ERROR ANALYSIS OF HIGHER ORDER TIME-DOMAIN FINITE ELEMENT METHODS FOR THE ONE-DIMENSIONAL MAXWELL'S EQUATIONS

By

Jeffrey Cox

James C Newman III
Professor of Computational Engineering
Committee Chair

Kidambi Sreenivas
Professor of Computational Engineering
Committee Member

Robert Webster
Professor of Computational Engineering
Committee Member

ERROR ANALYSIS OF HIGHER ORDER TIME-DOMAIN FINITE ELEMENT METHODS FOR THE ONE-DIMENSIONAL MAXWELL'S EQUATIONS

By

Jeffrey Cox

A Thesis Submitted to the Faculty of the University of
Tennessee at Chattanooga in Partial Fulfillment
of the Requirements of the Degree of
Master of Science: Engineering

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

August, 2016

ABSTRACT

A well known nodal discontinuous Galerkin finite element method has been extended for higher order temporal accuracy using several schemes. While common in computational fluid dynamics, less research has been conducted with these methods for computational electromagnetics. A stabilized finite element method utilizing the Streamline/Upwind Petrov-Galerkin approach is explored. This work examines several higher order temporally accurate schemes to test their viability for the Maxwell equations. Only the one-dimensional case is considered. The temporal integration methods utilized are the first two backward differentiation formula (BDF), second through fourth order modified extended backward differentiation formula (MEBDF), and second through fourth order explicit first stage singly diagonally implicit Runge-Kutta (ESDIRK) schemes. A problem using a simple Gaussian pulse to which the analytical solution is known is used to verify the desired order of accuracy. Fifth-order spatial integration using Legendre polynomials, so spatial errors will be much smaller than temporal errors.

Table of Contents

Chapter

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

In computational electromagnetics, two methods dominate the field: the finite difference time domain algorithm and the method of moments. The finite difference time domain is extremely simple to implement, but unfortunately is very difficult to extend past second order spatial accuracy. The method of moments is most commonly used in frequency domain problems, and while it can have a high degree of accuracy, it scales very poorly for larger problems [1]. The finite-element method in contrast provides high order spatial accuracy while still scaling very well for large domains. Until recently, the difficulty of formulating stable and high-order accurate schemes has prevented their widespread use, but the development of stable finite element methods should now open the door to their use [2]. Furthermore, since finite element methods use a much smaller stencil than finite difference time domain methods, they provide excellent parallel performance.

In this thesis, a variety of the higher order finite element methods are investigated, and it is shown that they provide the expected order of accuracy for the one-dimensional case. Though algorithmically more challenging than the finite difference time domain methods commonly in use, the potential savings for large problems that need a high degree of accuracy cannot be ignored. In particular, it is extremely straightforward to vary the spatial order of the finite element schemes used here, which allows for high fidelity modeling using relatively few points in the domain.

Chapter 2

LITERATURE REVIEW

There has long been an interest in applying finite element techniques to computational electromagnetics. The first attempt appears to have been made by J.H. Coggon in 1971, where he successfully applied a finite element methodology to replace the energy variational equation with matrix equations [3]. Little further work was done using finite element techniques until 1988, when Jin *et al*. [4] used a hybrid finite element method to solve for electromagnetic scattering. This was followed in 1990 by Yuan [5], who also used a hybrid moment and finite element method to simulate a more general case involving 3-D scattering.

Discontinuous Galerkin methods for the time domain Maxwell equations did not become popular for computational electromagnetics until the 21st century. In the past fifteen years, great strides have been made in applying finite element methods to computational electromagnetics. The extensively used finite difference time domain method proposed by Yee [6] in 1966, that has dominated this field, is unsuitable for many of the larger problems being studied today. The difficulty with extending finite difference techniques past second order spatial accuracy severely limits their utility. Though a number of researchers have investigated applying high order spatially accurate discontinuous Galerkin methods to the time domain Maxwell equations, the work of Hesthaven has been particularly influential [2, 7–9]. In her many papers Hesthaven presents a methodology for obtaining effectively arbitrarily high order spatial accuracy. In the current work, the discontinuous-Galerkin formulation of Hesthaven is utilized and extended to include high order temporal accuracy.

However, up to this point, most temporal integration schemes used with finite element methods for the Maxwell equations have been either explicit schemes or low order implicit schemes. In 2009, Fahs [10] proposed a fourth order leap-frog time integration scheme, and proved it to be stable. Note that, while fourth order, this method is explicit, and thus time step cannot be increased past the critical timestep ensuring stability. Anderson $et\ al.$ [11] in 2011 presented an implicit scheme using a Petrov-Galerkin approach, but the method is limited to second order temporal accuracy. Therefore, to date the investigation into the use of higher order temporally accurate implicit schemes to evaluate the Maxwell equations is nonexistent in the literature.

Chapter 3

GOVERNING EQUATIONS

The time-domain Maxwell equations may be written as

$$\frac{\partial \vec{D}}{\partial t} = \vec{\nabla} \times \vec{H} \tag{3.1}$$

$$\frac{\partial \vec{B}}{\partial t} = \vec{\nabla} \times \vec{E} \tag{3.2}$$

where $\vec{D}$ is the electric flux density, $\vec{H}$ is the magnetic field, $\vec{B}$ is the magnetic flux density, $\vec{E}$ is the electric field, and $\vec{\nabla}$ is the gradient operator. In free space, these quantities are related by the associated constitutive equations

$$\vec{D} = \epsilon \vec{E} \tag{3.3}$$

$$\vec{B} = \mu \vec{H} \tag{3.4}$$

where ϵ and μ are the permittivity and permeability, respectively. On substitution of equations (3.3) and (3.4) into equations (3.1) and (3.2), for the one-dimensional case yields

$$\epsilon \frac{\partial E^y}{\partial t} = -\frac{\partial H^z}{\partial x} \tag{3.5}$$

$$\mu \frac{\partial H^z}{\partial t} = -\frac{\partial E^y}{\partial x} \tag{3.6}$$

A requirement of the Maxwell equations is that the fields must be orthogonal to each other, so for the one-dimensional case, the E and H fields can be assumed to each be along a different axis. In this thesis, the E field is considered to be along the y axis, and H field along the z axis. These equations may be written in conservative form as

$$Q\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{F}(U)}{\partial x} = 0 \tag{3.7}$$

where

$$\vec{U} = \begin{bmatrix} E^y \\ H^z \end{bmatrix}, \quad \vec{F}(U) = \begin{bmatrix} H^z \\ E^y \end{bmatrix}, \quad Q = \begin{bmatrix} \epsilon & 0 \\ 0 & \mu \end{bmatrix} \tag{3.8}$$

which are the field vector, the flux vector and the material properties matrix.

Chapter 4

SPATIAL DISCRETIZATION

In this chapter, the finite element methods utilized in this thesis are presented. In section 4.1, the spatial discretization and the weak Galerkin form of the governing equations are derived. In section 4.1.1, the numerical flux used in the discontinuous-Galerkin formulation is discussed, and the final form of the discrete equations is presented. Section 4.2 presents the Petrov-Galerkin formulation; section 4.2.1 shows the Streamline/upwind Petrov-Galerkin Stabilization as used for the Maxwell equations.

4.1 Discontinuous Galerkin Formulation

The discontinuous Galerkin method used here follows that from Hesthaven's and Warburton's book, *Nodal Discontinous Galerkin Methods* [8]. To begin, the domain is split into K elements, of length h^k. Using this, a linear mapping for the domain is obtained as

$$x^k(r) = x^k_- + \frac{r+1}{2}h^k \tag{4.1}$$

where r ranges from -1 to 1, and x^k_- is the physical location of the start of each element. Within each element, the field vector solution U may be expressed as a polynomial

$$\vec{U}^k(x,t) = \sum_{i=0}^{N} \vec{U}^k_i(t) l^k_i(x^k) \tag{4.2}$$

The polynomials l_i^k are considered herein to be the family of Lagrange polynomials. These polynomials may be derived using the following equation

$$l_i^k(x) = \prod_{j=0;j\neq i}^{N} \frac{x^k - x_j^k}{x_i^k - x_j^k}, \quad l_i^k(x_j^k) = \partial_{ij} \tag{4.3}$$

However, it is more convenient, both in the formulation and for mathematical reasons to define the polynomial basis as

$$\phi_n(x) = \frac{P_n(x)}{\sqrt{2n+1}} \tag{4.4}$$

in which $P_n(x)$ is the classical Legendre polynomial [12]. Using this, the solution may be represented as

$$U^k(x,t) = \sum_{i=0}^{N} \vec{U}_i^k(t) l_i^k(x^k) = \sum_{n=0}^{N} \vec{U}_n^k(t) \phi_n^k(x^k) \tag{4.5}$$

A matrix may be defined using the Legendre polynomials and the Legendre Gauss Lobatto quadrature points such that

$$\mathrm{V}_{ij} = P_j(x_i) \tag{4.6}$$

which is commonly known as the Vandermonde matrix. This matrix is critical in that it allows a transformation from nodal coefficients to modal values, which permits efficient integral evaluation at the Gauss Lobatto quadrature points [13].

