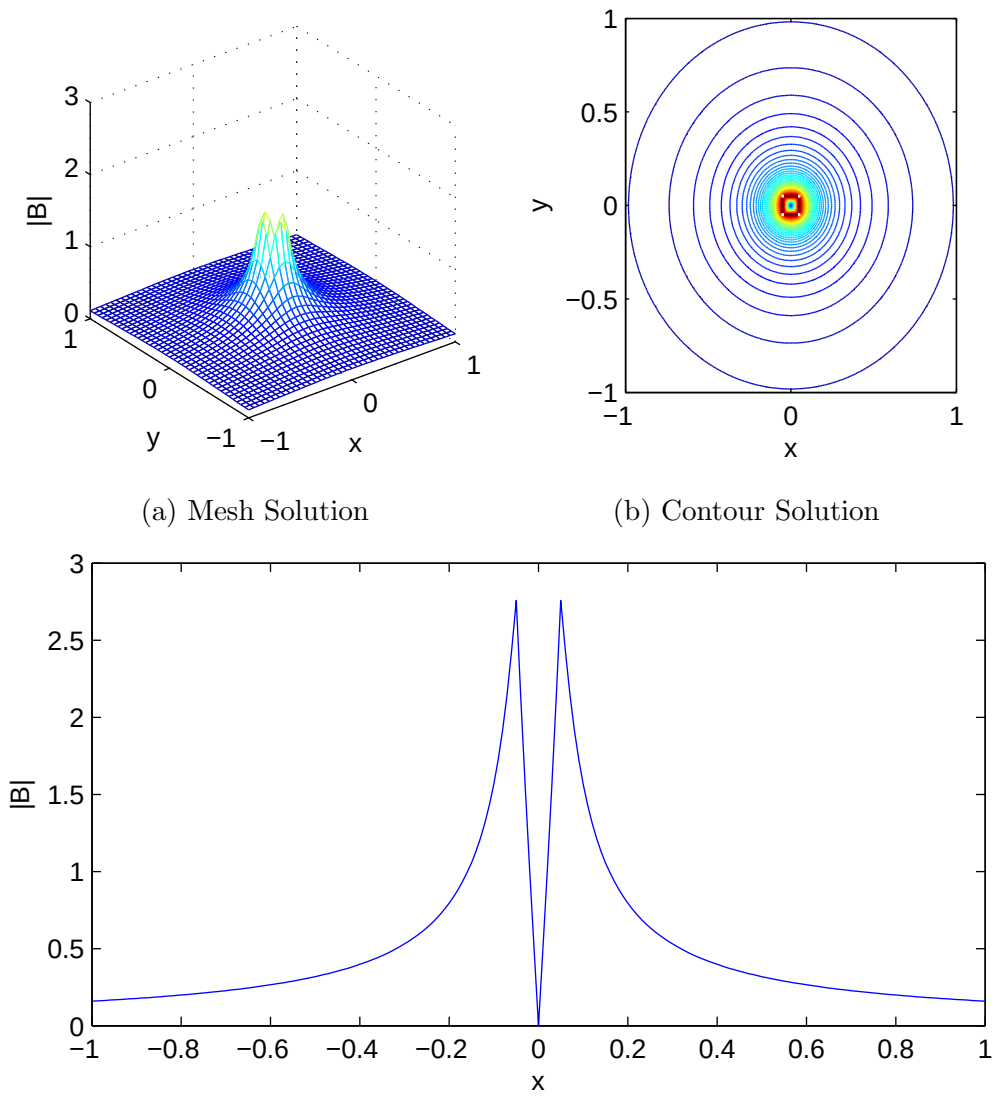
The rectangular upwind solver is used to simulate the wire current. The solution to the magnetic field yielded by the solver is shown in a 3D mesh in Figure 4.9 and is shown in contours of the magnitude in Figure 4.10. As shown, the solution does not reflect the analytic solution; four spurs propagate out from the wire. Each along the spurs, the solution does not reflect the analytic solution as shown in Figure 4.11.

The simulation using the rectangular upwind solver is repeated, this time, twisting the interior cells of the grid by up to 45° as shown in Figure 4.7. The solution to the magnetic field yielded by the solver is shown in a 3D mesh in Figure 4.12 and is shown in contours of the magnitude in Figure 4.13. As shown, the solution does not reflect the analytic solution; the four spurs that propagate out from the wire along the grid lines shows that the error is a grid effect.

4.2.3 Triangular Upwind

The triangular upwind solver is used to simulate the wire current. The solution to the magnetic field yielded by the solver is shown in a 3D mesh in Figure 4.14 and is shown in contours of the magnitude in Figure 4.15. As shown, the solution bears a qualitative resemblance to the analytic solution; this is shown in the axis solution of Figure 4.16, which tracks, within the resolution of the grid, the analytic solution.

The triangular upwind scheme is simulated again, this time using the twisted grid as shown in Figure 4.7. The solution to the magnetic field yielded by the solver is shown in a 3D mesh in Figure 4.17, is shown in contours of the magnitude in Figure 4.18, and in



(a) Mesh Solution

(b) Contour Solution

(c) Cross-Section, X-Axis

Figure 4.8: This is the analytic solution for the steady state magnetic field around a wire current. (a) shows the 3D, this should be axisymmetric in nature. (b) shows the contours of the analytic solution which should be circular around the wire. (c) shows the X-axis cross section.

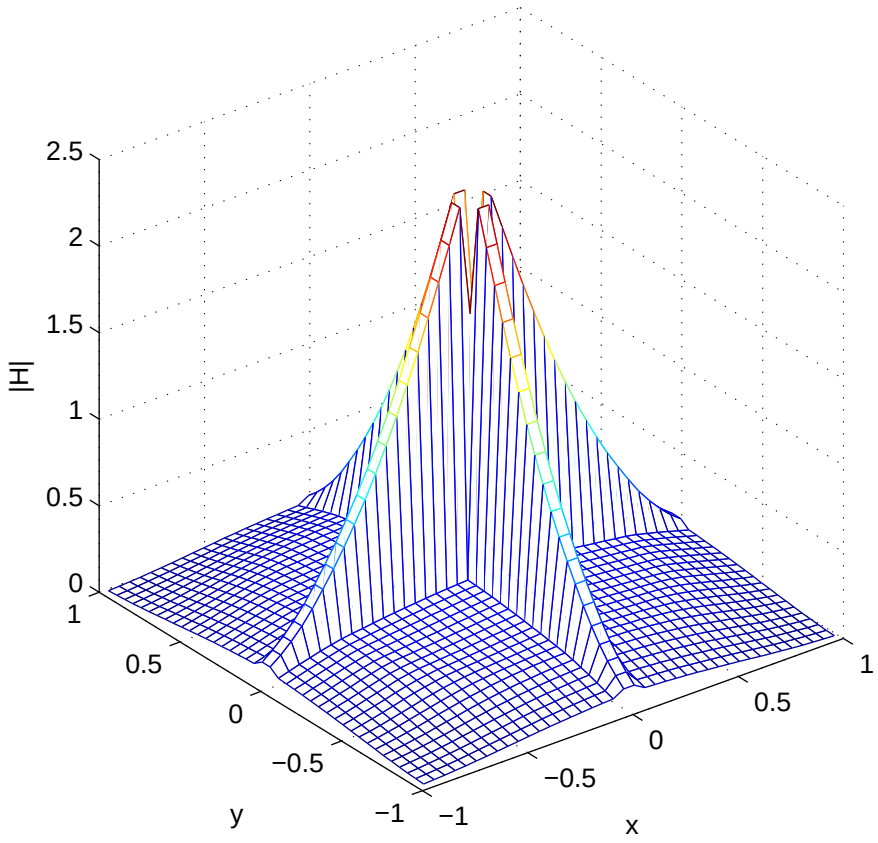


Figure 4.9: This shows the 3-D mesh solution to the wire current using rectangular upwind solver which is shown to fail miserably in this graph.

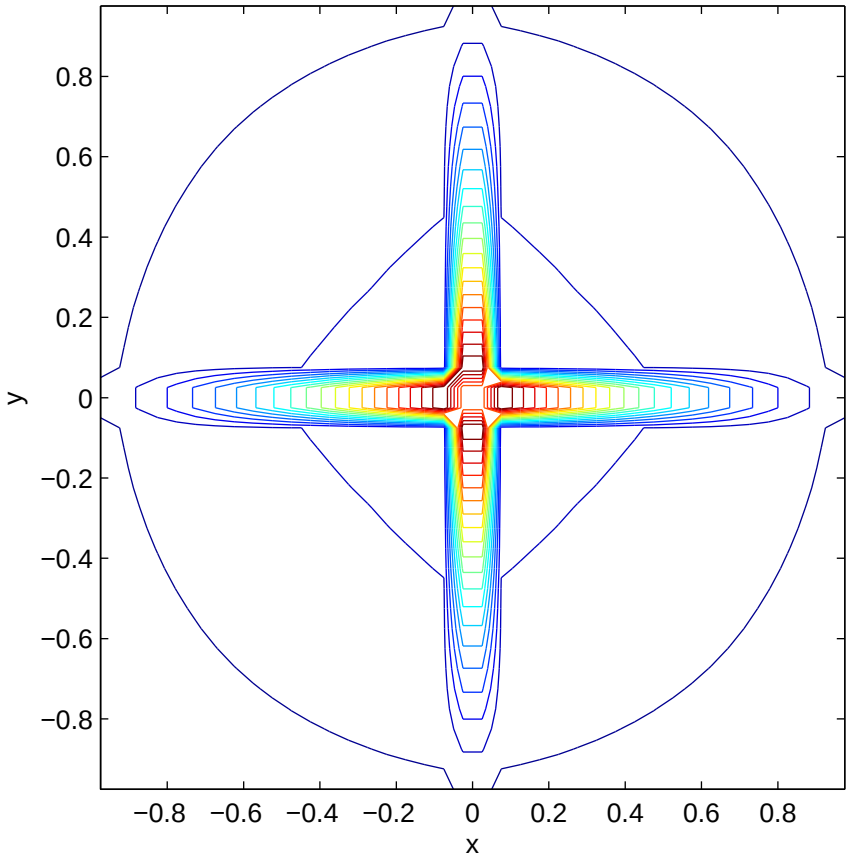


Figure 4.10: This shows the contour of the magnetic field to the wire current using the rectangular upwind solver; as evidenced by the four spurs, this solver fails miserably.

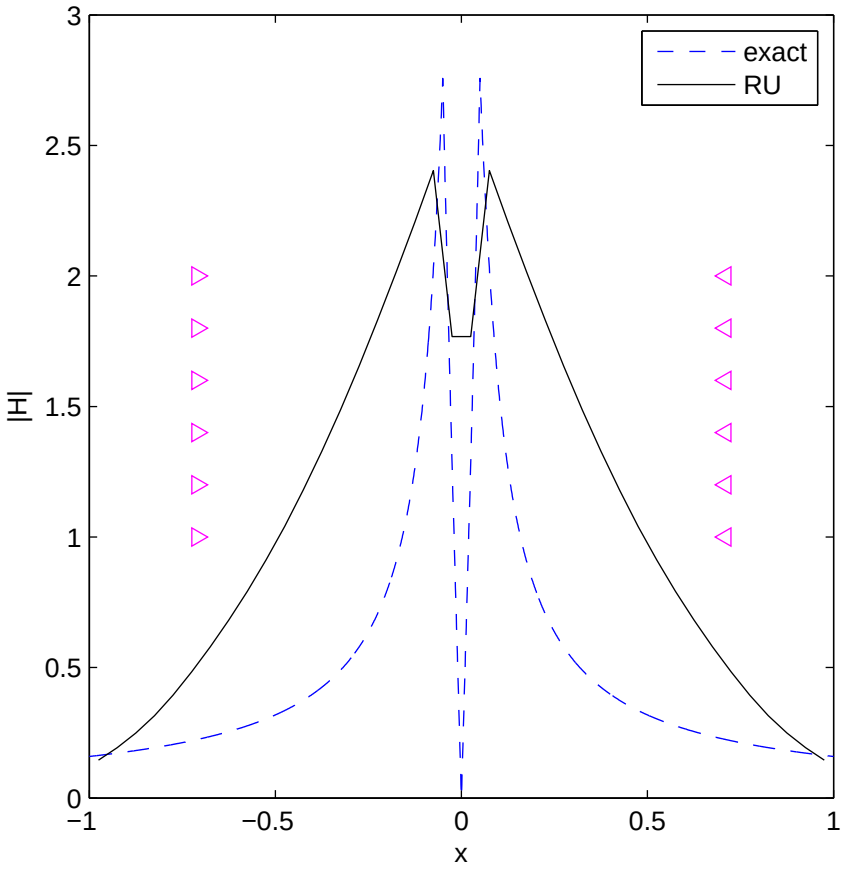


Figure 4.11: This shows the magnetic field solution as solved for by the rectangular upwind solver as plotted along the x-axis. As can be seen, the solver solution does not adhere to the analytic solution. The stacked triangles on both sides mark where the decay term is switched on for $r > 0.71$.

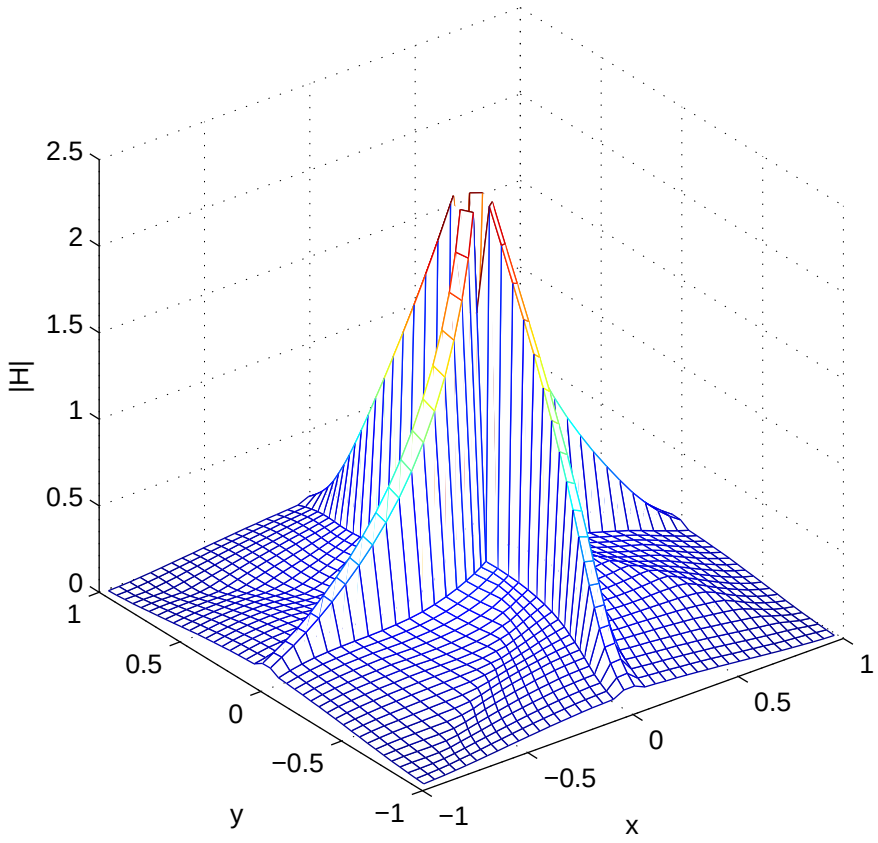


Figure 4.12: This shows the 3-D mesh solution for the magnetic field from a wire current using rectangular upwind solver on a grid with a twist; this solver fails miserably.

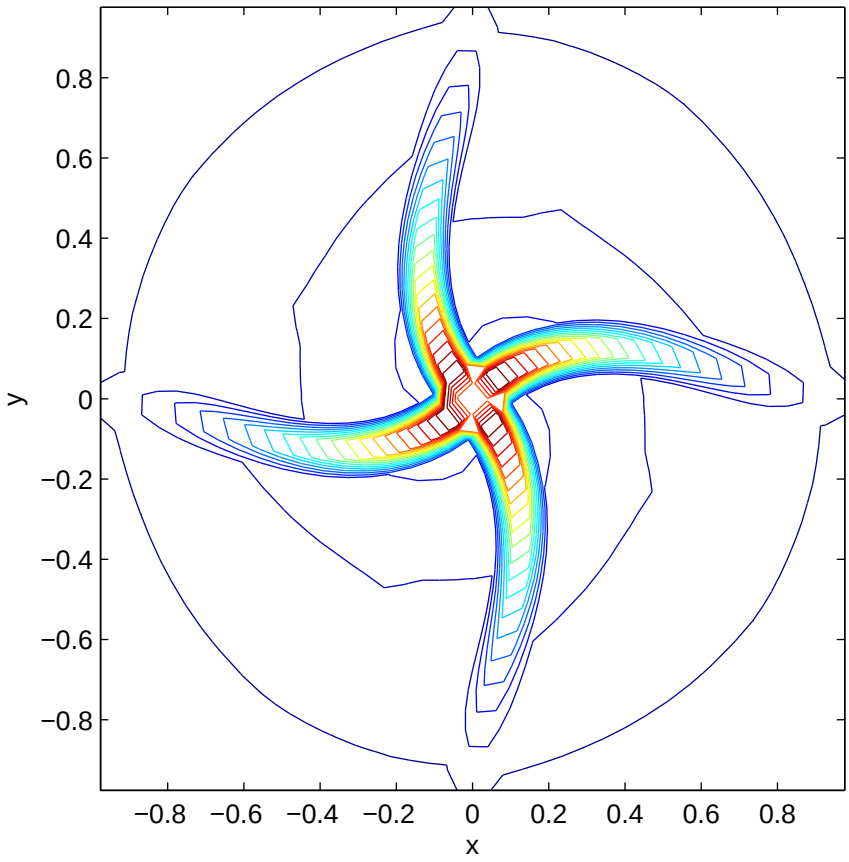


Figure 4.13: This shows the contour of the magnetic field solution from a wire current using rectangular upwind solver with a twist. As noticed by the spurs, the solver fails in a grid dependent fashion.