Using this formulation, substituting $\vec{U}^k$ into the Maxwell equations from Chapter 3, we obtain

$$\int_{x_-^k}^{x_+^k} \left(Q \frac{\partial \vec{U}_N^k}{\partial t} + \frac{\partial F(\vec{U}_N^k)}{\partial x} \right) l_i^k(x) dx = 0 \tag{4.7}$$

where N is the number of quadrature points used in the discretization. x_-^k and x_+^k indicate the first and last points of each element. Using this and integrating once by parts, the weak Galerkin form becomes

$$\int_{x_-^k}^{x_+^k} \left(Q \frac{\partial \vec{U}_N^k}{\partial t} l_i^k(x) - \hat{n} \cdot F(\vec{U}_N^k) \frac{dl_i^k}{dx} \right) dx = \oint_{x_-^k}^{x_+^k} \hat{n} \cdot F(\vec{U}_N^k) l_i^k(x) dx \tag{4.8}$$

For the discontinuous Galerkin scheme, each element contains its own copy of the solution at the nodes. At the nodes that are shared by multiple elements, that is at the inter-element faces, the solution will be multi-valued. Thus the surface integral on the right hand side of equation 4.8 is not uniquely defined. Possible remedies of this issue could be to use a linear combination of the available data or to assume one value over the other based on physical considerations. However, the Maxwell equations in the current thesis are formulated as a hyperbolic system and as such requires stabilization. To this end, the flux in the surface integral of equation 4.8 is replaced with a numerical flux term, F^*, in order to provide sufficient dissipation for stability. Symbolically, the form the numerical flux may be written as

$$\hat{n} \cdot F^* = \hat{n} \cdot F^*(\vec{U}^-, \vec{U}^+) \tag{4.9}$$

where $\hat{n}$ is the unit normal vector to the face, and $\vec{U}^-$ and $\vec{U}^+$ are the values of the field vector in the two elements that share the connecting node. The weak Galerkin form with this flux then becomes

$$\int_{x_-^k}^{x_+^k} \left(Q \frac{\partial \vec{U}_N^k}{\partial t} + \frac{\partial F(\vec{U}_N^k)}{\partial x} \right) l_i^k(x) dx = \oint_{x_-^k}^{x_+^k} \hat{n} \cdot [F(\vec{U}^-) - F^*] l_i^k(x) dx \tag{4.10}$$

where $F(\vec{U}^-)$ is the flux of the previous element. Assuming that the material coefficients Q do not vary within elements, but may be different between elements, allows the global mass matrix and stiffness matrix to be evaluated as

$$\mathrm{M}_{ij}^{1d} = \int_{x_-^k}^{x_+^k} l_i^k(x) l_j^k(x) dx^k = \frac{h^k}{2} \int_{-1}^{1} l_i(r) l_j(r) dr \tag{4.11}$$

$$\mathrm{S}_{ij}^{1d} = \int_{x_-^k}^{x_+^k} l_i^k(x) \frac{dl_j^k(x)}{dx} dx^k = \frac{h^k}{2} \int_{-1}^{1} l_i(r) \frac{dl_j(r)}{dr} dr \tag{4.12}$$

Rather than calculating these integrals directly, it is simpler to use the Vandermonde matrix obtained above to derive them as

$$\mathrm{M}^{1d} = (\mathrm{V}^{-1})^T \mathrm{V}^{-1}, \quad \mathrm{S} = (\mathrm{V}^T)^{-1}\mathrm{W}\mathrm{V}^{-1}, \quad \mathrm{W}_{ij} = \int_{-1}^{1} \phi_i(r)\phi_j'(r)dr \tag{4.13}$$

Additionally, the properties of the orthogonal polynomials make the calculation of the W matrix trivial. These matrices will be used in conjunction with the numerical flux in the next section to obtain the final form of the equations used in the current work.

4.1.1 Numerical Flux and Material Properties

Though there are many options for the numerical flux, an upwind flux has been chosen here. An upwind flux takes advantage of the fact that with Maxwell's equations characteristic variables always determine which direction information is propagated in. This allows the numerical flux with a relatively simple formula

$$\hat{n} \cdot F^{\star} = \frac{1}{2}\begin{cases} \bar{Z}^{-1}(\hat{n_x}\{Z(U_2^+ + U_2^-)\} + (U_1^- - U_1^+)) \\ \bar{Y}^{-1}(\hat{n_x}\{Y(U_1^+ + U_1^-)\} + (U_2^- - U_2^+)) \end{cases} \tag{4.14}$$

Z and Y are the local impedance and conductance, respectively, and defined as

$$Z^{\pm} = \sqrt{\frac{\mu^{\pm}}{\epsilon^{\pm}}}, \quad Y^{\pm} = (Z^{\pm})^{-1} \tag{4.15}$$

Averaging the values from each element gives

$$\bar{Z} = \frac{Z^+ + Z^-}{2}, \quad \bar{Y} = \frac{Y^+ + Y^-}{2} \tag{4.16}$$

Finally, this flux term is substituted into the Galerkin form obtained in the previous section to yield the final equations used for the spatial discretization. For the sake of notational convenience, the vectors E_N^k and H_N^k are used in place of the previous notation of U_1 and U_2 to indicate the fields of element k

$$\begin{aligned}\epsilon^k \frac{h^k}{2} \mathrm{M}^{1d} \frac{dE_N^k}{dt} + \mathrm{S}H_N^k = e_0 &\left[\frac{1}{Z} \Big(-Z^+ (H_N^{k-} - H_N^{k+}) - (E_N^{k-} - E_N^{k+}) \Big) \right]_{x_-^k} \\ &+ e_N \left[\frac{1}{\bar{Z}} \Big(-Z^+ (H_N^{k-} - H_N^{k+}) - (E_N^{k-} - E_N^{k+})] \Big) \right]_{x_+^k}\end{aligned} \tag{4.17}$$

$$\begin{aligned}\mu^k \frac{h^k}{2} \mathrm{M}^{1d} \frac{dH_N^k}{dt} + \mathrm{S}E_N^k = e_0 &\left[\frac{1}{Y} \Big(-Y^+ (E_N^{k-} - E_N^{k+}) - (H_N^{k-} - H_N^{k+}) \Big) \right]_{x_-^k} \\ &+ e_N \left[\frac{1}{Y} \Big(-Y^+ (E_N^{k-} - E_N^{k+}) - (H_N^{k-} - H_N^{k+}) \Big) \right]_{x_+^k}\end{aligned} \tag{4.18}$$

Using this scheme, it is easy to obtain high order spatial accuracy. Hesthaven's and Warburton's paper provides exact details of the proof [9]. In this thesis, 6 quadrature points are used to obtain fifth order spatial accuracy. Since the highest order time integration method implemented in the current work is fourth order, fewer spatial elements can be used while investigating the temporal design order of accuracy. That is, in the best case scenario, the spatial discretization error is still an order of magnitude smaller than the temporal error, allowing us to analyze the temporal accuracy without altering the number of elements in the domain.

4.2 Petrov-Galerkin Scheme

For the Petrov-Galerkin scheme, a version of the FUNSAFE [11, 14–21] framework modified for the 1D Maxwell equations is used. Similar to the discontinuous Galerkin method, the Petrov-Galerkin approach is a weighted residual method and begins with the following form

$$\oiiint_{\Omega} [\phi] \left(\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \vec{F} \right) d\Omega = 0 \tag{4.19}$$

where ϕ is the weighting function. This weighting function is comprised of two parts. Similar to a standard Galerkin discretization the first part is the same basis functions used to represent the field vector over the element. Unlike the discontinuous Galerkin method, Petrov-Galerkin formulations are continuous across inter-element faces. That is, the field vector is singly valued at all nodes. The second contribution to the weighting function is a stabilizing term and serves essentially the same purpose as the numerical flux used in the discontinuous Galerkin approach. Therefore, in general, the weighting function for Petrov-Galerkin formulations may be written as

$$[\phi] = N_i[I] + [P] \tag{4.20}$$

where N_i represents the i^{th} Lagrangian basis function given previously in equation 4.3 for the element, [I] is the identity matrix with the dimension of the field vector, and [P] represents the perturbation to the test functions to provide stability. As will be presented in section 4.2.1, the Streamline/Upwind Petrov-Galerkin (SUPG) method is used for defining the stabilization in the current work.

On inserting the weighting function given in equation 4.20 into equation 4.19 and separating terms yields

$$\oiiint_{\Omega} N_i[I]\left(Q\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \vec{F}\right)d\Omega + \oiiint_{\Omega}[P]\left(Q\frac{\partial \vec{U}}{\partial t} + \nabla \cdot \vec{F}\right)d\Omega = 0 \tag{4.21}$$

Performing integration by parts with respect to the spatial gradient in the first term produces the weak statement of the governing equations as

$$\oiiint_{\Omega}\left(N_i Q\frac{\partial \vec{U}}{\partial t} - F \cdot \nabla N_i\right)d\Omega + \oiiint_{\Omega}[P]Q\left(\frac{\partial \vec{U}}{\partial t} + \nabla \cdot F\right)d\Omega + \oiint_{\Gamma} N_i F \cdot \hat{n}\partial\Gamma = 0 \tag{4.22}$$

Since the Petrov-Galerkin method is continuous between elements, the surface integral vanishes for all interior nodes. That is, the surface integral only contributes to faces on the physical boundary, and is used to strongly enforce boundary conditions involving specification of fluxes.