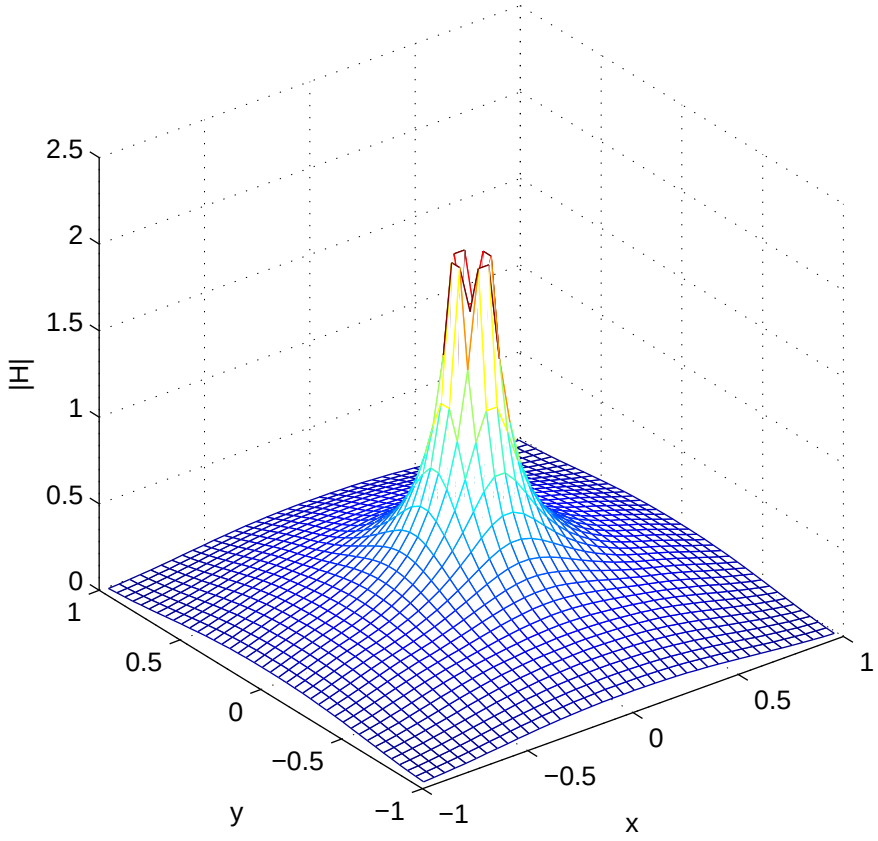


Figure 4.14: This shows the 3-D mesh solution to the wire current using triangular upwind solver; as seen, the solution shows the general symmetries of the analytic solution.

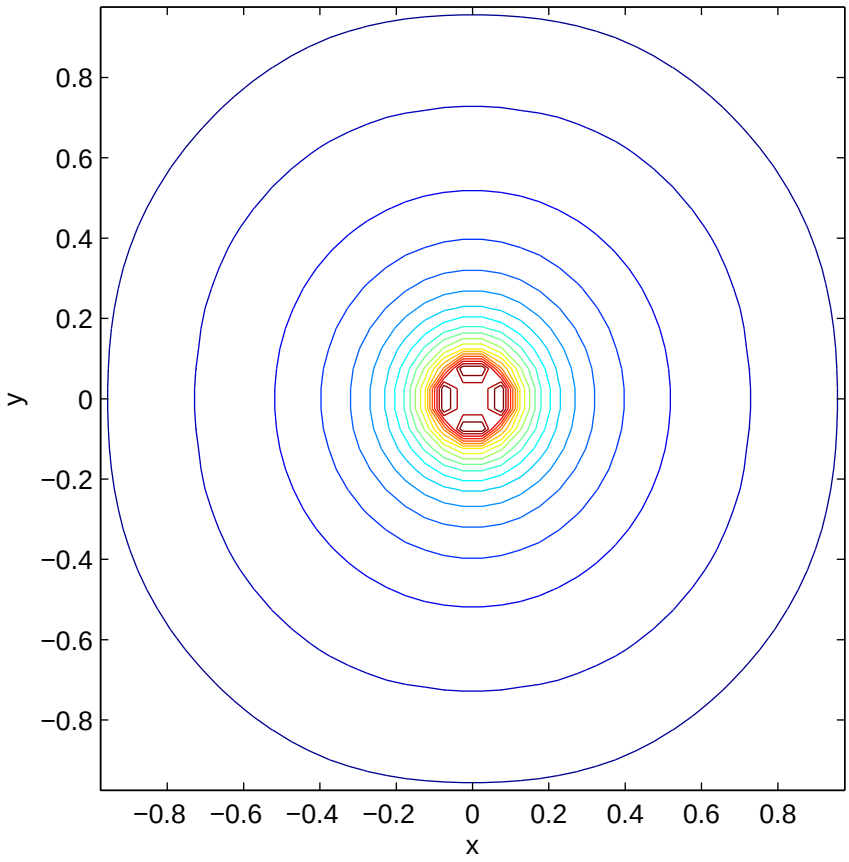


Figure 4.15: This shows the contours of the solution of the magnetic field from the wire current using the triangular upwind solver. As seen, the solution shows the radial symmetry expected of a wire current.

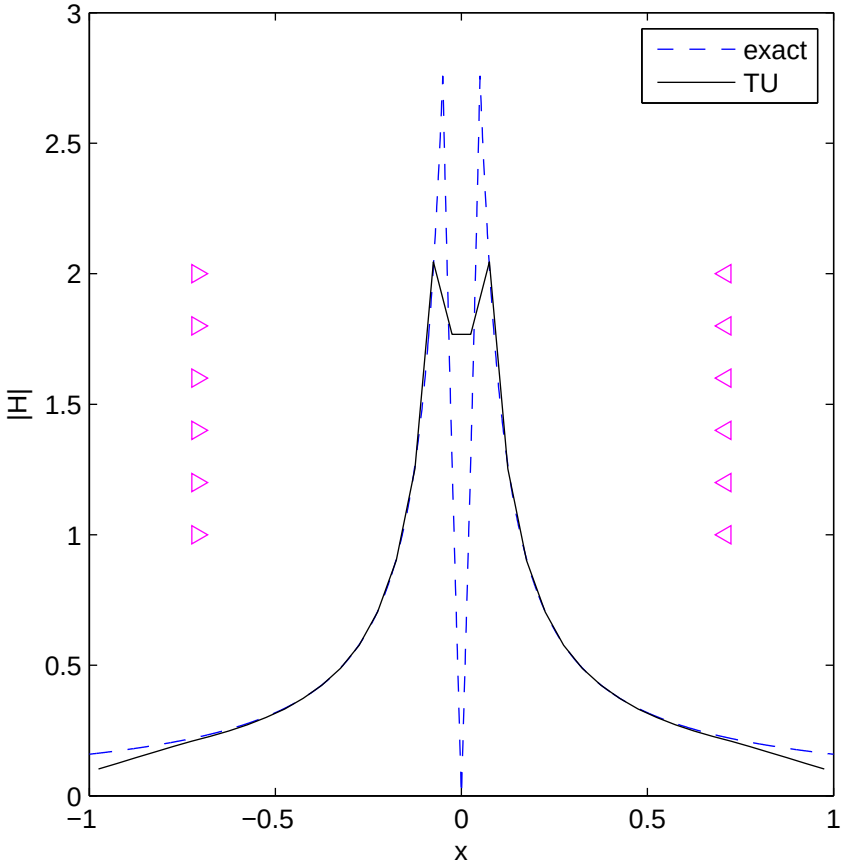


Figure 4.16: This shows the magnetic field solution as solved for by the triangular upwind as plotted along the x-axis. As seen, the solution closely follows the analytic solution until the damping region starts at $|r| > 0.71$. The stacked triangles on both sides mark where the decay term is switched on for $r > 0.71$.

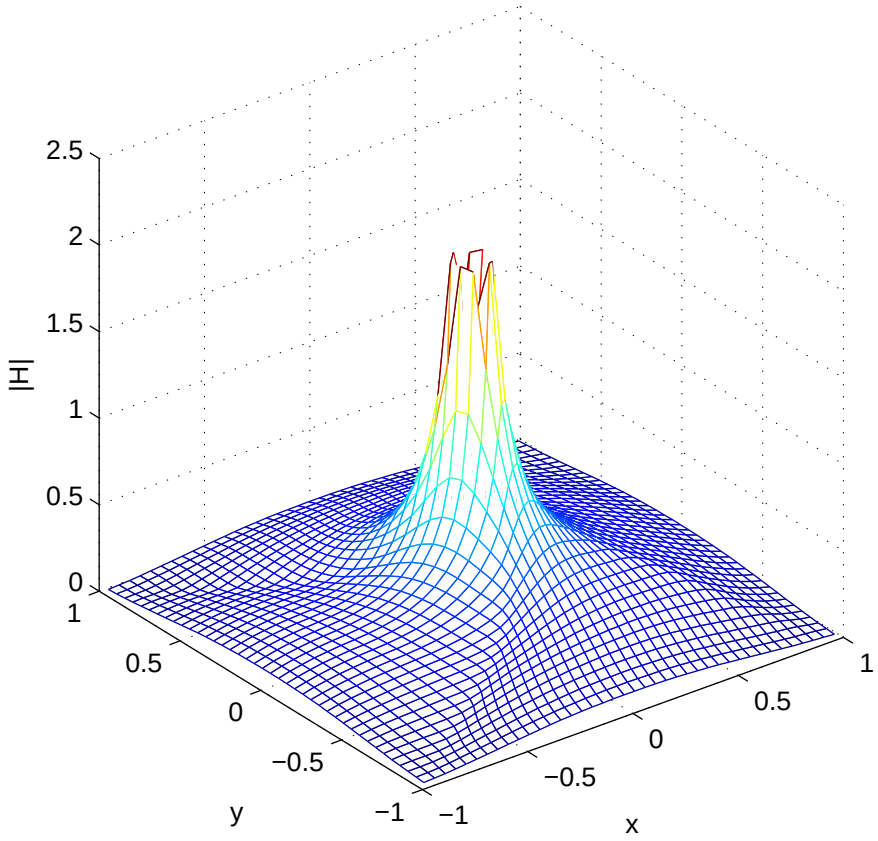


Figure 4.17: This shows the 3-D mesh for the solution to the magnetic field from the wire current using the triangular upwind solver applied to a grid with a twist. As can be seen, the solution appears to be correct and does not exhibit the grid effects as was demonstrated with the rectangular upwind solver.

a plot along j constant line near the x-axis in Figure 4.19. As shown, the solution bears a qualitative resemblance to the analytic solution; with the contour plot showing that the spiral artifacts seen in the mesh plot is an illusion due to the twisted grid; however, no dominant errors due to grid effects can be seen.

4.2.4 Triangular WENO

The triangular WENO solver is used to simulate the wire current. The solution to the magnetic field yielded by the solver is shown in a 3D mesh in Figure 4.20 and is shown

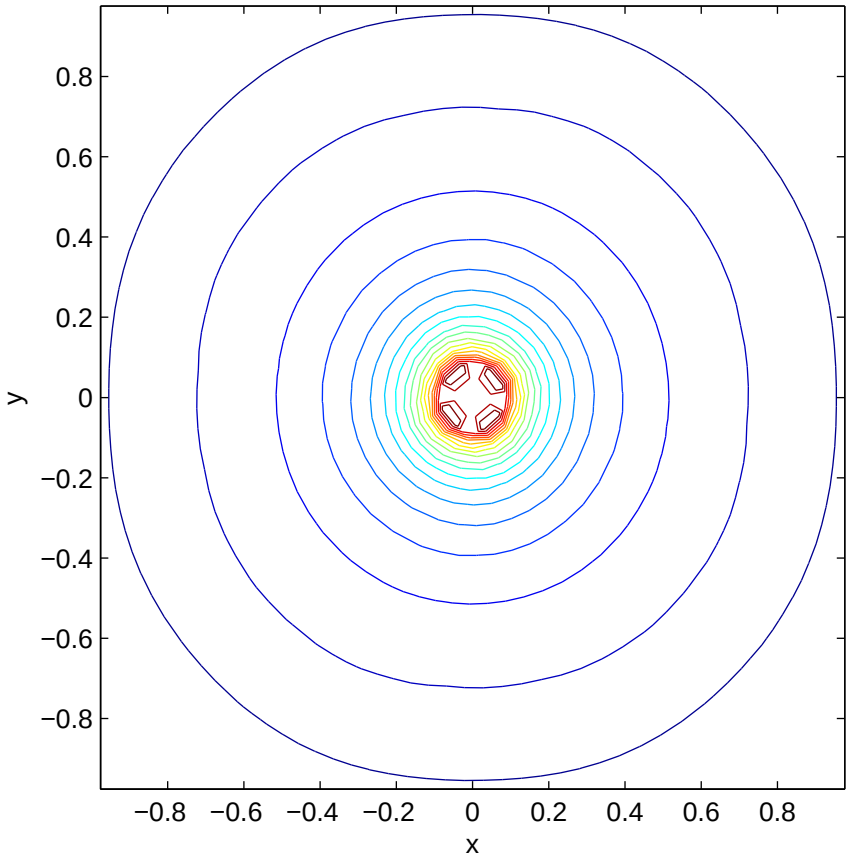


Figure 4.18: This shows the contours for the solution to the magnetic field from the wire current using the triangular upwind solver applied to a grid with a twist. As can be seen, the solution appears to be correct and does not exhibit the grid effects as was demonstrated with the rectangular upwind solver.

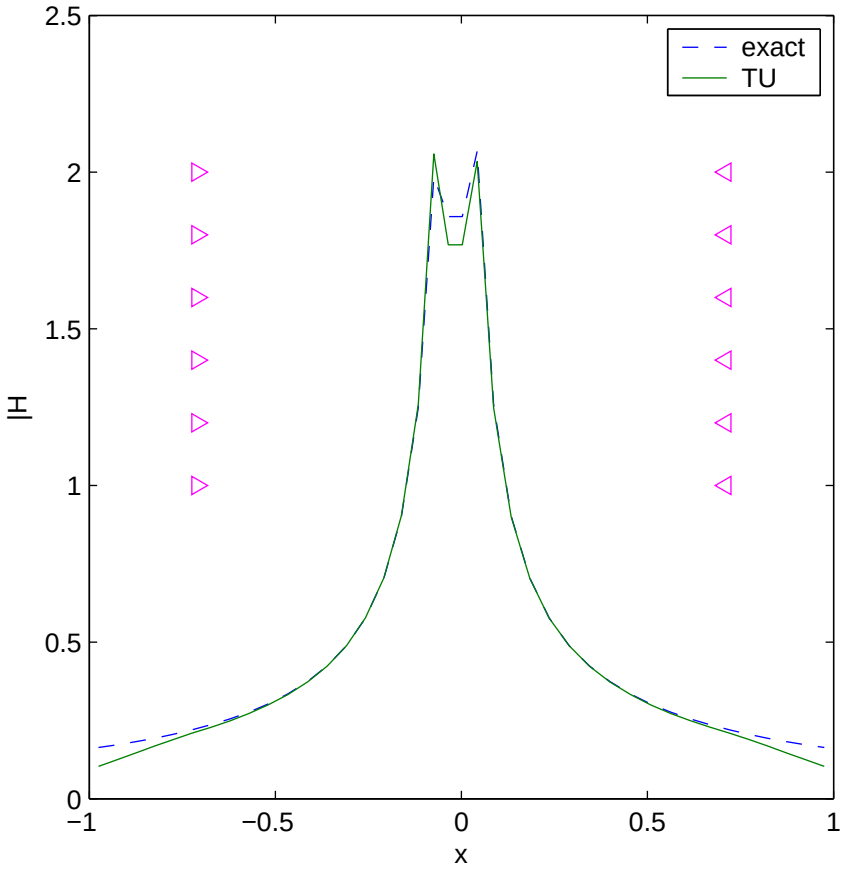


Figure 4.19: This compares the computed solution and analytic solution, plotted along an j-constant line of the mesh, with the edge and midpoints lying on the $x-axis$. The stacked triangles on both sides mark where the decay term is switched on for $r > 0.71$.

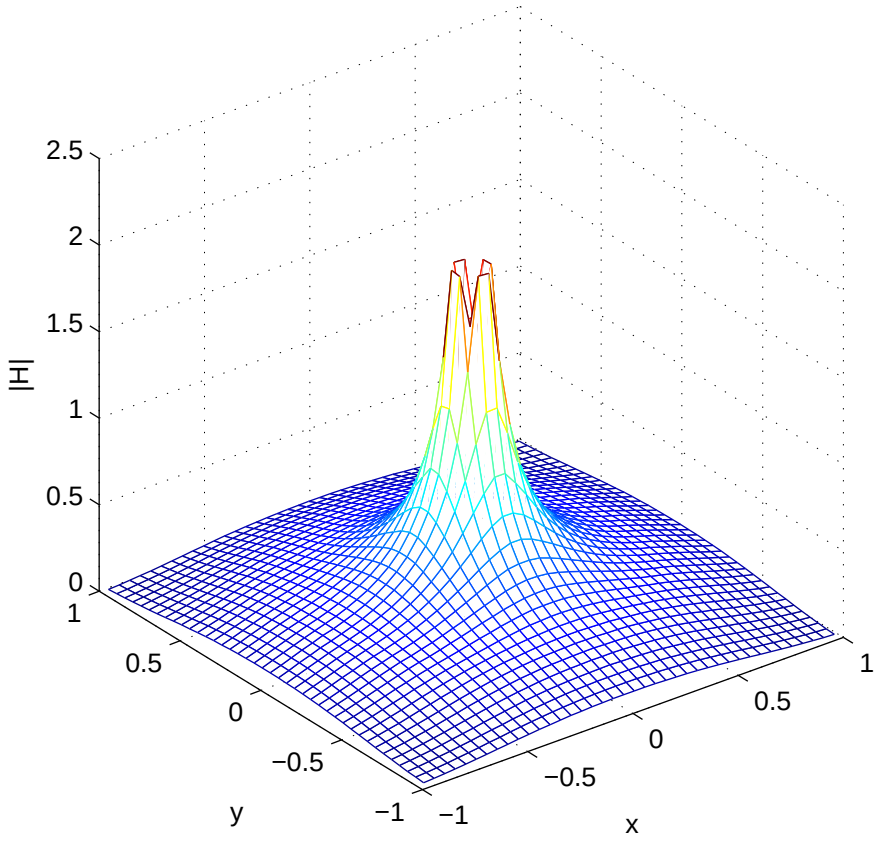


Figure 4.20: This shows the 3-D mesh for the solution to the magnetic field from the wire current using triangular WENO solver. As can be seen, the solution appears to be correct.

in contours of the magnitude in Figure 4.21. As shown, the solution bears a qualitative resemblance to the analytic solution; this is shown in the axis solution of Figure 4.22, which tracks, within the resolution of the grid, the analytic solution.