4.2.1 Stabilization

In the current work, stabilization of the Maxwell equations is accomplished using the SUPG approach. This method provides dissipation along preferential directions to eliminate odd-even point decoupling that often occurs with the standard Galerkin formulation. The perturbation to the test function may be evaluated as

$$[P] = \left(\frac{\partial N}{\partial x}[A]\right)[\tau] \tag{4.23}$$

where [A] is the flux Jacobian matrix, and $[\tau]$ can be derived as follows

$$[\tau]^{-1} = \sum_{i=1}^{n} \left| \frac{\partial N_i}{\partial x}[A] \right| \tag{4.24}$$

The flux Jacobian matrix is simply

$$[A] = \frac{\partial F}{\partial U} = \begin{bmatrix} 0 & \frac{1}{\mu} \\ \frac{1}{\epsilon} & 0 \end{bmatrix} \tag{4.25}$$

In order to evaluate the absolute value of the matrix on the right hand side of equation 4.24, the eigensystem for this matrix must be defined. To this end, the eigenvalues of this system are

$$[\Lambda] = \begin{bmatrix} \frac{N_x}{\sqrt{\mu\epsilon}} & 0 \\ 0 & \frac{-N_x}{\sqrt{\mu\epsilon}} \end{bmatrix} \tag{4.26}$$

And the right eigenvectors are found to be

$$[B] = \begin{bmatrix} \sqrt{\frac{\epsilon}{\mu}} & -\sqrt{\frac{\epsilon}{\mu}} \\ 1 & 1 \end{bmatrix} \tag{4.27}$$

Hence, the absolute value of the matrix may be easily determined using

$$\left|\frac{\partial N}{\partial x}[A]\right| = [B][|\Lambda|][B]^{-1} \tag{4.28}$$

where $[B]^{-1}$ represents the inverse of the right eigenvector matrix.

Chapter 5

TEMPORAL INTEGRATION

In this chapter, the temporal integration schemes used in the current work are presented. In section 5.1 the Jacobian matrix and the Newton methodology used for the implicit schemes is discussed. In section 5.2, the first and second order backward differentiation formulas are introduced. Section 5.3 presents a modification to the basic backward differentiation formulas and extends them to multi-stage methods in order to obtain A-stability for third and fourth order accuracy. Finally, in section 5.4 implicit Runge-Kutta methods are discussed.

5.1 Jacobian Matrix and Newton Methodology

For each of the implicit methods, a Newton's method was used for obtaining estimates at the $n-th$ timestep. Newton's method is one of the most efficient ways to find the roots of a real-valued function. In its most general form, it can be expressed as

$$x^{n+1} = x^n - \frac{f(x^n)}{f'(x^n)} \tag{5.1}$$

or in delta form as

$$x^{n+1} = x^n + \Delta x, \qquad \text{where} \qquad \Delta x = -\frac{f(x^n)}{f'(x^n)} \tag{5.2}$$

To apply Newton's method to the Maxwell equations, an unsteady residual is developed such that

$$R^* = M\frac{d\vec{U}}{dt} + R = 0 \tag{5.3}$$

where R is the spatial residual.

For the Maxwell equations, since the derivative $f'(x_n)$ must be evalued with respect to both the residual from U_1 and U_2 fields, it becomes a 2×2 matrix that takes the following form

$$J = \begin{bmatrix} \frac{\partial R_1^*(U^{n+1})}{\partial U_1} & \frac{\partial R_1^*(U^{n+1})}{\partial U_2}) \\ \frac{\partial R_2^*(U^{n+1})}{\partial U_1} & \frac{\partial R_2^*(U^{n+1})}{\partial U_2}) \end{bmatrix} \tag{5.4}$$

and thus the Newton formulation becomes

$$\Delta \vec{U} = -J^{-1} R^*(\vec{U}_n) \tag{5.5}$$

which is solved using an LU decomposition of J and forward-backward substitution. The Newton's method is iterated until the unsteady residual R^* is driven to zero.

5.2 Backward Differentiation Formula

The Backward Differentiation Formula (BDF) is one of the most well known and straightforward implicit methods to implement. However, while easy to code and fast to evaluate, the highest A-stable BDF scheme is only of order 2. As such, only the BDF 1 and 2 are used here, and they are defined as follows

$$R^* = \frac{\mathrm{M}}{\Delta t}\left(U^{n+1} - U^n\right) + R(t_{n+1}, U_{n+1}) = 0 \tag{5.6}$$

and

$$R^* = \frac{\mathrm{M}}{\Delta t}\left(U^{n+1} - \frac{4}{3}U^n + \frac{1}{3}U^{n-1}\right) + R(t_{n+1}, U_{n+1}) = 0 \tag{5.7}$$

Fortunately, with some adjustments, it is possible to use these simple formulas as a basis for higher order methods.

5.3 Modified Extended Backward Differentiation Formula

Linear multistep methods like the Backwards Differentiation Formula, are only A-stable if their order is at most two. However, due to their ease of implementation, a great deal of effort has gone into developing ways to extend them to higher order. The modified extended backward differentiation formula achieves higher order accuracy by splitting the integration over a timestep into several stages. Though beyond the scope of this thesis, Modified Extended Backward Differentiation Formula (MEBDF) schemes can be shown to be A-stable up to fourth order [22]. For the $k-th$ order MEBDF implemented here, the first stage is a BDF of order $k-1$ to obtain a value for the $n+1$ timestep as

$$R^{*} = \frac{\mathrm{M}}{\Delta t}\left(\hat{\gamma}_0 \bar{U}^{n+1} + \sum_{j=1}^{k} \hat{\gamma}_j U^{n-j}\right) + \hat{\beta}_1 R(U^{\bar{n}+1}) = 0 \tag{5.8}$$

where $\hat{\gamma}_j$ and $\hat{\beta}_1$ are constant coefficients given in table 5.1. The second stage uses the solution from the first stage to perform another BDF of order $k-1$ to obtain the solution at a superfuture step as

$$R^{*} = \frac{\mathrm{M}}{\Delta t}\left(\hat{\gamma}_0 \bar{U}^{n+2} + \hat{\gamma}_1 \bar{U}^{n+1} + \sum_{j=2}^{k} \hat{\gamma}_j U^{n-j}\right) + \hat{\beta}_1 R(U^{\bar{n}+2}) = 0 \tag{5.9}$$

Finally, in the third stage, the values for $\bar{U}^{n+1}$ and $\bar{U}^{n+2}$ are used to obtain a $k-th$ order final value at $n+1$.

$$R^{*} = \frac{\mathrm{M}}{\Delta t}\left(\bar{\gamma}_0 U^{n+1} + \sum_{j=1}^{k} \bar{\gamma}_j U^{n-j}\right) + \hat{\beta}_1 R(U^{n+1}) + (\bar{\beta}_1 - \hat{\beta}_1) R(\bar{U}^{n+1}) + \bar{\beta}_2 R(U^{\bar{n}+2}) = 0 \tag{5.10}$$

where $\bar{\gamma}_j$, $\bar{\beta}_1$ and $\bar{\beta}_2$ are constant coefficients given in table 5.2. Though the three stages are very computationally intensive as written, two simplifications can dramatically reduce computational cost [23]. First, after the initial time-step, the superfuture guess from stage 2 can be used as a starting guess for stage 1 of the next timestep. Secondly, the result of stage 1 can be used as an initial guess for stage 3. In most cases, this should cause the Newton method used to solve stages 1 and 3 to converge extremely quickly, and thus only stage 2 should involve the full computational cost typically associated with performing a full BDF. While certainly more computationally intensive than using a simple BDF, the ability to go beyond second order accuracy and maintain stability more than makes up for the added computational cost. One of the disadvantages of MEBDF schemes is that beyond order 2 they are not self-starting, meaning that for a k-th order MEBDF it is necessary to run $k-2$ lower order MEBDF steps at the start of the computation. This may adversely effect the final accuracy of the scheme, depending on how large a timestep is used. There are several ways to mitigate the impact of non-self starting schemes. One possible way to resolve this is to utilize variable time steps, and to split the initial timesteps into multiple smaller timesteps, small enough to retain the desired order of accuracy throughout the computation. Another possibility is to utilize a higher order self-starting scheme, such as an implicit Runge-Kutta, during the required initial timesteps. Alternatively, an algorithm has been developed by Tirana [24] that provides the required number of starting conditions for any multistep method of any order. The regular BDF coefficients used for the first two stages are given in table 5.1, while the MEBDF specific coefficients used for the third stage are given in table 5.2.

Table 5.1 Coefficients for the BDF schemes

Order	$\hat{\gamma}_0$	$\hat{\gamma}_1$	$\hat{\gamma}_2$	$\hat{\gamma}_3$	$\hat{\beta}_1$
1	1	-1			1
2	1	$-\frac{4}{3}$	$\frac{1}{3}$		$\frac{2}{3}$
3	1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$	$\frac{6}{11}$

Table 5.2 Coefficients for the MEBDF schemes

Order	$\bar{\gamma}_0$	$\bar{\gamma}_1$	$\bar{\gamma}_2$	$\bar{\gamma}_3$	$\bar{\beta}_1$	$\bar{\beta}_2$
2	1	-1			$\frac{3}{2}$	$-\frac{1}{2}$
3	1	$-\frac{28}{23}$	$\frac{5}{23}$		$\frac{22}{23}$	-$\frac{4}{23}$
4	1	$-\frac{279}{197}$	$\frac{99}{197}$	$-\frac{17}{197}$	$\frac{150}{197}$	$-\frac{18}{197}$

5.4 Implicit Runge-Kutta

An alternate method for obtaining higher order temporal accuracy is through the very popular Runge-Kutta schemes. One of the most efficient methods that is A-stable for high order accuracy is the explicit first stage single diagonal coefficient Runge-Kutta scheme (ESDIRK); it is also easy to implement. One of the primary advantages of this scheme is that the Jacobian matrix needs to only be calculated once, as the coefficient in front of the new solution calculated at each stage is the same. However, as compared to the MEBDF schemes, there is considerably higher computational cost to achieve the same order of accuracy. For example, while the MEBDF scheme only needs three stages to achieve fourth order accuracy, the ESDIRK scheme requires six though only fiveof those are implicit. For an m-stage ESDIRK scheme, the algorithm may be written as

[25]

$$
\begin{aligned}
&\text{(i)} \quad \vec{U}^1 = \vec{U}^n \qquad\qquad (5.11)\\
&\text{(ii) For} \quad s = 2, \ldots m\\
&\qquad R^* = \frac{\vec{U}^s - \vec{U}^n}{\Delta t} + \sum_{j=1}^{s} a_{sj} \mathrm{M}^{-1} R(\vec{U}^j) = 0\\
&\text{(iii)} \quad \vec{U}^{n+1} = \vec{U}^m
\end{aligned}
$$

where R and R^* represent the spatial residual and the unsteady residual, respectively. For a second order scheme, this will require 1 explicit stage and 1 implicit stage, a third order scheme will require 1 explicit and 3 implicit, and so on. The Butcher tables for the second, third and fourth order schemes are given in tables 5.3-5.5. Of note, the second order scheme is commonly known as the Crank-Nicholson scheme, and it is an extremely popular method that is easy to implement. In the current work, the aforementioned ESDIRK scheme is used for the discontinuous Galerkin method and could be utilized for the Petrov-Galerkin formulation, when linear material models are employed. For nonlienar problems, the mass matrix for the Petrov-Galerkin method is a function of the field variables, and implementation of an ESDIRK scheme would require the solution of an additional linear system. Therefore, the Petrov-Galerkin method uses an SDIRK scheme that has been reformulated to accomodate the possibility of a nonlinear mass matrix as

$$
\begin{aligned}
&\text{(i)} \quad \vec{U}^0 = \vec{U}^n \qquad\qquad (5.12)\\
&\text{(ii) For} \quad s = 1, \ldots m\\
&\qquad R^* = \frac{\mathrm{M}}{\Delta t}\left(\sum_{j=0}^{s} \hat{c}_j \vec{U}^s\right) + R(\vec{U}^s) = 0\\
&\text{(iii)} \quad \vec{U}^{n+1} = \vec{U}^m
\end{aligned}
$$

Additional details concerning this scheme may be found in [14]. The stage coefficients $\hat{c}_j$ are given in table 5.6 for the five stage, fourth order SDIRK scheme. 6

Table 5.3 Butcher table for ESDIRK2 scheme

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

Table 5.4 Butcher table for ESDIRK3 scheme

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{4055673282236}$	$a_{22} = \frac{4055673282236}{2027836641118}$	0	0
$\frac{3}{5}$	$\frac{2746238789719}{10658868560708}$	$-\frac{640167445237}{6845629431997}$	$a_{33} = a22$	0
1	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$a_{44} = a22$
	$\frac{1471266399579}{7840856788654}$	$-\frac{4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$c_6 = a22$

Table 5.5 Butcher table for ESDIRK4 scheme

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$a_{22} = \frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$a_{33} = a22$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$a_{44} = a22$	0	0
$\frac{17}{20}$	$\frac{15267082809}{1553762565600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$a_{55} = a22$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$a_{66} = a22$
	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$c_6 = a22$

Table 5.6 Recast coefficients for the SDIRK4 scheme

s	$Nodes$	$\hat{c}_0$	$\hat{c}_1$	$\hat{c}_2$	$\hat{c}_3$	$\hat{c}_4$	$\hat{c}_5$
1	$\frac{1}{4}$	-4	4	0	0	0	0
2	$\frac{3}{4}$	4	-8	4	0	0	0
3	$\frac{11}{20}$	$\frac{52}{25}$	$-\frac{168}{25}$	$\frac{16}{25}$	4	0	0
4	$\frac{1}{2}$	$\frac{16}{17}$	$-\frac{89}{17}$	$\frac{25}{34}$	$-\frac{15}{34}$	4	0
5	1	$-\frac{28}{3}$	$\frac{37}{3}$	$\frac{103}{6}$	$-\frac{275}{2}$	$\frac{340}{3}$	4

Chapter 6

RESULTS

6.1 Problem Examined

For the purpose of verifying that each method achieved the expected temporal convergence, a simple Gaussian pulse was inserted into the domain centered around $x = 0.5$

$$E_0(x) = e^{-\sigma(x-x_0)^2} \tag{6.1}$$

where σ is the standard deviation of the Gaussian distribution, and here is set to be 10. The analytical solution at any given time is given by

$$E(t) = e^{-\sigma(x-x_0-\lambda t)^2} \tag{6.2}$$

where λ is the wave speed. For the 1D Maxwell equations, the wave speed is simply the speed of light in the medium. The speed of light in any given medium can be easily found using the following definition

$$c = \sqrt{\frac{1}{\epsilon\mu}} \tag{6.3}$$

For this problem, it was assumed that the wave propagated through a vacuum, where $\epsilon = \epsilon_0$ and $\mu = \mu_0$. For the sake of simplicity, ϵ_0 and μ_0 were both non-dimensionalized to be 1, as a result of which c was also equal to 1. Another advantage to these conditions is that for a wave propagating in the positive x-direction, the magnetic field will be equal in both sign and magnitude to the electric field defined in equation 6.2. Perfect electrical conductor, or PEC, boundary conditions are used, and may be implemented as

$$E^y(0,t) = E^y(L,t) = 0 \tag{6.4}$$

For all cases examined, the total amount of non-dimensional time for the simulation was 4. Figure 6.1 shows the initial and final states of the electric field for the problem examined. Figures 6.2-6.5 show snapshots of the electric field at the final time for various timesteps, with the number of timesteps increased by a factor of 4. Qualitatively, these snapshots show the expected results, namely that higher order methods converge to the exact solution much more quickly than the lower order methods. In particular, the IRK4 method can be readily seen to be the most accurate, becoming nearly indistinguishable from the analytical solution for a non-dimensional timestep as large as 0.25. While slightly different in magnitude, both the Petrov-Galerkin and discontinuous Galerkin schemes had very similar shapes, and thus figures 6.2-6.5 contain results only from the discontinuous Galerkin approach.