The triangular WENO solver is not repeated for the twisted grid because the coefficients for the WENO flux scheme were derived for a regular orthogonal grid, an assumption that the twisted grid violates.

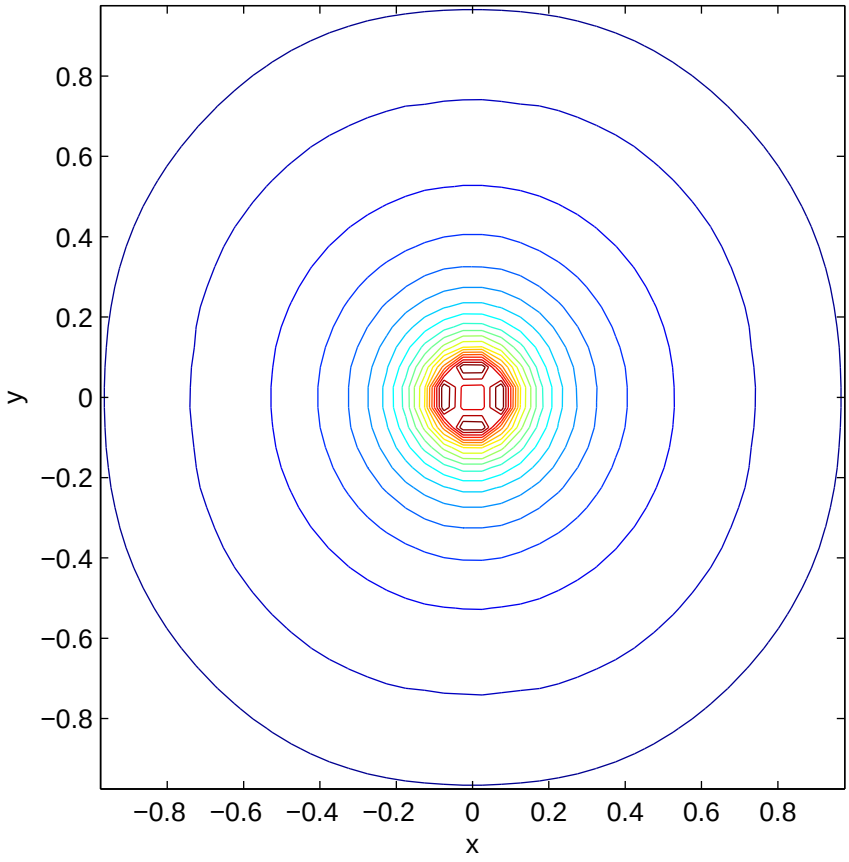


Figure 4.21: This shows the contours for the solution to the magnetic field from the wire current using the triangular WENO solver. As can be seen, the solution appears to be correct.

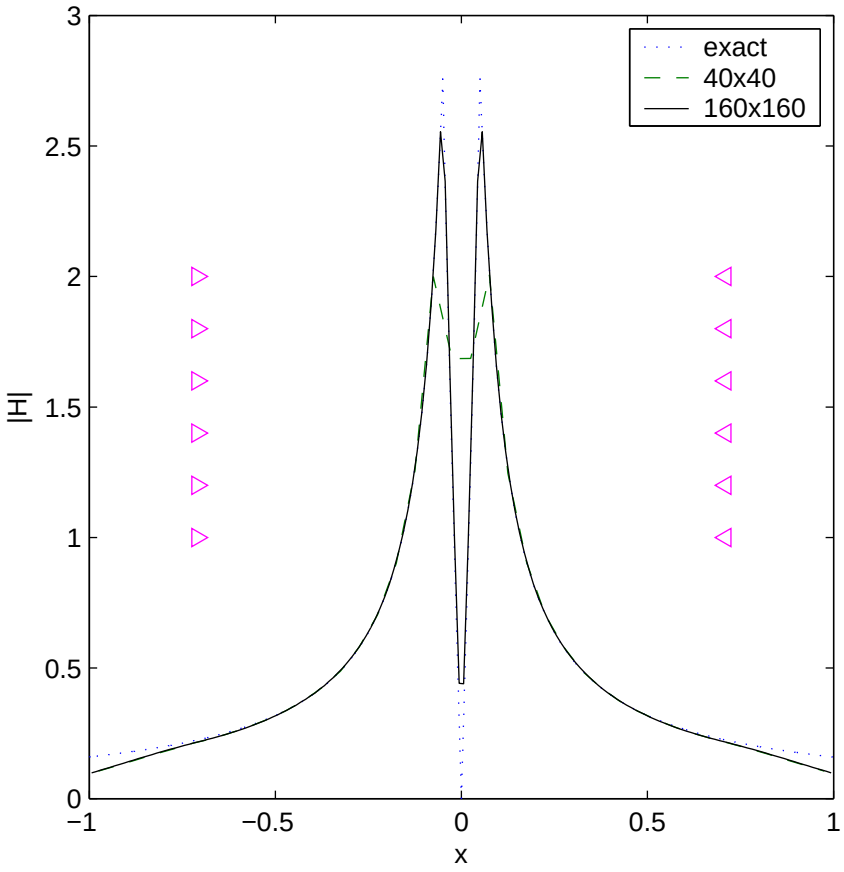


Figure 4.22: This shows the magnetic field solution as solved for by the triangular WENO as plotted along the x-axis. As seen, the solution closely follows the analytic solution until the damping region starts at $|r| > 0.71$. The stacked triangles on both sides mark where the decay term is switched on for $r > 0.71$. This demonstrates the closeness with two different simulations, one on a 40x40 grid, the other on a 160x160 grid.

Chapter 5

DISCUSSION

The Rectangular Upwind solver failed to produce a suitable solution, this was unexpected. This chapter discusses the results and explains why the triangular grid succeeds where the rectangular grid fails in the presence of sources.

In simulating the plane wave propagation, using the rectangular grid appeared to be successful, as shown in Figure 4.2. The solution did exhibit strong dissipation which is normal for a first order scheme; a higher order scheme such as Harten, Lax Wendroff, or WENO would improve the solution.

In simulating a wire current, switched on at $t = 0$ and left for steady state, the rectangular upwind solver fails to produce the physically correct solution. The resulting erroneous solution is not physically correct nor is it qualitatively close, for the solution is supposed to be axisymmetric, which, the four spurs "ridgelines" do not follow. Each ridgeline follows the grid, as shown in the solution on a twisted grid, and the solution along the ridgeline does not follow the analytic solution; the slope of the ridgelines depend on the boundary conditions used. The divergence-free condition is violated along the edges of the ridgelines, as shown in Figure 5.1, however, this error decreases $\mathcal{O}(\Delta x)$ as the grid is refined. Using higher order flux scheme were tried, but these do not resolve the problem.

Why?

As noted earlier, the characteristics of the Maxwell Equations show two eigenvalues that are zero. These zero eigenvalues cause poor dimensional coupling, which results in excessively high off-angle dissipation. Therefore, the erroneous solution starts off correctly with an initial outward pulse that is axisymmetric, but dissipation and poor dimensional coupling cause the ridgelines to form, and these ridgelines are maintained by the divergence error that results. The conclusion is that the zero eigenvalues are the main culprits; this takes the form of poor dimensional coupling, excessive off angle dissipation, and divergence

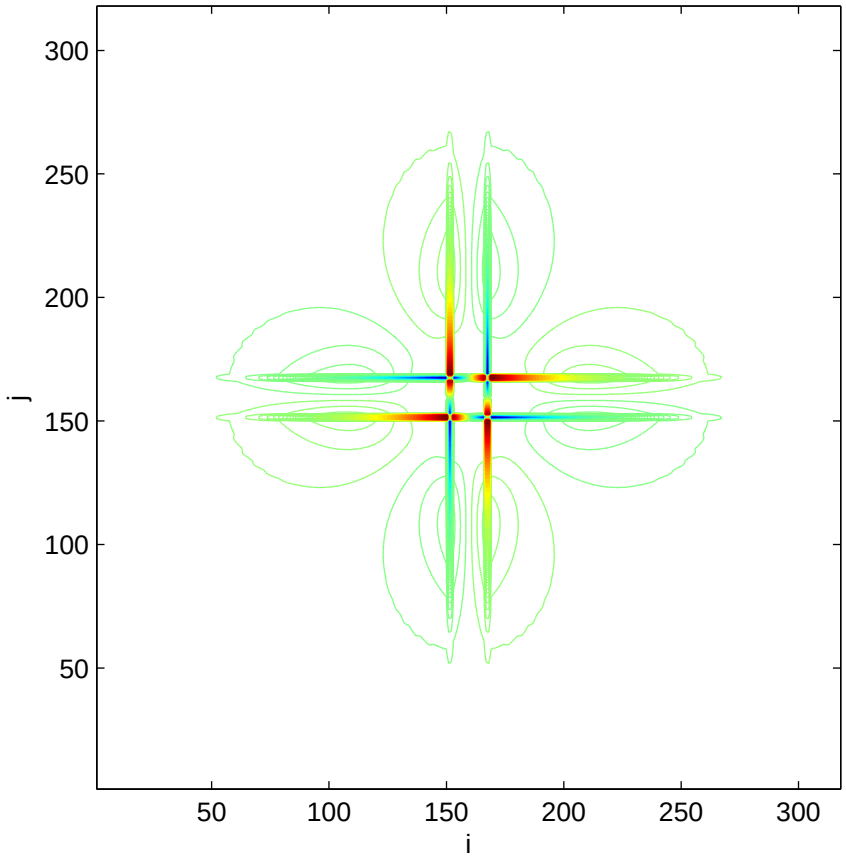


Figure 5.1: This contour plot shows the divergence errors in the simulation for the wire current using the rectangular upwind solver on a 320x320 grid, with 1,250 contours being shown. As shown, the major divergence errors are at the fine edge of the ridgelines; the rest of the solution shows negligible errors compared to the edges of the ridgelines.

errors.