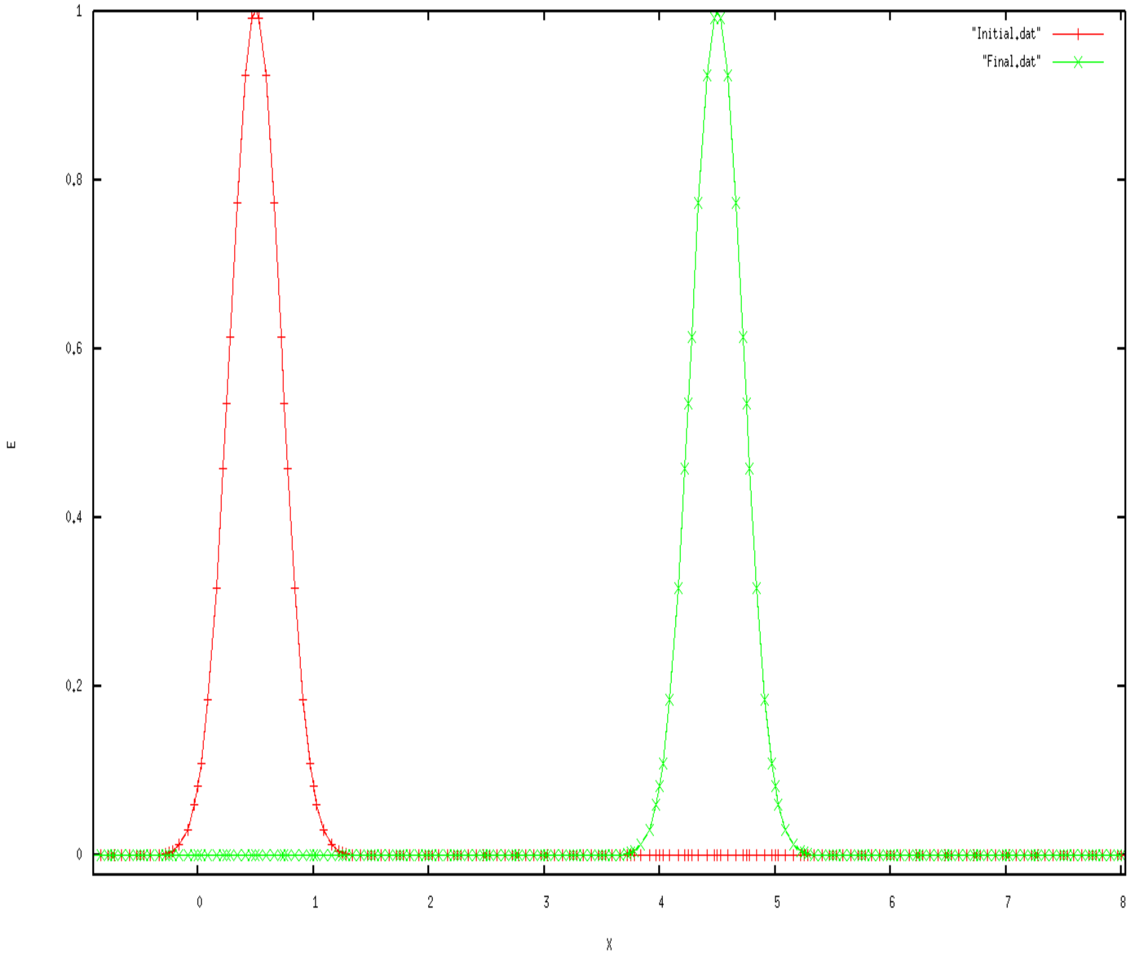

Figure 6.1 Initial and final exact solutions

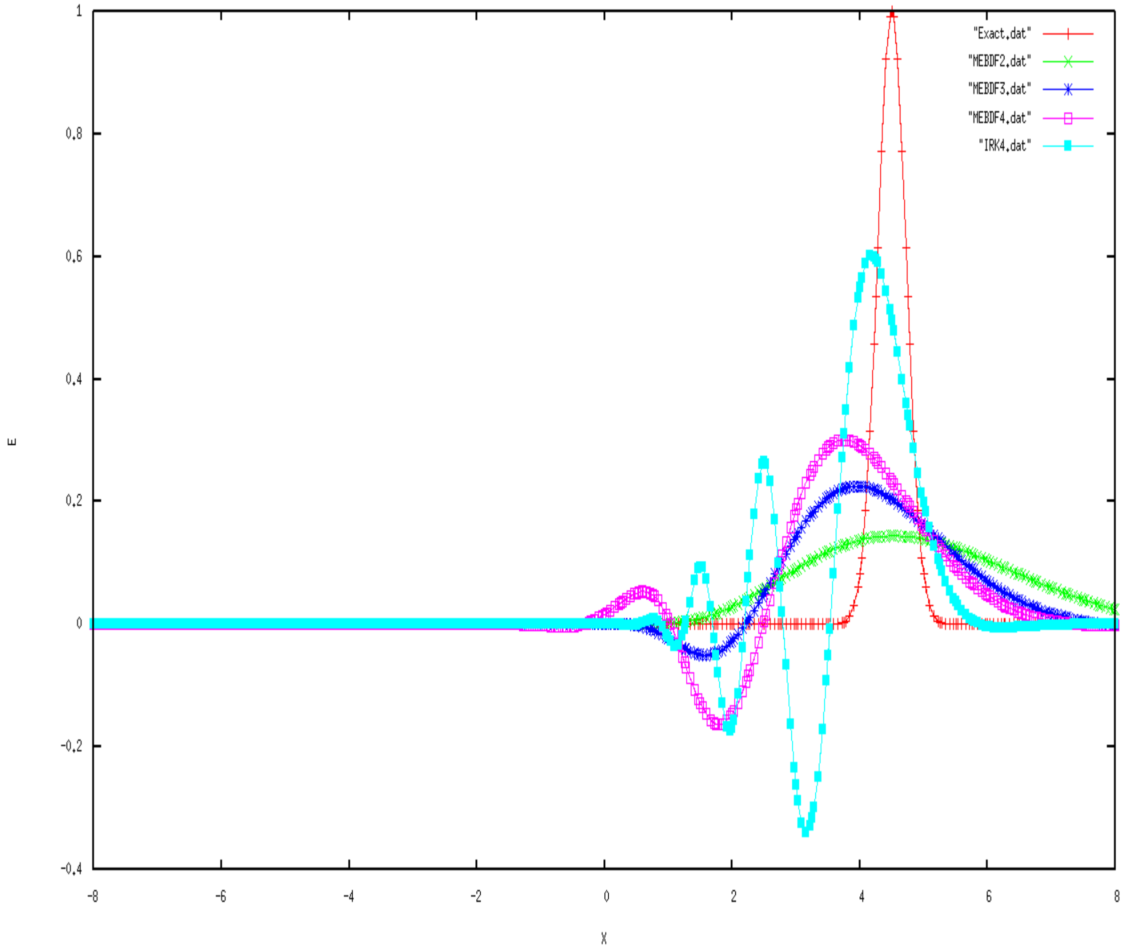

Figure 6.2 Electric field for discontinuous Galerkin at final time using $\Delta t = 1.0$

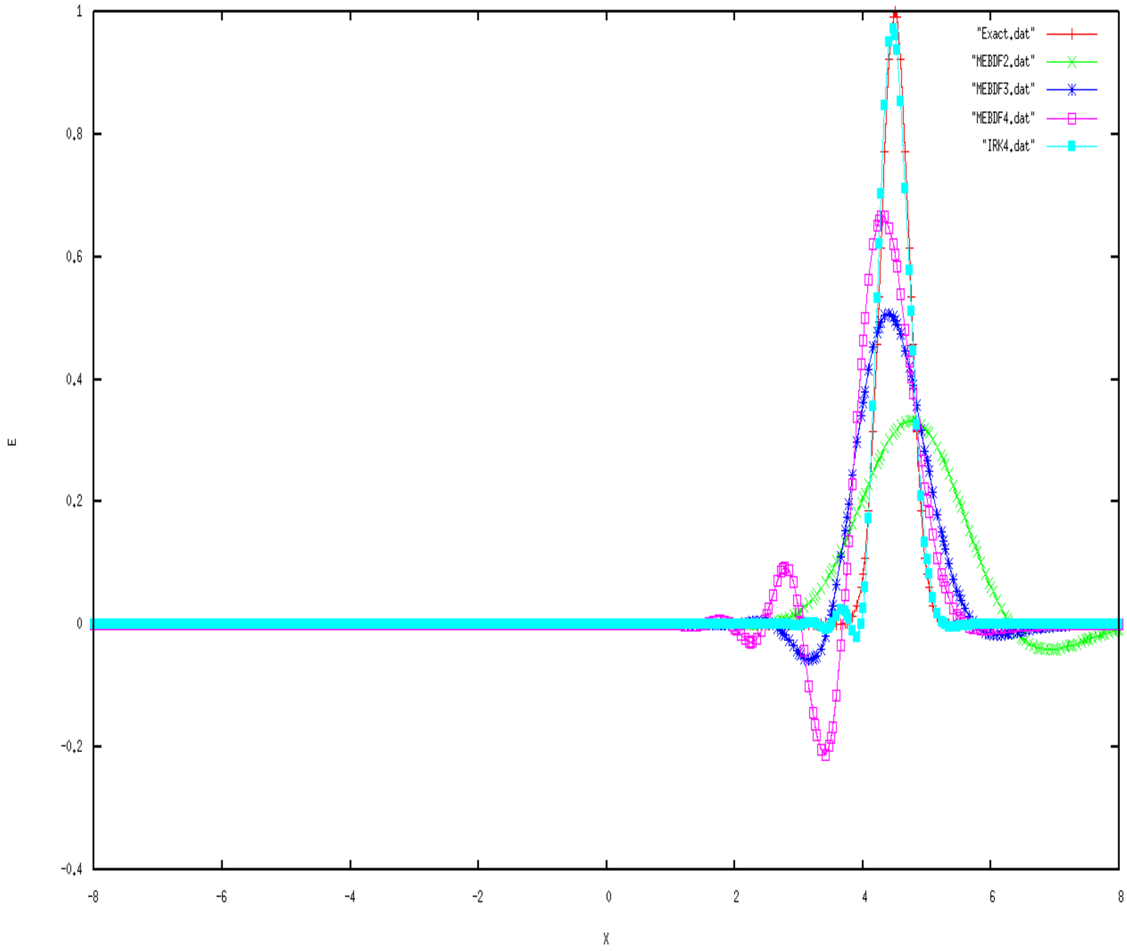

Figure 6.3 Electric field for discontinuous Galerkin at final time using $\Delta t = 0.25$

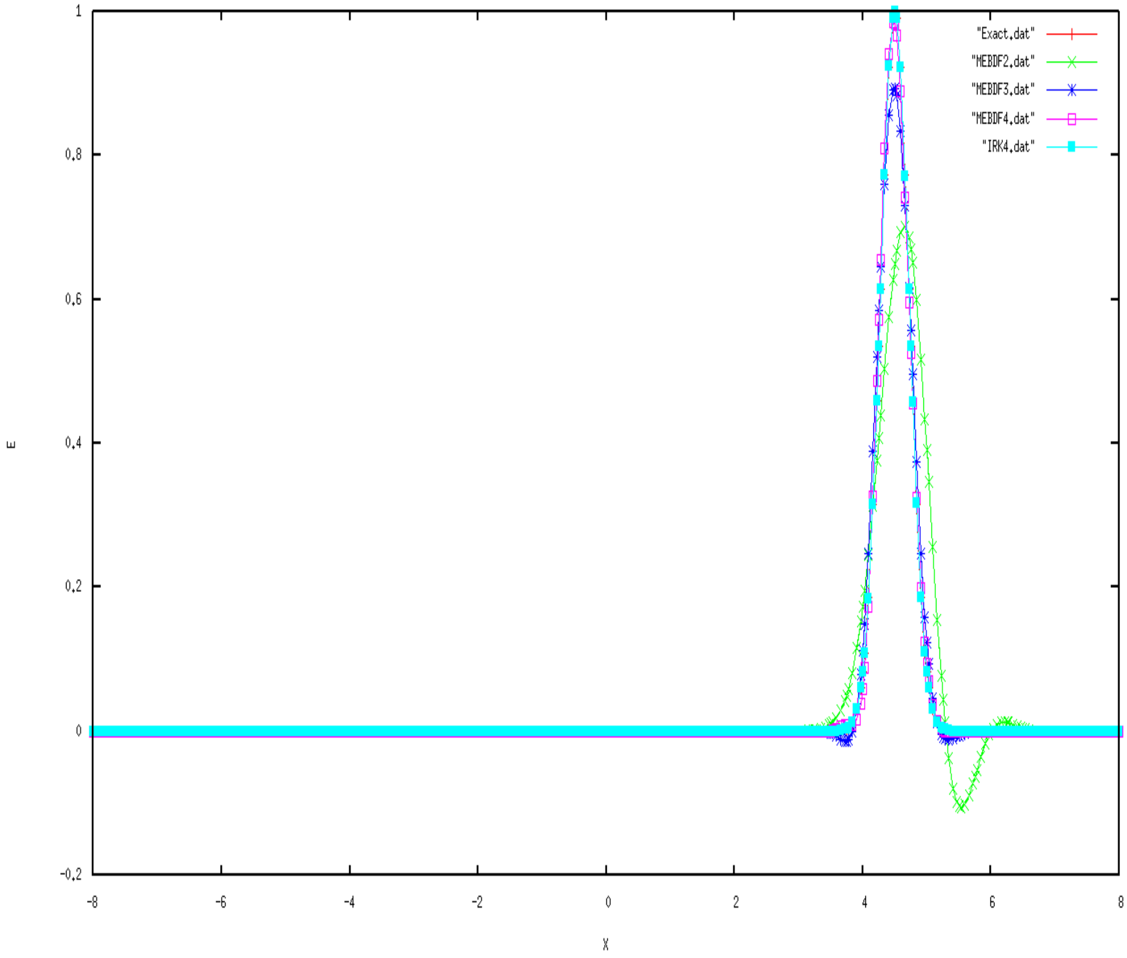

Figure 6.4 Electric field for discontinuous Galerkin at final time using $\Delta t = 0.0625$

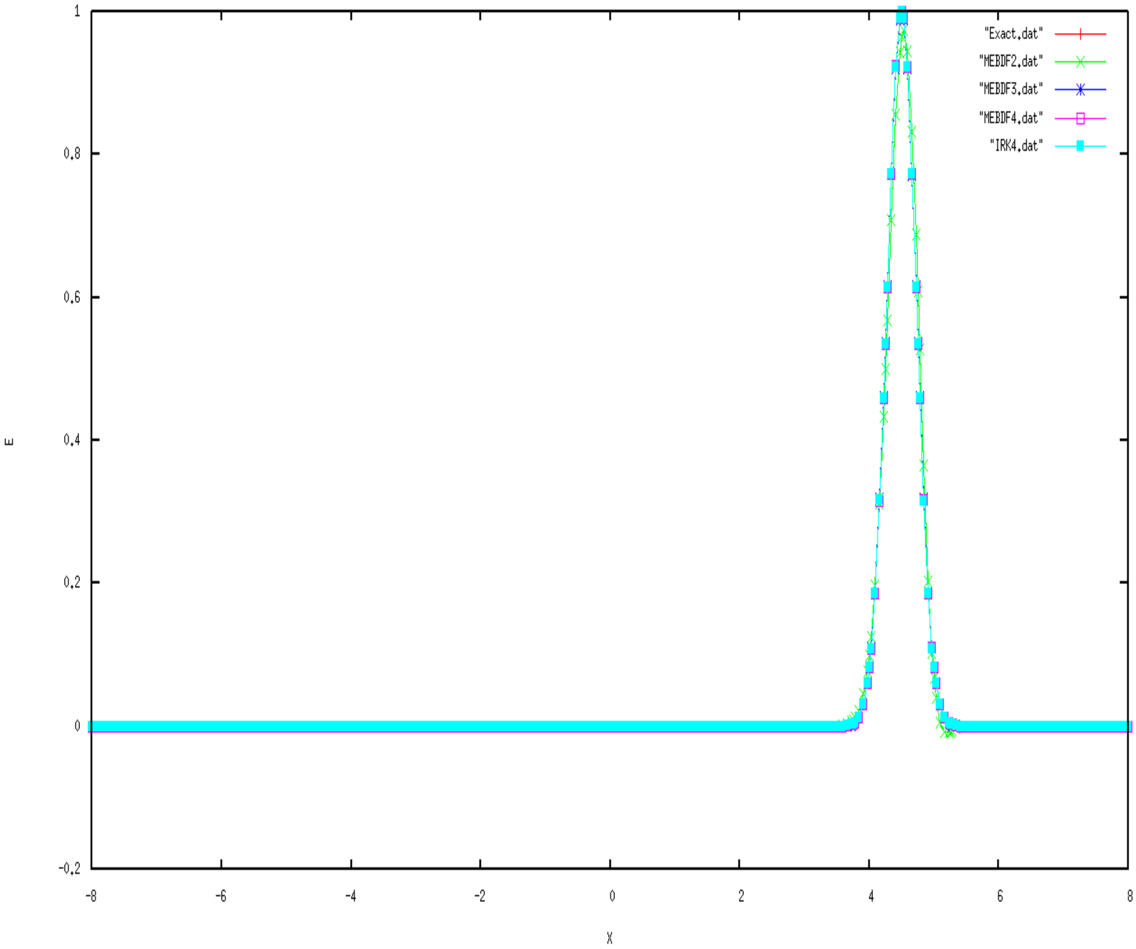

Figure 6.5 Electric field for discontinuous Galerkin at final time using $\Delta t = 0.015625$

6.2 Convergence

For all methods, the design order of accuracy for each time integration scheme was obtained. The error was calculated as the difference between the exact solution and the numerical solution, averaged across all nodes as follows

$$Error = \frac{1}{n}\sum_{j=1}^{n}(Uexact_j - U_j) \tag{6.5}$$

where n is the number of nodes. Tables 6.1-6.8 display the exact results calculated, as well as the observed order of accuracy, given by

$$order = \frac{log(error_n) - log(error_{n-1})}{log(\Delta t_n) - log(\Delta t_{n-1})} \tag{6.6}$$

where figures 6.6 and 6.7 provide a visual display of the results. Without exception, the IRK methods provided much higher initial accuracy than the same order MEBDF methods. Though not rigorously compared, the Petrov-Galerkin approach yielded errors approximately half that of running a simulation with the same parameters with the discontinuous Galerkin method.