It is the poor dimensional coupling that lead to the notion of using a triangular grid, to impose stronger dimensional coupling. The triangular grid was implemented by splitting each grid cell into two by adding in a diagonal; this was later increased to four subcells per cell using both diagonals as this proved advantageous in implementing boundary conditions.

Dimensional coupling is the ability of a scheme to propagate transverse waves in a particular direction; for example, y^+ going waves in the x direction. This coupling can be expressed by taking the flux Jacobians of Equation 3.6, and expressing each one in terms of positive and negative going waves, which, for the rectangular grid and $v = 1$ gives a coupling matrix

$$A^+B^+ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 \\ -\frac{1}{4} & 0 & 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \tag{5.1}$$

where

B^+ is the positive going flux Jacobian for the y direction.

Whereas, for a triangular grid, the coupling between x^+ and the positive across subinterfaces 4 and 5 (the diagonal going from upper left to lower right), the coupling matrix is

$$A^+\frac{\partial \boldsymbol{F}_4}{\partial \boldsymbol{Q}}^+ = \begin{bmatrix} \frac{1-\sqrt{2}}{8} & 0 & 0 & 0 & 0 & -\frac{1-\sqrt{2}}{8} \\ \frac{1-\sqrt{2}}{8} & 0 & 0 & 0 & 0 & -\frac{1-\sqrt{2}}{8} \\ 0 & 0 & \frac{2-\sqrt{2}}{8} & \frac{2-\sqrt{2}}{8} & 0 & 0 \\ 0 & 0 & \frac{1-\sqrt{2}}{8} & \frac{1-\sqrt{2}}{8} & 0 & 0 \\ 0 & 0 & \frac{1-\sqrt{2}}{8} & \frac{1-\sqrt{2}}{8} & 0 & 0 \\ -\frac{2-\sqrt{2}}{8} & 0 & 0 & 0 & 0 & \frac{2-\sqrt{2}}{8} \end{bmatrix} \tag{5.2}$$

where

$\frac{\partial \boldsymbol{F}_4}{\partial \boldsymbol{Q}}$ is the flux Jacobian across subinterfaces 4 and 5.

The coupling matrix for the triangular case is less sparse than the rectangular matrix.

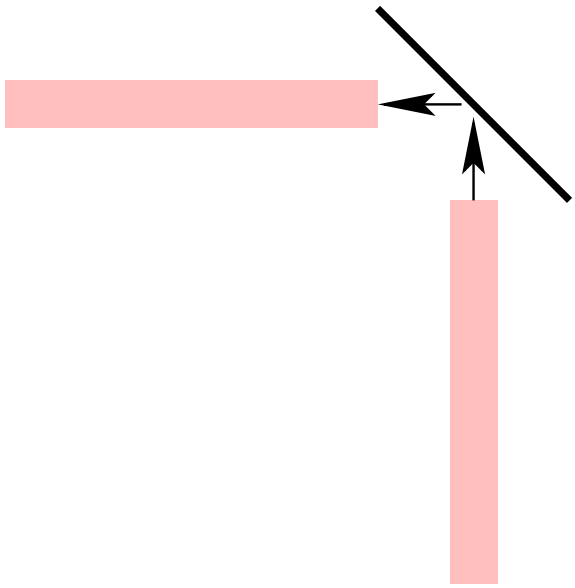


Figure 5.2: This figure shows light reflecting off a 45° mirror so it can continue propagating in at a right angle to its original direction.

The triangular upwind solver is applied to the wire current simulation. The magnetic field, shown in a 3D mesh in Figure 4.15, shows a qualitatively correct solution, which is verified by the other figures. Applying this solver to the plane wave shown in Figure 4.3 demonstrates that the triangular grid is capable of simulating wave propagation.

The triangular grid works, therefore, the dimensional coupling is better. By having the triangular interfaces, any wave propagating along the x-axis must turn at the diagonal interfaces; this forces transverse waves to start up. The effects of the diagonal to the propagating waves is similar to a mirror affecting beams of light. As shown in Figure 5.2, a diagonal mirror is required to change the direction of the light beam; the same thing is true of the solver, a diagonal is required to have adequate transverse propagation.

Once a first order conceptual solver works, the order of accuracy must be improved. Many higher order finite volume flux schemes are reliant upon a rectangular grid, therefore, a more robust method, weighted essentially non-oscillatory (WENO) scheme is used. The results of this solver are shown in Chapter 4 for both the plane wave propagation and the

wire current; both simulations result in satisfactory solutions. One oddity in this scheme, shown in Figure 4.4 is the superconvergence before it asymptotes to second order accuracy; this is a result of higher order error terms dominating the error in the coarser grids, these terms drop off as the grid is refined.

Other workarounds exist for the problems encountered on the rectangular grid; Munz[6] adds in a corrective potential, divergence cleaning can be used, changing to a potential formulation also works. These techniques sidestep the underlying issues, whereas the triangular grid directly addresses the poor dimensional coupling which permits simulations on arbitrary triangular grids using explicit methods.

Chapter 6

CONCLUSIONS AND FUTURE WORK

This document covers the essential developments for a Maxwell equation solver that can be incorporated into a two fluid plasma solver; the physics, the algorithms to be compared, the results of simulations with several solvers, and the core reasoning to choosing the solver.

The equations being solved are

$$\frac{\partial H}{\partial t} = -v^2 \nabla \times \boldsymbol{D}, \tag{6.1}$$

$$\frac{\partial D}{\partial t} = \nabla \times \boldsymbol{H} - \boldsymbol{J}_f \tag{6.2}$$

where details can be found in chapter 2. These equations were worked into a finite volume time domain (FVTD) formulation on both the rectangular and triangular grids.

In the presence of steady state sources, the rectangular upwind solver showed strong grid effects in the erroneous solution. The errors resulted from poor coupling between the characteristics propagating in two different directions. A stronger coupling is imposed by dividing up the grid cells, forming a triangular grid. Simulations showed that solvers, based on the triangular grid, worked to produce proper results.

Solving the Maxwell equations with sources requires attention to the scheme being used. The issues involved in the Maxwell equations could show up in other sets of equations that are based on two curl relationships with sources. However, that problem may prove more tractable if the flux Jacobians and their eigenvalues are based upon the conserved variables, as the component not being propagated would vary from time step to time step and from grid cell to grid cell.

Extensions of this work and further research could pursue in any of several directions. A more rigorous mathematical insight might better explain why the triangular grid succeeds where the rectangular grid fails. With this code working in two dimensions, it should be straightforward to extend the scheme into three dimensions with six subcells (in the shape

of pyramids) being used for each grid cell. Other methods exist for circumventing the grid effects problems; switching from a field formulation to a potential formulation is known to work, corrective terms can be introduced but these can become computationally intensive, simply the equations and relax the time step constraints by using Darwinian approximations, and divergence cleaning.

LIST OF REFERENCES

[1] L. Nimoy. Star trek vi: The undiscovered country. movie, 1991. Paramount Studios.

[2] J. Loverich. A finite volume algorithm for the two-fluid plasma system in one dimension. Master's thesis, University of Washington, 2003.

[3] J. Loverich and U. Shumlak. Approximate riemann solver for the two-fluid plasma model. *Journal of Computational Physics*, 187:620–638, 2003.

[4] K. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, pages 302–307, 1966.

[5] D. Sullivan. *Electromagnetic Simulation Using the FDTD Method*. IEEE Press, 2000.

[6] R. Schneider C. Munz and U. Vob. A finite-volume method for the maxwell equations in the time domain. *Society for Industrial and Applied Mathematics*, 22(2):449–475, 2000.

[7] J. Shang and R. Fithen. A comparative study of characteristic-based algorithms for the maxwell equations. *Journal of Computational Physics*, 125:378–394, 1996.

[8] R. Resnick D. Halliday and K. Krane. *Physics*, volume 2. John Wiley & Sons, Inc., fourth edition, 1992.

[9] D. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, third edition, 1999.

[10] J. Kong. *Electromagnetic Wave Theory*. John Wiley & Sons, Inc., 1986.

[11] G. Jiang and C. Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126:202–228, 1996.

[12] C. Hu and C. Shu. A technique of treating negative weights in weno schemes. *Journal of Computational Physics*, 175:108–127, 2002.

[13] S. Ray A. Peterson and R. Mittra. *Computational Methods for Electromagnetics*. IEEE Press, 1998.

[14] D. Balsara and C. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasinly high order of accuracy. *Journal of Computational Physics*, 160:405–452, 2000.

[15] B. Bidegaray and J. Ghidaglia. Multidimensional corrections to cell-centered finite volume mthods for maxwell equations. *Applied Numerical Mathematics*, 44:281–298, 2003.