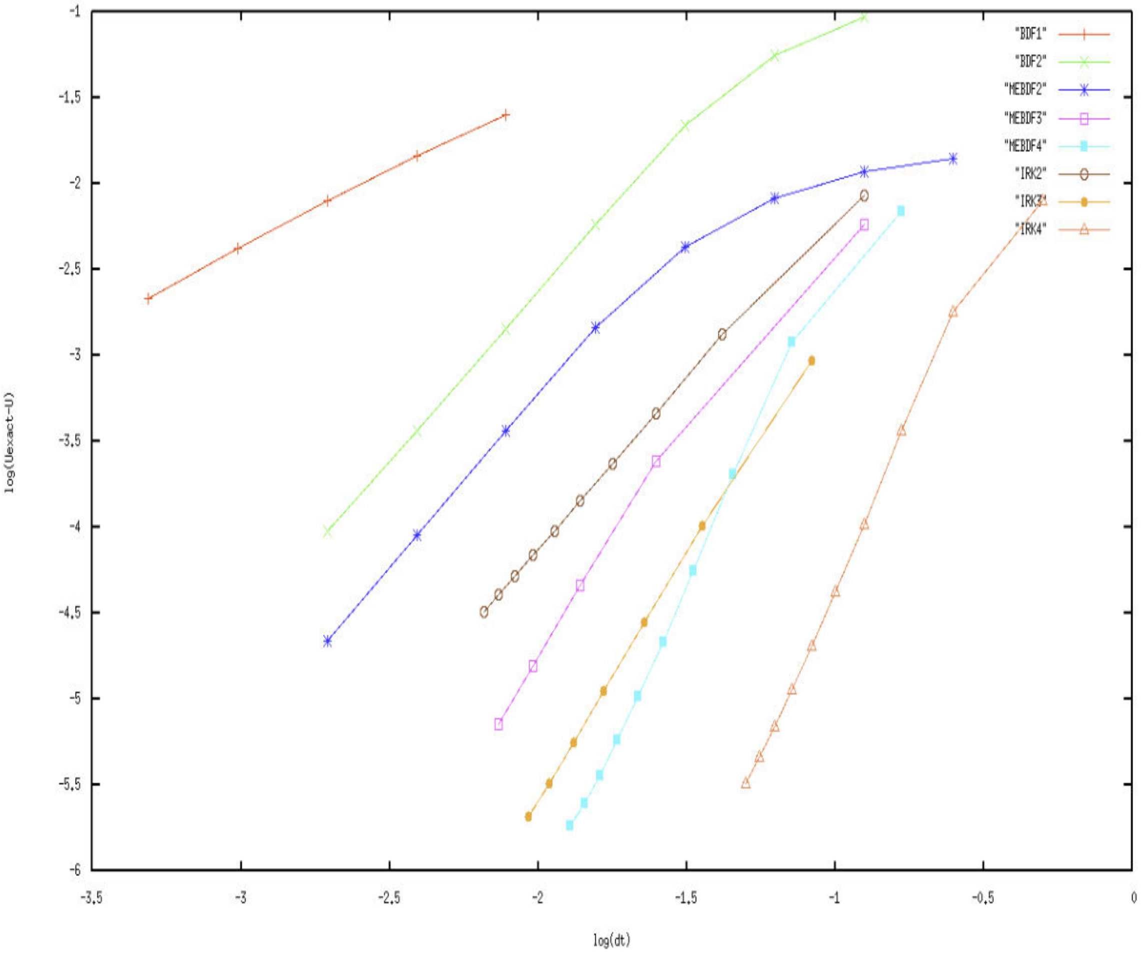

Figure 6.6 Discontinuous Galerkin temporal convergence

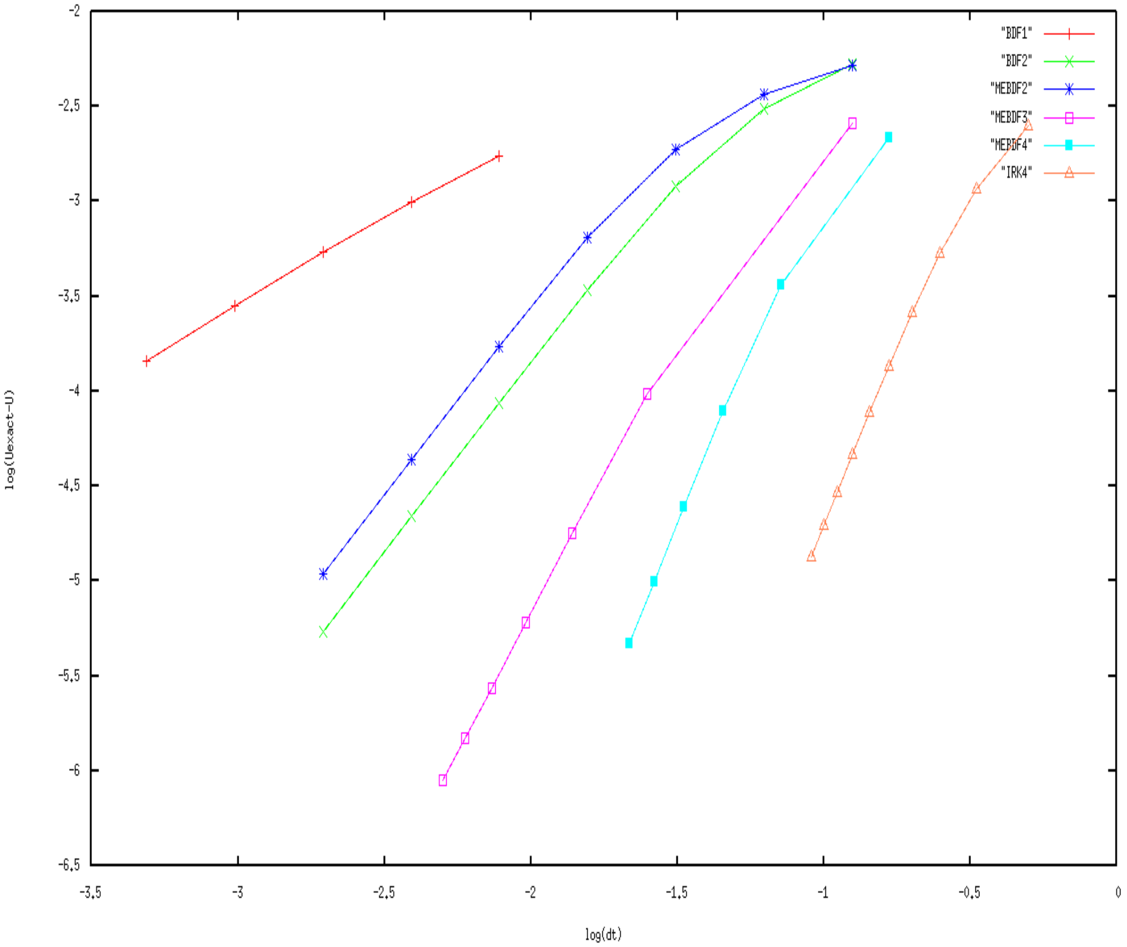

Figure 6.7 Petrov-Galerkin temporal convergence

Table 6.1 Error and observed order of accuracy using BDF1

Δt	nt	DG Error	DG Order	PG Error	PG Order
0.00390625	1024	0.0146128	0.777575	0.000996424	0.792116
0.00195312	2048	0.00800049	0.869077	0.000541191	0.880622
0.000976562	4096	0.0042057	0.927742	0.000282991	0.935382
0.000488281	8192	0.00215908	0.961931	0.000144848	0.966212

Table 6.2 Error and observed order of accuracy using BDF2

Δt	nt	DG Error	DG Order	PG Error	PG Order
0.0625	64	0.0553339	0.749343	0.0052833382	0.772490
0.03125	128	0.0217992	1.34389	0.0011924509	1.37502
0.015625	256	0.00582276	1.9045	0.0003383263	1.81744
0.0078125	512	0.00143943	2.01621	8.67776E-05	1.96301
0.00390625	1024	0.000362872	1.98796	0.000021794	1.99338
0.00195312	2048	9.41399E-05	1.94658	5.44023E-06	2.00219

Table 6.3 Error and observed order of accuracy using MEBDF2

Δt	nt	DG Error	DG Order	PG Error	PG Order
0.0625	64	0.00826148	0.500536	0.00367541	0.50204
0.03125	128	0.00426627	0.953424	0.00188741	0.96149
0.015625	256	0.0014573	1.54967	0.000642218	1.55527
0.0078125	512	0.000364043	2.00112	0.000172385	1.89742
0.00390625	1024	8.93368E-05	2.02678	4.36261E-05	1.98237
0.00195312	2048	0.00002187	2.0303	1.0946E-05	1.9947

Table 6.4 Error and observed order of accuracy using MEBDF3

Δt	nt	DG Error	DG Order	PG Error	PG Order
0.025	160	0.000243223	1.97076	9.67783E-05	2.87169
0.0138889	288	4.61724E-05	2.82687	1.78942E-05	2.94829
0.00961538	416	1.56673E-05	2.93917	6.05157E-06	2.94931
0.00735294	544	7.0921E-06	2.95454	2.74319E-06	2.91944
0.00595238	672	0.000003826	2.92069	1.48026E-06	2.85775

Table 6.5 Error and observed order of accuracy using MEBDF4

Δt	nt	DG Error	DG Order	PG Error	PG Order
0.0714286	56	0.00120335	2.05635	3.61523E-04	2.11134
0.0454545	88	0.000205264	3.91289	7.88339E-05	3.36954
0.0333333	120	5.56097E-05	4.21061	2.46124E-05	3.75325
0.0263158	152	2.15509E-05	4.01009	9.85439E-06	3.87216
0.0217391	184	1.03305E-05	3.84871	4.66000E-06	3.92090

Table 6.6 Error and observed order of accuracy using IRK2

Δt	nt	DG Error	DG Order
0.0416667	96	0.00132734	1.68832
0.025	160	0.000462324	2.06463
0.0178571	224	0.000234897	2.01241
0.0138889	288	0.000142195	1.99729
0.0113636	352	9.53618E-05	1.99093
0.00961538	416	6.84376E-05	1.98592
0.00833333	480	5.15464E-05	1.98071
0.00735294	544	4.02575E-05	1.97491
0.00657895	608	3.23417E-05	1.96842

Table 6.7 Error and observed order of accuracy using IRK3

Δt	nt	DG Error	DG Order
0.0357143	112	0.000102269	2.61636
0.0227273	176	2.77434E-05	2.8864
0.0166667	240	1.11418E-05	2.94143
0.0131579	304	5.56154E-06	2.93933
0.0108696	368	3.20521E-06	2.88449

Table 6.8 Error and observed order of accuracy using IRK4

Δt	nt	DG Error	DG Order	PG Error	PG Order
0.333333	12	0.00371461	1.86609	0.001158915	1.8893095896
0.25	16	0.00179837	2.5215	0.000533022	2.6997691886
0.2	20	0.000803316	3.61153	0.000259158	3.2316575008
0.166667	24	0.000361817	4.37474	0.000136096	3.5326247721
0.142857	28	0.000183595	4.40092	7.69737E-05	3.6970509575
0.125	32	0.000104207	4.24134	4.63989E-05	3.790780625
0.111111	36	6.42222E-05	4.10952	2.94900E-05	3.847940665
0.1	40	4.20331E-05	4.02325	1.95842E-05	3.8850419785
0.0909091	44	2.88021E-05	3.96611	1.34910E-05	3.9103585182

6.3 Dispersion and Dissipation Error

Though not rigorously derived here, Takacs [26] presents a useful method for quantitatively determining the dispersion and dissipation error. To accomplish this, three different measures of error: the total error, the dissipation error, and the dispersion error are derived. The total error is simply the mean square error, defined as

$$E_{TOT} = \frac{1}{n}\sum_{j=1}^{n}(Uexact_j - U_j)^2 \tag{6.7}$$

where n is the number of nodes and $Uexact$ represents the analytical solution at the nodes. This can be rewritten as

$$E_{TOT} = \frac{1}{n}\sum_{j=1}^{n}(Uexact_j - U_j)^2 = \sigma^2(Uexact - U) + (\bar{Uexact} - \bar{U})^2 \tag{6.8}$$

where $\sigma^2(\cdot)$ represents the variance of the quantity in parenthesis, and the overbar indicates the mean value. It can be shown that the above is equivalent to

$$E_{TOT} = \sigma^2(Uexact) + \sigma^2(U) - 2\rho\sigma(Uexact)\sigma(U) + (\bar{Uexact} + \bar{U})^2 \tag{6.9}$$

where ρ is the correlation coefficient between $Uexact$ and U. If $Uexact$ and U are exactly correlated, then the only error can be dissipation error, and so thus the dissipation error may be defined as

$$E_{DISS} = [\sigma(Uexact) - \sigma(U)]^2 + (\bar{Uexact} - \bar{U})^2 \tag{6.10}$$

If ρ is not equal to 1, then the additional error must be due to dispersion, and the dispersion error is defined as

$$E_{DISP} = 2(1-\rho)\sigma(Uexact)\sigma(U) \tag{6.11}$$

Using these definitions, it is possible to quantify the dissipation and dispersion errors associated with all the methods analyzed previously. The results of applying these definitions can be seen in tables 6.9-6.13. The most significant result is that for virtually all methods, the dissipation errors converge much faster than the dispersion errors. At large timesteps, the dissipation error and dispersion error were of similar magnitudes for all methods except the ESDIRK2 and ESDIRK4 methods, which had extremely small dissipation errors regardless of timestep size.