[16] M. Chari and S. Salon. *Numerical Methods in Electromagnetism*. Academic Press, 2000.

[17] R. Goldston and P. Rutherford. *Introduction to Plasma Physics*. Institute of Physics Publising, 1997.

[18] C. Hirsch. *Numerical Computation of Internal and External Flows*. John Wiley & Sons, Inc., 1988.

[19] C. Hu and C. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999.

[20] J. Kevorkian. *Partial Differential Equations: Analytical Solution Techniques*. Springer, 2000.

[21] D. Kröner. *Numerical Schemes for Conservation Laws*. John Wiely & Sones and B. G. Teubner Publisers, 1997.

[22] J. Langseth and R. LeVeque. A wave propagation method for three-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 165:126–166, 2000.

[23] R. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, 1992.

[24] R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge United Press, 2002.

[25] M. Remaki S. Piperno and L. Fezoui. A nondiffusive finite volume scheme for the three-dimensional maxwell's equations on unstructured meshes. *Society for Industrial and Applied Mathematics*, 39:2089–2108, 2002.

[26] M. Sadiku. *Elements of Electromagnetics*. Oxford University Press, 2001.

[27] L. Trefethen and D. Baus III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[28] B. Udrea. *An Advanced Implicit Solver for MHD*. PhD thesis, University of Washington, 1999.

Appendix A

MAGNETIC FIELD OF A WIRE

This appendix outlines the steps to determine the magnetic field for both square and triangle wires. It presents the analytic equations describing the magnetic field and compares the solutions.

A.1 Overview

The magnetic field of a round wire is

$$|\boldsymbol{B}| = \frac{J}{2\mu_0} \begin{cases} \frac{d^2}{4r} & r \geq \frac{d}{2}, \\ r & r < \frac{d}{2}. \end{cases} \tag{A.1}$$

This solver represents a current as either a square or triangle wire. The differences between the round, square, and triangle wires are needed.

A.2 Setup

Three types of wires are used for analysis. Section A.4 uses a wire with an arbitrary cross section, as shown in Figure A.1. Section A.5 uses a wire with a square cross section, as shown in Figure A.2. Section A.6 uses a wire with a triangle cross section, as shown in Figure A.3. For each case, the centroid of the wire is at the origin and the current is uniform within the wire.

Local coordinates are used within the wire. a describes the distance along the x-axis. b describes the distance along the y-axis. c describes the distance along the z-axis. $\boldsymbol{p}$ is defined as the separation between a point (a,b,c) in the wire and some other point (x,y). p is

$$\boldsymbol{p}\,(x, y, a, b, c) \equiv (x - a)\,\hat{\boldsymbol{e}}_x + (y - b)\,\hat{\boldsymbol{e}}_y - c\hat{\boldsymbol{e}}_z. \tag{A.2}$$

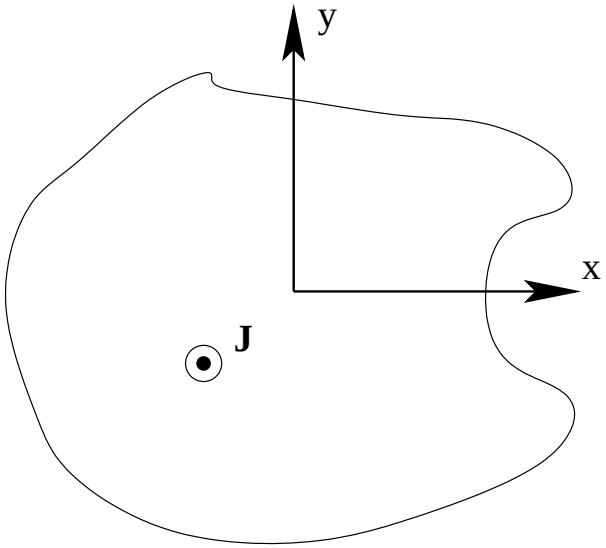


Figure A.1: Cross section of an arbitrary wire.

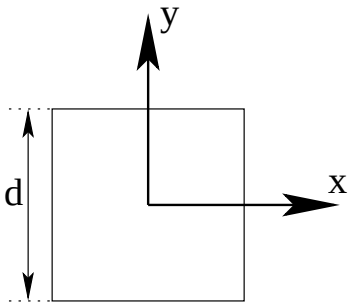


Figure A.2: Cross section of an square wire.

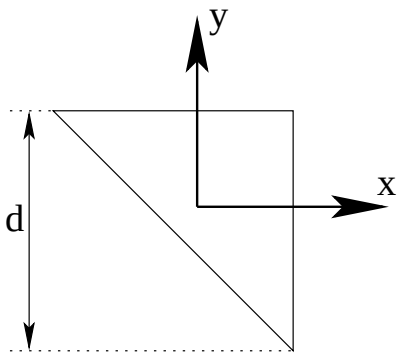


Figure A.3: Cross section of an triangle wire.

The current is uniform within the wire and travels only in the $\hat{\boldsymbol{e}}_z$ direction. Therefore

$$\boldsymbol{J} = J_0 \hat{\boldsymbol{e}}_z$$

where J_0 is a constant. $\boldsymbol{r}$ is defined as the projection of $\boldsymbol{p}$ into the x-y plane so

$$\boldsymbol{r}\,(x, y, a, b) \equiv (x - a)\,\hat{\boldsymbol{e}}_x + (y - b)\,\hat{\boldsymbol{e}}_y. \tag{A.3}$$

A.3 Biot-Savart

The Biot-Savart Law is used to find the analytic solution for the magnetic field. The Biot-Savart Law is[9]

$$\boldsymbol{B} = \frac{\mu_0}{4\pi} \iiint \frac{\boldsymbol{J} \times \hat{\boldsymbol{p}}}{p^2}\, \mathrm{d}^3\boldsymbol{p}. \tag{A.4}$$

Apply $\boldsymbol{p}$ and $\boldsymbol{J}$ from Section A.2 and

$$\begin{aligned}
\boldsymbol{B} &= \frac{\mu_0}{4\pi} \iiint \frac{\boldsymbol{J} \times \boldsymbol{p}}{p^3}\, \mathrm{d}^3\boldsymbol{p}, \\
\boldsymbol{B} &= \frac{\mu_0 J_0}{4\pi} \iiint \frac{\hat{\boldsymbol{e}}_z \times \boldsymbol{p}}{p^3}\, \mathrm{d}^3\boldsymbol{p}, \\
\boldsymbol{B} &= \frac{\mu_0 J_0}{4\pi} \iiint \frac{-(y - b)\,\hat{\boldsymbol{e}}_x + (x - a)\,\hat{\boldsymbol{e}}_y}{\sqrt{(r^2 + c^2)^3}}\, \mathrm{d}^3\boldsymbol{p}.
\end{aligned} \tag{A.5}$$

A.4 Z-Direction

Equation A.5 is integrated along the z-axis using the integration variable c. Let Z be some positive real number and integrate Equation A.5 from $c = -\frac{Z}{r} \rightarrow \frac{Z}{r}$. So,

$$\begin{aligned}
\boldsymbol{B} &= \frac{\mu_0 J_0}{4\pi} \iint_A \int_{-\frac{Z}{r}}^{\frac{Z}{r}} \frac{-(y-b)\,\hat{\boldsymbol{e}}_x + (x-a)\,\hat{\boldsymbol{e}}_y}{\sqrt{(r^2+c^2)^3}}\,\mathrm{d}c\,\mathrm{d}^2\boldsymbol{r}, \\
\boldsymbol{B} &= \frac{\mu_0 J_0}{4\pi} \iint_A \{-(y-b)\,\hat{\boldsymbol{e}}_x + (x-a)\,\hat{\boldsymbol{e}}_y\} \int_{-\frac{Z}{r}}^{\frac{Z}{r}} \frac{1}{\sqrt{(r^2+c^2)^3}}\,\mathrm{d}c\,\mathrm{d}^2\boldsymbol{r}
\end{aligned}$$

where A is the cross-sectional area of the wire. Therefore

$$\boldsymbol{B} = \frac{\mu_0 J_0}{4\pi} \iint_A \frac{-(y-b)\,\hat{\boldsymbol{e}}_x + (x-a)\,\hat{\boldsymbol{e}}_y}{r^3} \int_{-\frac{Z}{r}}^{\frac{Z}{r}} \frac{1}{\sqrt{\left(1+\frac{c^2}{r^2}\right)^3}}\,\mathrm{d}c\,\mathrm{d}^2\boldsymbol{r}.$$

Let $u \equiv \frac{c}{r}$ so $\mathrm{d}c = r\mathrm{d}u$ and $Z_u \equiv \frac{Z}{r}$ thus

$$\begin{aligned}
\boldsymbol{B} &= \frac{\mu_0 J_0}{4\pi} \iint_A \frac{-(y-b)\,\hat{\boldsymbol{e}}_x + (x-a)\,\hat{\boldsymbol{e}}_y}{r^3} \int_{-Z_u}^{Z_u} \left(1+u^2\right)^{-\frac{3}{2}} r\,\mathrm{d}u\,\mathrm{d}^2\boldsymbol{r}, \\
\boldsymbol{B} &= \frac{\mu_0 J_0}{4\pi} \iint_A \frac{-(y-b)\,\hat{\boldsymbol{e}}_x + (x-a)\,\hat{\boldsymbol{e}}_y}{r^2} \int_{-Z_u}^{Z_u} \left(1+u^2\right)^{-\frac{3}{2}}\,\mathrm{d}u\,\mathrm{d}^2\boldsymbol{r}.
\end{aligned}$$

Extract the inner integral so

$$\begin{aligned}
\lim_{Z_u\to\infty} \int_{-Z_u}^{Z_u} \left(1+u^2\right)^{-\frac{3}{2}}\,\mathrm{d}u = \lim_{Z_u\to\infty} \left[\frac{u}{\sqrt{u^2+1}}\right]_{-Z_u}^{Z_u} &= \lim_{Z_u\to\infty} 2\frac{Z_u}{\sqrt{Z_u^2+1}} \\
\approx 2\frac{Z_u}{\sqrt{Z_u^2}} &= 2.
\end{aligned}$$

Therefore, the equation for the magnetic field is reduced to

$$\boldsymbol{B} = \frac{\mu_0 J_0}{2\pi} \iint_A \frac{-(y-b)\,\hat{\boldsymbol{e}}_x + (x-a)\,\hat{\boldsymbol{e}}_y}{r^2}\,\mathrm{d}^2\boldsymbol{r}. \tag{A.6}$$

A.5 Square Wire

A square wire of width d is shown in Figure A.2. Equation A.6 is integrated over this cross-section. This integration is performed using Maple[1].