Table 6.9 Dissipation and dispersion errors for $\Delta t = 1.0$

Method	DG Diss.	DG Disp.	PG Diss.	PG Disp.
BDF1	0.0119933	0.00818326	0.0119401	0.00815046
BDF2	0.00510713	0.0170544	0.00508071	0.0169928
MEBDF2	0.0105264	0.00810161	0.0104833	0.00805841
MEBDF3	0.00658416	0.0105486	0.00655191	0.0105090
MEBDF4	0.00327579	0.0157313	0.00325806	0.0156713
IRK2	1.80625E-08	0.0110526	NA	NA
IRK3	0.00330606	0.0354245	NA	NA
IRK4	4.23969E-05	0.0144009	4.14922E-05	0.0143380

Table 6.10 Dissipation and dispersion errors for $\Delta t = 0.25$

Method	DG Diss.	DG Disp.	PG Diss.	PG Disp.
BDF1	0.00772726	0.00765277	0.00768940	0.00762421
BDF2	0.00140109	0.0135509	0.00139193	0.0134950
MEBDF2	0.00405772	0.00871085	0.00403758	0.008671812
MEBDF3	0.00157594	0.00529679	0.00156592	0.00526706
MEBDF4	0.000231638	0.00655172	0.000229053	0.006514731
IRK2	1.23422E-09	0.0154975	NA	NA
IRK3	0.000336503	0.00222856	NA	NA
IRK4	1.26892E-08	0.000206973	5.63435E-09	0.000181814

Table 6.11 Dissipation and dispersion errors for $\Delta t = 0.015625$

Method	DG Diss.	DG Disp.	PG Diss.	PG Disp.
BDF1	0.00348317	0.00494589	0.00346432	0.00492014
BDF2	0.000038153	0.00306184	3.72970E-05	0.00303339
MEBDF2	0.000355564	0.00401256	0.000352132	0.00398416
MEBDF3	3.19741E-05	0.000236579	3.11964E-05	0.000223189
MEBDF4	6.64569E-08	0.000037022	3.16186E-08	3.62087E-05
IRK2	1.38949E-09	0.000572107	NA	NA
IRK3	1.22151E-06	1.26272E-05	NA	NA
IRK4	7.68478E-14	3.06462E-09	9.53003E-14	5.93442E-09

Table 6.12 Dissipation and dispersion errors for $\Delta t = 0.00390625$

Method	DG Diss.	DG Disp.	PG Diss.	PG Disp.
BDF1	0.000889826	0.00154608	0.000883302	0.00152802
BDF2	5.21164E-08	3.39046E-05	3.09133E-08	3.67122E-05
MEBDF2	1.00189E-06	0.000134917	8.57947E-07	0.000131536
MEBDF3	2.17444E-08	2.48574E-07	1.88538E-08	1.85942E-07
MEBDF4	1.82132E-14	6.6767E-10	6.97796E-14	1.0E-09
IRK2	1.99624E-11	2.07042E-06	NA	NA
IRK3	4.17082E-10	0.000000005	NA	NA
IRK4	2.5425E-14	3.20798E-11	7.57973E-15	4.63753E-12

Table 6.13 Dissipation and dispersion errors for $\Delta t = 0.00390625$

Method	DG Diss.	DG Disp.	PG Diss.	PG Disp.
BDF1	0.000889826	0.00154608	0.000883302	0.00152802
BDF2	5.21164E-08	3.39046E-05	3.09133E-08	3.67122E-05
MEBDF2	1.00189E-06	0.000134917	8.57947E-07	0.000131536
MEBDF3	2.17444E-08	2.48574E-07	1.88538E-08	1.85942E-07
MEBDF4	1.82132E-14	6.6767E-10	6.97796E-14	1.0E-09
IRK2	1.99624E-11	2.07042E-06	NA	NA
IRK3	4.17082E-10	0.000000005	NA	NA
IRK4	2.5425E-14	3.20798E-11	7.57973E-15	4.63753E-12

CHAPTER 7

CONCLUSION

In this work, a number of higher order temporal schemes are implemented for the one-dimensional Maxwell equations within finite element frameworks. Since the finite-difference time domain method is so prevalent in the field of computational electromagnetics, very little investigation has been performed using finite element methods for the Maxwell equations. To this end, high order temporal accuracy has been incorporated into discontinuous Galerkin and Petrov-Galerkin formulations. The time integration methods used here were the first and second order backward differentiation formulas (BDF), second, third and fourth order modified extended backward differentiation formulas (MEBDF), and second, third and fourth order implicit Runge-Kutta (IRK) schemes. The Petrov-Galerkin formulation utilizes a singly diagonally implicit Runge-Kutta (SDIRK) scheme whereas the discontinuous Galerkin method employs an explicit first stage stiffly accurate singly diagonally implicit Runge-Kutta (ESDIRK) approach. For all of these methods, it was shown that they achieved the expected convergence rate for a wave propagation problem to which the analytical solution was known.

BIBLIOGRAPHY

[1] David B Davidson. *Computational electromagnetics for RF and microwave engineering*. Cambridge University Press, 2005.

[2] J.S. Hesthaven and T Warburton. Discontinuous galerkin methods for the time-domain maxwell's equations. *ACES Newsletter*, 2004.

[3] JH Coggon. Electromagnetic and electrical modeling by the finite element method. *Geophysics*, 36(1):132–155, 1971.

[4] J-M Jin and Valdis V Liepa. Application of hybrid finite element method to electromagnetic scattering from coated cylinders. *IEEE transactions on antennas and propagation*, 36(1):50–54, 1988.

[5] Xingchao Yuan. Three-dimensional electromagnetic scattering from inhomogeneous objects by the hybrid moment and finite element method. *IEEE Transactions on Microwave Theory and Techniques*, 38(8):1053–1058, 1990.

[6] Kane S Yee et al. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Trans. Antennas Propag*, 14(3):302–307, 1966.

[7] A Ditkowski, K Dridi, and Jan S Hesthaven. Convergent cartesian grid methods for maxwell's equations in complex geometries. *Journal of Computational Physics*, 170(1):39–80, 2001.

[8] James S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods Algorithms, Analysis and Applications*. Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA, 2008.

[9] Jan S Hesthaven and Timothy Warburton. Nodal high-order methods on unstructured grids: I. time-domain solution of maxwell's equations. *Journal of Computational Physics*, 181(1):186–221, 2002.

[10] Hassan Fahs and Stéphane Lanteri. A high-order non-conforming discontinuous galerkin method for time-domain electromagnetics. *Journal of computational and applied mathematics*, 234(4):1088–1096, 2010.

[11] W Kyle Anderson, Li Wang, Sagar Kapadia, Craig Tanis, and Bruce Hilbert. Petrov–galerkin and discontinuous-galerkin methods for time-domain and frequency-domain electromagnetic simulations. *Journal of Computational Physics*, 230(23):8360–8385, 2011.

[12] Gabor Szeg. *Orthogonal Polynomials*, volume 23. American Mathematical Soc., 1939.

[13] D Wirasaet, EJ Kubatko, CE Michoski, S Tanaka, JJ Westerink, and C Dawson. Discontinuous galerkin methods with nodal and hybrid modal/nodal triangular, quadrilateral, and polygonal elements for nonlinear shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 270:113–149, 2014.

[14] James C Newman and William K Anderson. Investigation of unstructured higher-order methods for unsteady flow and moving domains. In *22nd AIAA Computational Fluid Dynamics Conference*, page 2917, 2015.

[15] J Taylor Erwin, W Kyle Anderson, Sagar Kapadia, and Li Wang. Three-dimensional stabilized finite elements for compressible navier–stokes. *AIAA journal*, 51(6):1404–1419, 2013.

[16] J Taylor Erwin, W Kyle Anderson, Li Wang, and Sagar Kapadia. High-order finite-element method for three-dimensional turbulent navier-stokes. In *21st AIAA Computational Fluid Dynamics Conference*, page 2571, 2013.

[17] Li Wang, W Kyle Anderson, Sagar Kapadia, and Lafayette K Taylor. Multiscale large eddy simulation of turbulence using high-order finite element methods. In *Seventh AIAA Theoretical Fluid Mechanics Conference*, 2014.

[18] W KYLE ANDERSON, Li Wang, James Newman, and Sagar Kapadia. Extension of the petrov-galerkin time-domain algorithm for dispersive media. *IEEE microwave and wireless components letters*, 23(5):234–236, 2013.

[19] Chao Liu, James C Newman III, W Kyle Anderson, and Behzad Reza Ahrabi. Three-dimensional dynamic overset method for stabilized finite elements. In *22nd AIAA Computational Fluid Dynamics Conference*, page 3423, 2015.

[20] Chao Liu, James C Newman, and William K Anderson. A streamline/upwind petrov galerkin overset grid scheme for the navier-stokes equations with moving domains. In *32nd AIAA Applied Aerodynamics Conference*, page 2980, 2014.

[21] Behzad R Ahrabi, W Kyle Anderson, and James C Newman III. High-order finite-element method and dynamic adaptation for two-dimensional laminar and turbulent navier-stokes. In *32nd AIAA Applied Aerodynamics Conference*, 2014.

[22] G-Y Psihoyios and JR Cash. A stability result for general linear methods with characteristic function having real poles only. *BIT Numerical Mathematics*, 38(3):612–617, 1998.

[23] A Nigro, A Ghidoni, S Rebay, and F Bassi. Modified extended bdf scheme for the discontinuous galerkin solution of unsteady compressible flows. *International Journal for Numerical Methods in Fluids*, 76(9):549–574, 2014.

[24] R Tirani and C Paracelli. An algorithm for starting multistep methods. *Computers & Mathematics with Applications*, 45(1):123–129, 2003.

[25] Yidong Xia, Xiaodong Liu, Hong Luo, and Robert Nourgaliev. A third-order implicit discontinuous galerkin method based on a hermite weno reconstruction for time-accurate solution of the compressible navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 79(8):416–435, 2015.

[26] Lawrence L Takacs. A two-step scheme for the advection equation with minimized dissipation and dispersion errors. *Monthly Weather Review*, 113(6):1050–1065, 1985.

VITA

Jeffrey Cox was born and grew up in the country of Yemen where he was homeschooled until the end of high school. He earned a B.A. in Physics from Covenant College on Lookout Mountain Georgia. From there, he went on to pursue a master's degree in Computational Engineering at the SimCenter at the University of Tennessee at Chattanooga, which he completed in August, 2016. He hopes to someday pursue a Ph.D. in a related field.