---

[1]Maple is a registered trademark of Waterloo Maple Inc.

The x-component is

$$B_x = \frac{\mu_0 J_0}{2\pi} \left\{ \begin{array}{l} \left(-y - \frac{d}{2}\right) \tan^{-1}\left(\frac{2x-d}{2y+d}\right) + \left(y - \frac{d}{2}\right) \tan^{-1}\left(\frac{2x-d}{2y-d}\right) + \\ \left(y + \frac{d}{2}\right) \tan^{-1}\left(\frac{2x+d}{2y+d}\right) + \left(-y + \frac{d}{2}\right) \tan^{-1}\left(\frac{2x+d}{2y-d}\right) + \\ \left(-\frac{x}{2} + \frac{d}{4}\right) \ln\left(2x^2 - 2xd - 2yd + d^2 + 2y^2\right) + \\ \left(\frac{x}{2} - \frac{d}{4}\right) \ln\left(2x^2 - 2xd + 2yd + d^2 + 2y^2\right) + \\ \left(\frac{x}{2} + \frac{d}{4}\right) \ln\left(2x^2 + 2xd - 2yd + d^2 + 2y^2\right) + \\ \left(-\frac{x}{2} - \frac{d}{4}\right) \ln\left(2x^2 + 2xd + 2yd + d^2 + 2y^2\right) \end{array} \right\}. \tag{A.7}$$

The y-component is

$$B_y = \frac{\mu_0 J_0}{2\pi} \left\{ \begin{array}{l} \left(-x - \frac{d}{2}\right) \tan^{-1}\left(\frac{2y-d}{2x+d}\right) + \left(x - \frac{d}{2}\right) \tan^{-1}\left(\frac{2y-d}{2x-d}\right) + \\ \left(x + \frac{d}{2}\right) \tan^{-1}\left(\frac{2y+d}{2x+d}\right) + \left(-x + \frac{d}{2}\right) \tan^{-1}\left(\frac{2y+d}{2x-d}\right) + \\ \left(\frac{y}{2} - \frac{d}{4}\right) \ln\left(2x^2 - 2xd - 2yd + d^2 + 2y^2\right) + \\ \left(-\frac{y}{2} - \frac{d}{4}\right) \ln\left(2x^2 - 2xd + 2yd + d^2 + 2y^2\right) + \\ \left(-\frac{y}{2} + \frac{d}{4}\right) \ln\left(2x^2 + 2xd - 2yd + d^2 + 2y^2\right) + \\ \left(\frac{y}{2} + \frac{d}{4}\right) \ln\left(2x^2 + 2xd + 2yd + d^2 + 2y^2\right) \end{array} \right\}. \tag{A.8}$$

A.6 Triangle Wire

A triangle wire of width d is shown in Figure A.3. Equation A.6 is integrated over this cross-section. This integration is performed using Maple.

The x-component is

$$B_x = \frac{\mu_0 J_0}{2\pi} \{\}. \tag{A.9}$$

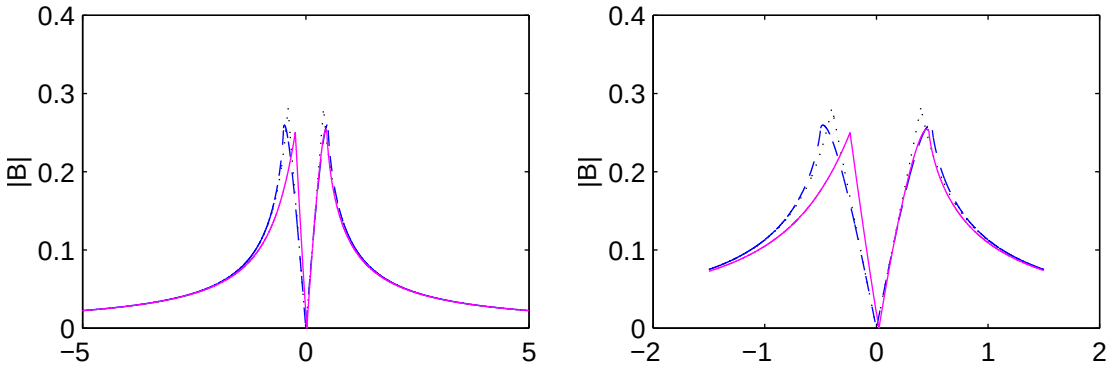
The y-component is

$$B_y = \frac{\mu_0 J_0}{2\pi} \{\}. \tag{A.10}$$

A.7 Comparison

The magnetic field for the round, square, and triangle wires is compared in Figure A.4. For each wire, the cross-sectional area is 1, and the current is 1.

Figure A.4a shows that the magnetic fields converge in the far-field. The solutions are nearly identical at a distance of 3 units.



(a) $x, y = -5 \rightarrow 5$ (b) $x, y = -1.5 \rightarrow 1.5$

Figure A.4: This is a comparison of the magnetic field for the round, square, and triangle wire. The dotted line represents the round wire. The dashed line represents the square wire. The solid line represents the triangle wire. (a) comparison taken along the diagonal from (-5,-5) to (5,5). (b) comparison taken along the diagonal from (-1.5,-1.5) to (1.5,1.5). For each wire, the cross-sectional area is 1, and the current is 1.

Figure A.4b shows that the magnetic fields diverge in the near field. The peaks of each field occurs at the surface of their respective wires. The solutions exhibit differing behavior near the surface of the wires.

Appendix B

METHOD OF UNDETERMINED COEFFICIENTS AS APPLIED TO INTERPOLATION

The method of undetermined coefficients is a general technique for solving problems where the general form of the solution is known. This is applied to the interpolation at a known point but the values of the function are unknown.

For example, take some function $u(t)$ that is known at finite time intervals $k = 0...N$ where N is the number of intervals such that

$$u_k = u(t - k\,\Delta t), \qquad t_k = t - k\,\Delta tm$$

and k is an index into the past. The method of undetermined coefficients can be applied to determine the first derivative in time $\dot{u}(t)$ at $k = 0$ to second order accuracy.

A solution is assumed of the form involving the first three values of u

$$\dot{u} \approx au_0 + bu_1 + cu_2. \tag{B.1}$$

where a, b, c are to be determined.

A Taylor series expansion is applied at u_1 and u_2 relative to u_0 so

$$u_1 \approx u_0 - \Delta t\dot{u}_0 + \frac{1}{2}(\Delta t)^2 \ddot{u}_0, \tag{B.2}$$

$$u_2 \approx u_0 - 2\Delta t\dot{u}_0 + 2(\Delta t)^2 \ddot{u}_0 \tag{B.3}$$

Expand Equation (B.1) for

$$\dot{u}_0 \approx au_0 + b\left(u_0 - \Delta t\dot{u}_0 + \frac{1}{2}(\Delta t)^2 \ddot{u}_0\right) + c\left(u_0 - 2\Delta t\dot{u}_0 + 2(\Delta t)^2 \ddot{u}_0\right). \tag{B.4}$$

Convert this into a matrix form

$$\begin{bmatrix} u_0 & u_0 & u_0 \\ 0 & -\Delta t\dot{u}_0 & -2\Delta t\dot{u}_0 \\ 0 & \frac{1}{2}(\Delta t)^2 & 2(\Delta t)^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ u_0 \\ 0 \end{bmatrix}. \tag{B.5}$$

Simplify,

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & \frac{1}{2} & 2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{\Delta t} \\ 0 \end{bmatrix} \tag{B.6}$$

Solve

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & \frac{3}{2} & 1 \\ 0 & -2 & -2 \\ 0 & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{\Delta t} \\ 0 \end{bmatrix} \tag{B.7}$$

Therefore

$$\dot{u}_0 \approx \frac{1}{\Delta t} \left(\frac{3}{2} u_0 - 2u_1 + \frac{1}{2} u_2 \right) \tag{B.8}$$

which is a second order accurate approximation of $\dot{u}$ at $k = 0$.

This example illustrated the technique that is applied to interpolation for the WENO of this thesis.